М.Свами, К.Тхуласираман

ГРАФЫ, СЕТИ И АЛГОРИТМЫ

М.: Мир, 1984, 454 стр.

В книге специалистов из Канады и Индии излагаются основы теории графов и ее применение к сетям с сосредоточенными параметрами в электро- и вычислительной технике. Рассматриваются вопросы цикломатики, связности, устойчивости, вложимости и раскраски графов, что позволяет определить чувствительность сети, а также разработать эффективные алгоритмы анализа и оптимизации графов.

Для специалистов по электротехническим сетям и вычислительной технике.

Содержание

Предисловие редактора перевода 5
Предисловие 7
Часть I. Теория графов 11
1. Основные понятия **11**
1.1. Основные определения 11
1.2. Подграфы и дополнения 13
1.3. Маршруты, цепи, пути и циклы 16
1.4. Связность и компоненты графа 18
1.5. Операции Над графами 19
1.6. Специальные графы 23
1.7. Точки сочленения и разделимые графы 25
1.8. Изоморфизм и 2-изоморфизм 27
1.9 Замечания, касающиеся литературы 29
Упражнения 30
2. Деревья, разрезающие множества и циклы **33**
2.1. Деревья, остовы и кодеревья 33
2.2. k-деревья, остовные k-деревья, леса 37
2.3. Ранг и цикломатическое число 39
2.4. Базисные циклы 39
2.5. Разрезающие множества 40
2.6. Разрез 41
2.7. Базисные разрезающие множества 43
2.8. Остовы, циклы и разрезающие множества 44
2.9. Замечания, касающиеся литературы 47
Упражнения 48
3. Эйлеровы и гамильтоновы графы **50**
3.1. Эйлеровы графы 51
3.2. Гамильтоновы графы 55
3.3. Замечания, касающиеся литературы 60
Упражнения 60
4. Графы и векторные пространства **62**
4.1. Группы и поля 62
4.2. Векторные пространства 64

4.3. Векторное пространство графа 67
4.4. Размерность подпространств циклов и разрезов 71
4.5. Связь между подпространствами циклов и разрезов 73
4.6. Ортогональность подпространств циклов и разрезов 74
4.7. Замечания, касающиеся литературы 76
Упражнения 76
5. Ориентированные графы 78
5.1. Основные определения и понятия 78
5.2. Графы и отношения 84
5.3. Ориентированные и корневые деревья 85
5.4. Ориентированные эйлеровы графы 88
5.5. Ориентированные остовы и ориентированные эйлеровы цепи 91
5.6. Ориентированные гамильтоновы графы 93
5.7. Ациклические ориентированные графы 95
5.8. Турниры 96
5.9. Замечания, касающиеся литературы 98
Упражнения 98
6. Матрицы графов 100
6.1. Матрица инциденций 100
6.2. Матрица разрезов 102
6.3. Цикломатическая матрица 105
6.4. Соотношение ортогональности 108
6.5. Подматрицы матриц разрезов, инциденций и циклов 110
6.6. Унимодулярные матрицы 113
6.7. Число остовов 115
6.8. Число остовных 2-деревьев 118
6.9. Число ориентированных остовов в ориентированном графе 120
6.10 Матрица смежности 123
6.11. Графы Коутса и Мэзона 126
6.12. Замечания, касающиеся литературы 134
Упражнения 134
7. Планарность и двойственность 137
7.1. Планарные графы 137
7.2. Формула Эйлера 139
7.3. Теорема Куратовского и другие характеризации планарности 141
7.4. Двойственные графы 143
7.5. Планарность и двойственность 147
7.6. Замечания, касающиеся литературы 149
Упражнения 149
8. Связность и паросочетания 151
8.1. Связность или вершинная связность 151
8.2. Реберная связность 155
8.3. Графы с заданными степенями 156
8.4. Теорема Менгера 160

8.5. Паросочетания 161
8.6. Паросочетания в двудольных графах 163
8.7. Паросочетания графов общего вида 167
8.8. Замечания, касающиеся литературы 171
Упражнения 171
9. Покрытия и раскраски **174**
9.1. Независимые множества и вершинные покрытия 174
9.2. Реберные покрытия 179
9.3. Реберная раскраска и хроматический индекс 180
9.4. Вершинная раскраска и хроматическое число 184
9.5. Хроматические полиномы 186
9.6. Проблема четырех красок 189
9.7. Замечания, касающиеся литературы 190
Упражнения 190
10. Матроиды **193**
10.1. Основные определения 193
10.2. Фундаментальные свойства 196
10.3. Эквивалентные системы аксиом 198
10.4. Двойственность матроидов и графоиды 200
10.5. Ограничение, сужение и миноры матроида 204
10.6. Представимость матроидов 206
10.7. Бинарные матроиды 207
10.8. Ориентируемые матроиды 210
10.9. Матроиды и «жадный» алгоритм 212
10.10. Замечания, касающиеся литературы 214
Упражнения 215
Часть II. Теория цепей 218
11. Графы и электрические цепи **218**
11.1. Преобразование контуров и сечений 220
11.2. Системы контурных уравнений и уравнений сечений 223
11.3. Метод смешанных переменных 228
11.4. Главное разбиение графа 230
11.5. Уравнения состояния 235
11.6. Свойство неусиления в резистивных цепях 241
11.7. Замечания, касающиеся литературы 242
Упражнения 243
12. Резистивные n-полюсные цепи **245**
12.1. Введение 245
12.2. Y-матрицы резистивной n-полюсной цепи ранга га 253
12.3. Реализация (n+1)-узловых резистивных n-полюсных цепей (подход Седербаума) 263
12.4. Реализация цикломатической матрицы и матрицы сечений 268
12.5. Реализация (n+1)-узловых резистивных n-полюсных цепей (подход Гуиллемина) 281

12.6. Замечания, касающиеся литературы 287
Упражнения 288
13. Функция цепи и чувствительность цепи 292
13.1. Топологические формулы для RLC-цепей без взаимных индуктивностей 292
13.2. Топологические формулы для общих линейных цепей 298
13.3. Сопряженная цепь и вычисление чувствительности цепи 305
13.4. Замечания, касающиеся литературы 312
Упражнения 313
Часть III. Теория цепей 315
14. Алгоритмы анализа графов 315
14.1. Транзитивное замыкание 316
14.2. Транзитивная ориентация 322
14.3. Поиск в глубину 331
14.4. Двусвязность и сильная связность 338
14.5. Сводимость графа программы 343
14.6. Доминаторы в графе программы 351
14.7. Замечания, касающиеся литературы 359
Упражнения 359
15. Оптимизационные алгоритмы 361
15.1. Кратчайшие пути 361
15.2. Деревья с минимальной длиной взвешенных путей 368
15.3. Оптимальные деревья бинарного поиска 376
15.4. Максимальные паросочетания в графе 380
15.5. Максимальные паросочетачия в двудольном графе 389
15.6. Совершенное паросочетание, оптимальное назначение и составление расписаний 396
15.7. Потоки в транспортной сети 404
15.8. Оптимальное ветвление 418
15.9. Замечания, касающиеся литературы 422
Упражнения 423
Литература 425
Предметный указатель 444

Предметный указатель

Аксиома баз матроида 198
- независимости матроида 193
- ранга матроида 198
- циклов матроида 199
Алгоритм Габова 385
- Гильберта и Мура 166
- Данцига 423
- двусвязности 340
- Дейкстры 364
- декомпозиции Седербаума схемной матрицы 263
- доминаторов 354
- "жадный" 212
- Кана — Мункреса 402
- кратчайших путей из данной вершины ко всем остальным вершинам 364
- Ленгауэра и Тарьяна 354

- максимального паросочетания в графе общего вида 385
- - - - двудольном графе 394
- - - - транспортной сети (помечивающий алгоритм) 410
- между всеми парами вершин 367
- нахождения минимального остова 424
- оптимального вектора длин путей 374
- - ветвления 421
- - дерева бинарного поиска 381
- - назначения 402
- Пнуели, Лампла и Ивена 323
- поиска в глубину в неориентированном графе 334
- - - - - ориентированном графе 334
- приближенный 423
- реализации графов с заданными степенями 156—159
- - матрицы сечений — цикломатической матрицы 268
- - (n+ цепи, узловой, резистивной, n, полюсной, 1, 263
- сводимости графа программы 348
- сильной связности 343
- совершенного паросочетания 399
- столбце-ориентированный 319
- строко-ориентированный 319
- Тарьяна 348
- транзитивного замыкания 3201
- транзитивной ориентации 323
- Уоррена 320
- Уоршолла 318
- Флойда 367
- Форда — Фалкерсона 197
- Хаффмана 374
База матроида 194

Базис векторного пространства 65
Блок графа 26
Валентность 12
Вектор базисный 65
- длин путей 370
- инциденций 100
- координаты 67
- линейно независимый 65
- ортогональный 67
- разреза 76
- циклический 76, 106
n-вектор 67
Вершина 11
- висячая 12
- внешняя 384
- внутренняя 17
- - n-полюсной цепи 17
- звезды 255
- изолированная 79
- инцидентная 12
- конечная 78
- - маршрута 80
- ориентированного маршрута 86
- концевая 11, 16, 78, 80, 255
- короткозамыкаемая 255
- насыщенная (в паросочетании) 162
- начальная 80
- - в графе программы 131
- - маршрута 97
- - ориентированного маршрута 80
- несоотносимая 334
- полностью сканированная 332
- помеченная 88
- связанная 18
- сильно связанная 81
- смежная 12, 79
- соответствующая усекаемой строке 101
- терминальная 16
- центральная 49
Вес ребра 59
- цикла 59
Ветвление оптимальное 418
Ветвь дерева 33
- неполюсная 247
- полюсная 247
Включение последовательности 141
Вычисления параллельные 210

Гаммоид 217
Гиперплоскость матроида 216
Гипотеза четырех красок 190
Глубина 332
Граф ассоциированный
ориентированный 171
- ациклический 33, 80
- вершинно непересекающийся 28
- - k-раскрашиваемый 184
- взвешенный 135
- гамильтонов 55
- гамильтоново связный 61
- гомеоморфный 142
- двойственный 143
- двудольный 23
- деревьев 49
- k-дольный 23
- изоморфный 28
- изоморфный, 1, 28
- изоморфный, 2, 28
- квазисильно связный 83
- конденсированный 82
- Кoymca 127
- кубический 31
- лежащий в основе
неориентированный 79
- минимально связный 48, 82
- - - ориентированный 82
- Мэзона 126, 131
- неориентированный 11, 100, 103
- неразделимый 25
- однородный 23
- k-однородный 23
- ориентированный 11, 78, 100, 103
- - ациклический 80
- - гамильтонов 93
- - эйлеров 88
- параллельно-последовательный 150
- Петерсена 143
- планарный 137
- - с одной терминальной парой 150
- полный 23
- - двудольный 23
- - k-дольный 23
- - ориентированный 93
- потоков Кoymca 127
- программы 130
- произвольно гамильтонов 5&
- - эйлеров 54, 55
- простой 11
- пустой 11
- разделимый 25
- реберно непересекающийся 28
- - k-раскрашиваемый 180
- - k-хроматический 180
- к-реберно связный 155, 172
- рефлексивный ориентированный 85
- самодвойственный 150
- самодополнительный 30
- сводимый 344
- связной 18, 80
- - ориентированный 80
- сигнальный Мэзона 131
- сильно связный 81
- симметрично ориентированный 85
- с одной терминальной парой 150
- стягиваемый 22
- транзитивно ориентируемый 85, 322
- тривиальный 25
- укладываемый 137
- k-факторизуемый 172
- k-хроматический 185
- эйлеров 51
Графоид 203
Группа абелева 62, 63
Декодирование 369
Дерево 33, 85
- бинарного поиска 376
- бинарное 87
- в графе 33
- венгерское 397
- доминаторов 351
- корневое 85
- линейное 255
- нормальное 236
- оптимальное бинарного поиска 378
- ориентированное 85
- ПВГ 333

- помеченное 117
- расширенное бинарного поиска 376
k-дерево остовное 37
Дефицит 163
Диакоптика 242
Диаметр графа 30
Длина взвешенных путей 87
- ориентированного ребра 361
- пути 17, 361
- цикла 17
Доминатор (вершины)
непосредственный 351
Дополнение ортогональное 67
- подграфа 16
- простого графа 15
Дуга заходящая 79
- инцидентная 79
- исходящая 79
- параллельная 78
Задача о кёнигсбергских мостах 50
- - коммивояжере 59
- - свадьбах 161
- составления расписаний 396
Закон Кирхгофа для напряжений 218
- - - токов 218
Замыкание бинарного отношения
транзитивное 316
- графа 61
- - транзитивное 316
- подмножества 216
Звезда 255
Изоморфность 66
Индекс хроматический 180
Источник напряжения 218
- тока 218
Клика 192
Кобаза матроида 200
Кодерево 33
k-кодерево 37
Код префиксный 88
Ко-лес 38
Колесо 38
Коллапсирование вершины 350
Комбинация линейная 65
- параллельная параллельно-
последовательных графов 150
- последовательная параллельно-
последовательных графов 160
Компонента 18, 169
- графа 80
- двусвязная 341
- нечетная 169
- ориентированного графа 80
- сильно связная 81
- четная 169
Конденсированный образ 82
Контур 80
- гамильтонов 93
Конфигурация полюсная 245
- - в виде звезды 245
- - - - линейного дерева 245
Координата вектора 67
Копетля матроида 200
Коранг матроида 200
Корень в ориентированном графе 85
- компоненты (в ПВГ) 336
- ПВГ 332
Коцикл 200
- базисный матроида 201
- матроида 200
Лемма о раскраске дуг 98
Лес 38
Лист 15
Мажорантность 56
Маршрут 16, 80
- замкнутый 17, 80
- ориентированный 79
- открытый 17
Матрица базисная разрезающих
множеств 105
- - цикломатическая 107, 208
- бистохастическая 172
- гипердоминантная 258
- достижимости 121
- инциденций графа 100
- - ориентированного графа 100
- коциклическая матроида 209
- модифицированная сечений 289

- - цикломатическая 290
- однородно сужающаяся 260
- подстановок 172
- полных сопротивлений контура 225
- - - холостого хода 33
- полустепеней захода 121
- - исхода 121
- проводимостей короткого замыкания 246
- - сечений 248
- разрезов графа 103
- сечений 219
- - базисная 220
- смежности взвешенного ориентированного графа 126
- - графа 123, 126
- - ориентированного графа 123
- с равными алгебраическими дополнениями 117
степеней 117
- схемная 44
- узловых проводимостей 228
- унимодулярная 113
- усеченная инциденций 101
- цикломатическая графа 106
- - матроида 209
- - ориентированного графа 106
(O,1) матрица 166
Матроид 193
- бинарный 207
- графический 196
- двойственный 200
- двудольный 217
- изоморфный 196
- кографический 195
- неразделимый 216
- несвязный 216
- однородный 215
- ориентируемый 216
- паросочетаний 195
- правильный 217
- представимый 226
- разделимый 216
- разрезов 194
- регулярный 215
- связей 194, 216
- связный 216
- трансверсалей
- Фано 216
- циклический 194
- эйлеров 217
Место узкое 413
Метка вершины 384
- начальная 384
- ребра 384
Метод Коутса 126
- Мэзона 131
- Татта 121
Минор матроида 206
Множество базисное циклов 40, 43
- внутреннее устойчивое 174
- доминирующее 190
- зависимое матроида 194
- замкнутое матроида 216
- - по отношению к бинарному отношению 62
- инциденций 76
- лексикографически большее 213
- - максимальное 213
- независимое графа 174
- - максимальное графа 174
- матроида 194
- оптимальное по Гейлу 213
- разделяющее 152
- разрешающее 40
- - базисное 43, 76
L-множество 271
Модификация Эдмондса и Карпа помечивающего алгоритма 412
Мост 31
Назначение оптимальное 400
Напряжение между узлом и базой 222
- узловое 222
Нуль-граф 11
Нуль-пространство 207
Область планарного графа 139
Образ конденсированный графа 82

Обхват 30
Объединение графов 21
- матроидов 217
Ограничение графа 204
- матроида 205
- по пропускной способности 03, 4
Окрестность 219
Операции над графами 62
Определитель главный 115
Оптимизация кода 374
- сети 410
Ориентация графа 78
- ребра 78
- транзитивная 109
Ортогональность 67
Остов графа 33
- максимально удаленный 230
- ориентированный 87
Отец (вершины) 332
Отношение бинарное 84
- рефлексивное 84
- симметричное 84
- транзитивное 84
- эквивалентности 84
Отождествление (замыкание) вершиий 22
Паросочетание максимальное 162
- полное (в двудольном графе) 163
- совершенное 169
Переворачивание 28
Переменная контура 222
- сечения 222
Пересечение графов 21
Петля 11, 78
- матроида 194
- обратной связи 133
Подграф 13, 79
- венгерский 387
- вершинно-порожденный 14
- главный 233
- критический 418
- максимальный 14
- минимальный 14
- остовный 13
- порожденный 14, 79
- равенств 188
- реберно-порожденный 14
- собственный 13
k-подграф 20
Поддерево 33
- левое 376
- правое 376
Подмножество зависимое 194
- максимальное 14, 199
- минимальное 14
Подпространство ортогональное 67
- разрезов 70
- циклов 69
Поиск в глубину в графе 332
- - - - ориентированном графе 335
- - ширину 360
Покрытие вершинное 174
- минимальное 175, 179
- реберное 179
Поле Галуа 63, 64
Полином хроматический 186
Полудоминатор 352
Полустепень захода 79
- исхода 79
Полюс концевой 255
- подобно ориентированный 255
- противоположно ориентированный 255
Помечивание обратное 410
- прямое 410
Порядок редукции 350
Последовательность выводимая 231
- графическая 56
- де Брёйна 88
- остаточная 157
- очков 97
- Пруфера 147
- степеней 56
- чередующаяся (по отношению к независимому множеству) 176
Поток максимальный 406
Потомок (вершины) собственный 334
Предок (вершины) собственный 334

Преобразование контурное 221
- полюсное 222
- сечения 222
- узловое 222
Проблема оптимизации кода 372
- NP-полная 359
- расширения 383
- NP-сложная 359
- четырех красок 189
Представление матроида стандартное 206
Проекция стереографическая 138
Произведение полных сопротивлений 293
- проводимостей 293
- скалярное векторов 67
Пространство векторное 64
- нуль матрицы 207
Проход (через вершину в ПВГ) 332
Псевдовершина 420
Путь 17, 180
- добавляющий 168
- f-дополняющий 408
- кратчайший 361
- - по отношению к паросочетанию 390
- максимальный 282
- f-ненасыщенный 195
- ориентированный 80
- - гамильтонов 55
- M-чередующийся 167
Разбиение главное 233
- двудольное 23, 176
Разделитель 42
Размерность векторного пространства 65
Разметка допустимая вершинная 399
Разрез ориентированный 42, 98
Ранг 82
- графа 39
- матроида 194
- ориентированного графа 82
Ранжирование 97
Раскраска 202
- k-вершинная 182
- - правильная 184
- ориентируемого матроида 211
- k-реберная 180
- - оптимальная 180
- - правильная 180
Расстояние между вершинами 30
- - остовами 230
Расщепление вершины 28
- точки сочленения 28
Реализация (n+ полюсной, резисторной, цепи, 1, 263
Ребро 11
- висячее 12
- критическое 418
- f-насыщенное 408
- независимое 162
- f-ненасыщенное 408
- нециклическое 17
- f-нулевое 408
- обратное 332
- - по отношению к ориентированному пути 407
- пересекающее 121
- подходящее 418
- f-положительное 408
- полюсное 245
- помеченное 329
- последовательное 141
- прямое 332
- - по отношению к ориентированному пути 407
- - - - - ПВГ 332
- рассмотренное (в ПВГ) 332
- светлое 167
- смежное 12, 79
- темное 167
- циклическое 17
Редукция транзитивная 359
Свойство ассоциативности 62
- коммутативности 62
- неусиления 241
- схемности 257
Связность вершинная 151

- реберная 155
Сеть транспортная 404
Сечение ориентированное 241
- - полюсной цепи 241
Система различных представителей 165
- уравнений контурная 224
- - сечений 226
- - со смешанными переменными 230
Скаляр 65
Слияние последовательности 141
Слово кодовое 372
Соединение параллельное 290
- псевдопоследовательное 290
- факторное, ориентированного, графа, 1, 128
Сортировка топологическая 95
Составление расписания 403
Список смежности 316
Способность пропускная разреза 406
- - ребра 405
Степени свободы топологические 235
Степень вершины 79
- области 141
Сток 404
Стягивание ребра 22
Сужение матроида 204
Сумма кольцевая графов 21
- прямая подпространств 65
- характеристическая 372
Теорема Бержа о чередующейся цепи 168
- Бине—Коши 116
- Брукса 185
- Визинга 183
- добавления 196
- Кёнига 165
- Кёнига — Эгервари 166
- Куратовского 141
- Кэли 117
- Менгера 160
- Мендельсона и Далмеджа 169
- о максимальном потоке и минимальном разрезе 409
- - пяти красках 189
- Татта о совершенных паросочетаниях 170
- Теллежена 222
- Холла 164
- четырех красок 190
- Эдмондса о k-реберной связности 159
Теория экстремальных графов 190
Толщина 149
Точка сочленения 25
Трансверсаль частичная 165, 216
Транспозиция цепи 312
Треугольник 31
Турнир 96
Удаление вершины 22
- ребра 22
Узел 219
Укладка планарная 137
Улучшение реберной раскраски 183
Уравнения состояния 235
- узловые 228
Уязвимость 181
фактор, ориентированного, графа, 1, 124
k-фактор 172
Формула коэффициента усиления Каутса 130
- - - Мэзона 133
- Эйлера 139
Функция Аккермана 136
- коранга матронда 200
- ранга матроида 194
Хорда 33
- неполюсная 249
- полюсная 249
«Цветок» 168
Цепь 17, 80
- вывода 327
- замкнутая 17
- - ориентированная 80
- ориентированная 80
- - эйлерова 51, 88
- открытая 17, 80

- - ориентированная 80
- - - эйлерова 51, 88
- n-полюсная 246
- - нагруженная 246
- чередующаяся (по отношению к паросочетанию) 167
Цикл 17, 80
- базисный 40
- - матроида 410
- гамильтонов 55
- матроида 410
- ориентированный 80
Число вершинного покрытия 175
- внутренней устойчивости 174
- независимости 174
- паросочетаний 162
- реберного покрытия 179
- реберно-хроматическое 180
- скрещиваний 149
- хроматическое 185
- цикломатическое графа 82, 89
Эквивалентности классы 81
Элементы матроида 63, 193
Ячейка 140

Предисловие редактора перевода

Предлагаемая читателю книга посвящена теории графов, которая приобретает все возрастающий интерес у специалистов самых различных областей науки и техники. Привлекательность теории графов объясняется не только широчайшим разнообразием возможностей ее применения, но и красотой результатов, достигаемых простыми средствами. Сделав свои первые шаги еще в XVIII в., она постоянно набирала темпы и в настоящее время вошла в пору юности, когда открыто много путей и не ясно, какой из них главный. Действительно, в рамках теории графов имеется множество различных направлений; результаты ее формулируются с привлечением понятий теории множеств, топологии, алгебры, но методы теории графов нельзя рассматривать как простую совокупность методов этих разделов математики.

Теория графов и мографов (гиперграфов) является эффективным аппаратом формализации современных инженерных и научных задач, как уже сказано, в различных областях знаний. Так, например, в квантовой теории поля взаимодействие элементарных частиц с внешним полем описывается связными графами (диаграммами Фейнмана) [1, 2]; язык теории графов удобен при проведении системных исследований [3], описании организации генетических систем [4], автоматизированном управлении производством [5, 6]. В последнее время большое применение нашла теория графов в современной вычислительной технике и кибернетике: в теоретическом программировании [7], при проектировании ЭВМ на ЭВМ и сетей ЭВМ [8—12], баз данных [13, 14], систем логического управления [5, 14—18].

Книга написана профессором Университета Конкордии (г. Монреаль, Канада) М. Свами и профессором Индийского технологического института (г. Мадрас, Индия) К. Тхуласираманом. Основу ее составили курсы, читаемые авторами для студентов и аспирантов. Сфера научных интересов авторов лежит в области применения теории графов в электрических цепях. Этим объясняется выбор рассмотренной в части II книги области применения теории графов. В этой части рассматриваются графовые свойства цепей и применение графовых методов для реализации цепей с заданными свойствами. Материал этой части специфичен как по терминологии, так и по способам представления и предполагает хорошее знакомство с теорией графов в пределах, обеспечиваемых содержанием части I. Изложение этой части несколько отличается расстановкой акцентов по сравнению с другими монографиями по теории графов, что обусловлено учебной направленностью книги. С другой стороны, необходимо отметить, что в ней дано одно из самых широких введений в теорию матроидов, существующих в отечественной и переводной литературе.

Часть III также широко использует материал части I. В ней содержится описание множества алгоритмов, которые ориентированы главным образом на применение в вычислительной технике. Способ изложения алгоритмов, принятый в книге, отражает тенденции проявляющиеся при разработке программного обеспечения. Доказана корректность каждого из алгоритмов, и приведены оценки их сложности, что является несомненным достоинством книги. По-видимому, недостаток места не позволил авторам дать введение в теорию NP-полноты, прекрасно сделанное в работе [19].

Предлагаемую книгу отличает простота и лаконичность изложения, богатство иллюстраций, примеров и упражнений. Как учебное пособие она практически не

имеет недостатков. Однако квалифицированный специалист найдет в ней знакомый материал.

В книге часто используется термин «характеризация». Авторы понимают под ним любые необходимые и достаточные условия существования некоторого свойства. Мы оставили этот термин без изменения, хотя в отечественной литературе под характеризацией принято понимать определение для данного свойства множества запрещенных фигур.

Перевод выполнен М. В. Горбатовой (гл. 1—4), В. Л. Торховым (предисловие, гл. 5—10), канд. техн. наук С. А. Фроловым (гл. 11—13), В. Н. Четвериковым (гл. 14—15).

В. А. Горбатов

Библиография

1. Белов В. В., Воробьев Е. М., Шаталов В. Е. Теория графов.— М.: Высшая школа, 1976.
2. Фам Ф. Особенности процессов многократного рассеяния: Пер. с франц.— М.: Мир, 1972.
3. Горбатов В. А. Теория частично упорядоченных систем.— М.: Сов. радио, 1976.
4. Миркин Б. Г., Родин С. Н. Графы и гены.— М.: Наука, 1977.
5. Горбатов В. А., Кафаров В. В., Павлов П. Г. Логическое управление технологическими процессами.— М.: Энергия, 1978.
6. Брегман В. И. Графы в задачах управления производством.— М.: Статистика, 1974.
7. Ершов А. П. Введение в теоретическое программирование.— М.: Наука, 1977.
8. Горбатов В. А. Схемы управления ЦВМ и графы.— М.: Энергия, 1971.
9. Автоматизация проектирования сложных логических структур/ Под редакцией В. А. Горбатова.— М.: Энергия, 1978.
10. Ландау И. Я. Применение ЦВМ для проектирования ЦВМ.— М.: Энергия, 1974.
11. Сети ЭВМ/ Под редакцией В. М. Глушкова.— М.: Связь, 1977.
12. Вычислительные сети и сетевые протоколы: Пер. с англ.— Дэвис Д., Барбер Д., Прайс У., Соломонидес С.— М.: Мир, 1982.
13. Берзтисс А. Т. Структуры данных.— М.: Статистика, 1974.
14. Горбатов В. А., Павлов П. Г., Четвериков В. Н. Логическое управление информационными процессами.— М.: Энергоатомиздат, 1983.
15. Горбатов В. А. Семантическая теория проектирования автоматов.— М.: Энергия, 1979.
16. Горбатов В. А., Останков Б. Л., Фролов С. А. Регулярные структуры автоматного управления.— М.: Машиностроение, 1980.
17. Богомолов А. М., Грунский И. С., Сперанский Д. В. Контроль и преобразования дискретных автоматов.— Киев: Наукова думка, 1975.
18. Мелихов А. Н. Ориентированные графы и конечные автоматы.— М.: Наука, 1971.
19. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов: Пер. с англ.— М.: Мир, 1979.

Посвящается нашим родителям и учителям

Преклоняюсь перед учителем, который вернул зрение человеку, ослепленному недугом незнания

Вишвасара Тантра

Предисловие

В течение последних двух десятилетий среди лиц, занимающихся различными аспектами науки и техники, значительно возросла популярность теории графов — ветви дискретной математики. Родившись при решении головоломок и игр, таких, как задача о кёнигсберских мостах и игра Гамильтона, она стала мощным средством исследования и решения многих задач, возникающих при изучении больших и сложных систем. Действительно, существует ряд систем, изучение которых становится значительно проще с использованием теории графов. Это не удивительно, так как бинарные отношения между объектами некоторого множества удобно представлять графами, а описания систем содержат такие отношения между различными их подсистемами. Кроме того, теория графов оказалась полезной при изучении задач, возникающих в некоторых других разделах математики, таких, как теория групп и теория матриц. Всякий раз, когда встречалась новая область применения теории графов, возникала необходимость в введении и изучении новых понятий или в дальнейшей разработке некоторых известных понятий Эта необходимость в свою очередь возбуждала творческую активность в области исследований различных связанных с ними концепций. Такое непрерывное взаимодействие способствовало быстрому росту этой ветви математики.

Был написан ряд книг, обсуждающих различные аспекты теории графов: анализ, синтез, перечисление, алгоритмы и приложения. В предлагаемой книге делается попытка обсуждения с единых позиций теории графов, ее применения при исследовании электрических цепей и теоретических основ некоторых графовых алгоритмов. Прежде чем мы перейдем к детальному рассмотрению целей и содержания книги, обсудим кратко, каким образом теория графов оказалась полезной в исследовании электрических цепей, исследовании операций и вычислительной технике — трех основных дисциплинах, которые интересуют нас и взаимодействуют с теорией графов с большим успехом и к взаимной выгоде.

Хорошо известно, что основы теории графов были заложены Эйлером при решении задачи о кёнигсбергских мостах. Однако она не находила применения при решении задач в прикладной области до 1847 г., когда Кирхгофом была разработана теория деревьев применительно к исследованию электрических цепей. Основополагающую роль в их изучении играют законы Кирхгофа. Эти законы определяют связь между переменными напряжения и тока в цепи. Эти связи для данной электрической цепи не зависят от природы используемых элементов, скорее они зависят от того, каким образом соединены различные элементы, или, другими словами, от графа, представляющего цепь. Фактически уравнения, описывающие законы Кирхгофа для напряжений и токов, полностью определяются контурами и сечениями графа цепи. Здесь возникает вопрос: всякий ли контур и всякое ли сечение цепи обязательно определяют такие уравнения? Для получения ответов на этот и связанные с ним вопросы необходимо тщательное изучение понятий контура, сечения и дерева графа. Этим объясняется роль теории графов как важного аналитического средства при изучении электрических цепей. Многие важные открытия в теории цепей имеют главным образом теоретико-графовую природу, и усилия, ведущие к таким открытиям, приводят к значительному вкладу в теорию графов.

Транспортные сети и сети связи можно удобно представить графами, поэтому для исследования операций, связанных с такими задачами, возникающими при изучении сетей, как потоки, кратчайшие пути или конструирование надежных сетей,

оказался очень полезным теоретико-графовый подход. Эти исследования внесли значительный вклад в быстрое развитие теории графов в последние два десятилетия. Разработанная Фордом и Фалкерсоном теория потоков в сетях осветила ряд комбинаторных вопросов и позволила получить новые доказательства многих важных теорем теории графов. Закон Кирхгофа о сохранении потока (подобный закону Кирхгофа для токов в электрической цепи) играет центральную роль и в разработке теории потоков в сетях. В последние годы большое внимание было уделено конструированию сетей связи, имеющих заданные свойства. Эта задача обычно сводится к задаче построения экстремального графа (т. е. графа, имеющего минимальное или максимальное число ребер), обладающего заданным значением для одного из топологических параметров. Исследования в этой области имели большое значение для решения так называемых экстремальных задач в теории графов.

Последней областью, включенной в растущий список областей применения теории графов, является вычислительная техника. Для специалистов по вычислительной технике теория графов — это удобный язык выражения понятий из этой области; многие результаты теории графов имеют непосредственную связь с задачами, с которыми им приходится сталкиваться. Широкое применение теория графов недавно получила при исследовании так называемой проблемы оптимизации, возникающей при конструировании компиляторов. В рамках этих исследований были разработаны многие, неизвестные ранее теоретико-графовые понятия. Теория графов имеет большую привлекательность для специалистов по вычислительной технике и помимо использования ее как инструмента в решении конкретных задач. Одним из основных направлений в вычислительной технике является построение эффективных алгоритмов и анализ их сложности, и теория графов (а в общем, и комбинаторный анализ) предоставляет большие возможности для построения таких алгоритмов. Это действительно значительный вклад вычислительной техники в теорию графов.

Все это освещает роль, которую стала играть теория графов в науке и технике, и обусловливает возрастающий вес теории графов в университетских учебных планах. Хотя детальное знание некоторых понятий и результатов не обязательно для исследований в какой-либо определенной области применения, мы полагаем, что досконалное знание различных понятий и глубокое овладение методами, используемыми при изучении этих понятий, будут очень полезны для понимания новых применений, которые в противном случае будут неочевидны. Фактически это верно также и для многих других отраслей математики, таких, как теория комплексного переменного, теория матриц и теория операторов, играющих существенную роль при изучении системотехники.

Эта книга адресована студентам, специализирующимся в математике, электротехнике и вычислительной технике. Как следует из названия книги, она состоит из трех частей, касающихся теории графов, электрических сетей (цепей) и графовых алгоритмов.

В части I (гл. 1—10) мы обсуждаем теорию графов, в ней ставится цель ввести основные понятия и результаты теории графов. В этой части обсуждаются деревья, гамильтоновы и эйлеровы графы, ориентированные графы, матрицы графов, планарность, связность, паросочетания и раскраски. Мы включили в нее также введение в теорию матроидов. Среди представленных тем, связанных с матроидами, можно перечислить самодвойственную систему аксиом Минти, которая делает очевидной двойственность циклов и разрезающих множеств графа, лемму о раскраске дуг, жадный алгоритм и его тесную взаимосвязь с матроидами. В последние годы к теории матроидов проявляют интерес специалисты по теории электрических цепей, поскольку она позволяет глубоко проникнуть в те задачи, с которыми они сталкиваются. Если для математика теория матроидов дает большие возможности по обобщению теоретико-графовых понятий, то для специалиста по вычислительной технике — возможности по построению алгоритмов над матроидами.

В части II (гл. 11—13) обсуждаются аспекты теории электрических цепей, разработка которых имеет существенно теоретико-графовый характер. В гл. 11 обсуждается среди прочего понятие главного разбиения графа, его применение в ме-

тоде смешанных переменных анализа цепей и теоретико-графовое доказательство свойства неусиления резистивных цепей. В гл. 12 рассматриваются ряд результатов теории резистивных цепей и метод реализации цикломатических матриц и матриц сечений. В заключительной главе этой части выводятся топологические формулы для функций цепи. Эти формулы очевидным образом следуют из свойств матриц графа, рассматриваемых в части I. В части II обсуждаются также теорема Теллежена, имеющая теоретико-графовую природу, и ее применение для вычисления чувствительности цепей. Удивительно, что такая важная теорема столько лет была лишена какого-либо внимания.

Часть III, в которой рассматриваются графовые алгоритмы, состоит из двух глав: в гл. 14 рассматривается алгоритм анализа графов, в гл. 15 — алгоритмы, связанные с оптимизационными задачами на графах. Главное внимание уделяется теории, на основе которой осуществляется построение, доказательство правильности и анализ нескольких графовых алгоритмов. Среди прочих обсуждаются алгоритмы сводимости графов потоков, доминаторов, кратчайших путей, паросочетаний, оптимальных деревьев бинарного поиска, потоков в сетях и оптимальных ветвлений. Приводятся также анализ Хопкрофта и Карпа алгоритма двудольного паросочетания и анализ Эдмондса и Карпа помечивающего алгоритма Форда—Фалкерсона. Здесь виден глубокий вклад в теорию графов, сделанный специалистами по вычислительной технике. Из рассмотрения опущено обсуждение *NP*-полных задач, однако эта тема выходит за рамки книги.

Что касается необходимой подготовки для чтения книги, то аспиранты, интересующиеся математикой, почти не будут испытывать затруднения при чтении материала частей I и III. Изложение материала в части II предполагает,что студенты уже знакомы с основным курсом по теории электрических цепей.

Содержание книги может использоваться в рамках различных курсов. Некоторые возможные наименования перечислены ниже.

1. Курс «Теория графов» для студентов, специализирующихся в математике, электротехнике и вычислительной технике, может основываться на гл. 1—10 и разд. 15.7. Для студентов, специализирующихся в вычислительной технике, некоторые разделы гл. 6 можно опустить и добавить разд. 14.3 и 14.4 по алгоритмам поиска в глубину.

2. Курс «Графы и электрические цепи» для аспирантов, специализирующихся в электротехнике, может основываться на гл. 1—7 и 11—13 (кроме разд. 3.2 и 5.6—5.8).

3. Курс «Алгоритмическая теория графов» для студентов, специализирующихся в математике, электротехнике и вычислительной технике, может основываться на материале части III и связанным с ним материале части I. Студентам, специализирующимся в вычислительной технике, разд. 14.5 и 14.6 необходимо проработать особенно тщательно, включая обсуждение алгоритмов для задач манипулирования множествами, что окажется полезным при рассмотрении алгоритмов сводимости графов программы и доминаторов.

Основываясь на этой книге, авторы читали в Университете Конкордии (г. Монреаль, Канада) курс «Теория графов» для студентов, специализирующихся в математике и электротехнике, и курс «Графы и электрические цепи» для аспирантов, специализирующихся в математике. Последний курс читался также вторым автором (К. Тхуласираманом) в Индийском технологическом институте (г. Мадрас, Индия).

Второй автор (К. Тхуласираман) использовал некоторый материал части I в курсе «Комбинаторика и теория графов» и материал части III в курсе «Построение и анализ алгоритмов» для аспирантов Индийского технологического института в Мадрасе, специализирующихся в вычислительной технике.

Мы выражаем нашу искреннюю благодарность д-ру П. К. Раджану из Университета штата Северная Дакота в Фарго и г-ну Р. Яякумару из Индийского технологического института в Мадрасе, которые прочли бóльшую часть рукописи книги, отметили некоторые упущения и ошибки и высказали много предложений, оказавшихся полезными при доработке книги. Мы благодарны также проф. Индийского технологического института в Мадрасе В. Г. К. Мурти и К. Р. Мутукришнану,

проф. В. Рамачандрану и д-ру Л. Ройтману из Университета Конкордии в Монреале, д-ру К. Санкара Рао из Университета штата Северная Дакота в Фарго, д-ру Х. Нараянману из Индийского технологического института в Бомбее и г-дам А. Мохану и П. Нарандрану, студентам Индийского технологического института в Мадрасе за чтение различных частей рукописи и полезные комментарии, д-ру С. А. Чаудуму, Университет Мадураи (Мадураи, Индия), за указание на более простые доказательства результатов теории графов, д-ру В. Бейписвара Рао, Индийский технологический институт в Мадрасе, за разрешение на использование части его неопубликованной работы и проф. В. Хваталу из Университета Мак-Гилла в Монреале, Г. Киши из Токийского технологического института, Л. Ловацу (Венгрия), Р. Е. Тарьяну из Стэнфордского университета (Стэнфорд, шт. Калифорния) и К. Р. Партасарати и его аспирантам из Индийского технологического института в Мадрасе за полезные комментарии и предложения.

К. Тхуласираман выражает благодарность за помощь проф. В. Г. К. Мурти, под замечательным руководством которого он начал свои исследования.

Мы благодарим Университет Конкордии в Монреале за его огромную поддержку. К. Тхуласираман выражает благодарность Индийскому технологическому институту в Мадрасе и Университету Конкордии за их поддержку и ободрение, которые сделали возможным его участие в написании книги.

Мы благодарим наших жен Лейлу Свами и Санту Тхуласираман и детей за их терпение и понимание в течение всего периода нашей работы.

Наконец, нам бы хотелось поблагодарить Глорию Миллер и Камалу Рамачандран за прекрасную перепечатку рукописи.

Монреаль, Канада — *М. Н. С. Свами*
Мадрас, Индия — *К. Тхуласираман*
Сентябрь, 1980

Часть I. Теория графов

1. Основные понятия

Начнем изложение материала с введения нескольких основных понятий теории графов. Установим ряд результатов, основанных на этих понятиях. Эти результаты, иллюстрируя введенные понятия, служат также для приобщения читателя к определенным методам, часто используемым в доказательствах теорем теории графов.

1.1. Основные определения

Граф $G=(V, E)$ состоит из двух множеств: конечного множества элементов, называемых *вершинами*, и конечного множества элементов, называемых *ребрами*. Каждое ребро определяется парой вершин. Если ребра графа определяются упорядоченными парами вершин, то G называется *направленным* или *ориентированным* графом. В противном случае G называется *ненаправленным* или *неориентированным* графом. В первых четырех главах книги рассматриваются ненаправленные графы.

Для обозначения вершин графа будем использовать символы $v_1, v_2, v_3, \ldots$, а для обозначения ребер — $e_1, e_2, e_3, \ldots$. Вершины v_i и v_j, определяющие ребро e_l, называются *концевыми вершинами* ребра e_l. В этом случае ребро e_l обозначается как $e_l=(v_i, v_j)$. Заметим, что в множестве E допускается более чем одно ребро с одинаковыми концевыми вершинами. Все ребра с одинаковыми концевыми вершинами называются *параллельными*. Кроме того, концевые вершины ребра не обязательно различны. Если $e_l=(v_i, v_i)$, то ребро e_l называется *петлей*. Граф называется *простым*, если он не содержит петель и параллельных ребер. Граф G является графом *порядка* n, если множество его вершин состоит из n элементов.

Граф, не имеющий ребер, называется пустым. Граф, не имеющий вершин (и, следовательно, ребер), называется нуль-графом.

Графически граф может быть представлен диаграммой, в которой вершина изображена точкой или кружком, а ребро — отрезком линии, соединяющим точки или кружки, соответствующие концевым вершинам ребра. Например, если $V=\{v_1, v_2, v_3, v_4, v_5, v_6\}$ и $E=\{e_1, e_2, e_3, e_4, e_5\}$, такие, что $e_1=(v_1, v_2)$, $e_2=(v_1, v_4)$, $e_3=(v_5, v_6)$, $e_4=(v_1, v_2)$, $e_5=(v_5, v_5)$, тогда граф $G=(V, E)$ представляется так, как изображено на рис. 1.1. В этом графе e_1 и e_4 — параллельные

ребра, e_5 — петля. Говорят, что ребро *инцидентно* своим концевым вершинам. Две вершины смежны, если они являются концевыми вершинами некоторого ребра. Если два ребра имеют общую концевую вершину, они называются смежными.

Например, в графе на рис. 1.1 ребро e_1 инцидентно вершинам v_1 и v_2; v_1 и v_4 являются смежными вершинами, а e_1 и e_2 — смежными ребрами.

Число инцидентных вершине v_i ребер называется *степенью* вершины и обозначается $d(v_i)$. Иногда степень вершины называется также ее *валентностью*. Вершина степени 1 называется *висячей вершиной*. Единственное ребро, инцидентное висячей вершине, называется *висячим*. Вершина степени 0 называется *изолированной*. По определению петля при вершине v_i добавляет 2 в степень соответствующей вершины. Величины $\delta(G)$ и $\Delta(G)$ обозначают минимальную и максимальную степени вершины в G соответственно.

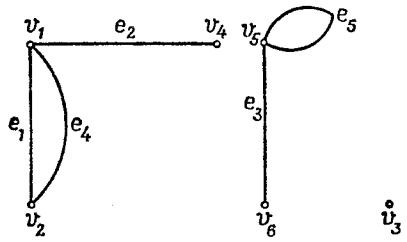


Рис. 1.1. Граф $G=(V, E)$.
$V=\{v_1, v_2, v_3, v_4, v_5, v_6\}$;
$E=\{e_1, e_2, e_3, e_4, e_5\}$.

В графе G на рис. 1.1 $d(v_1)=3$, $d(v_2)=2$, $d(v_3)=0$, $d(v_4)=1$, $d(v_5)=3$, $d(v_6)=1$.

Заметим, что v_3 — изолированная вершина, v_4 и v_6 — висячие вершины, e_2 — висячее ребро. Легко проверить, что сумма степеней вершин в данном графе G равна 10, тогда как число ребер равно 5. Таким образом, сумма степеней вершин графа G равна удвоенному числу ребер графа G и, следовательно, является четным числом. Более того, можно показать, что число вершин графа G нечетной степени также четно. Эти результаты свойственны не только графу на рис. 1.1. Они справедливы, как доказывается в следующих теоремах, для всех графов.

Теорема 1.1. Сумма степеней вершин графа G равна $2m$, где m — число ребер графа G.

Доказательство. Поскольку каждое ребро инцидентно двум вершинам, оно добавляет двойку к сумме степеней графа G. Следовательно, все ребра дают вместе сумму степеней $2m$.

Теорема 1.2. Число вершин нечетной степени в любом графе четно.

Доказательство. Пусть число вершин в графе G равно n. Не нарушая общности, предположим, что степени первых r вершин $v_1, v_2, \ldots, v_r$ четны, а степени оставшихся $n-r$ вершин нечетны. Тогда

$$\sum_{i=1}^{n} d(v_i) = \sum_{i=1}^{r} d(v_i) + \sum_{i=r+1}^{n} d(v_i). \qquad (1.1)$$

По теореме 1.1 сумма в левой части выражения (1.1) четна. Первая сумма в правой части также четна, поскольку каждый член суммы четный. Следова-

тельно, вторая сумма в правой части должна быть четной. Так как каждый член этой суммы нечетный, то число членов в сумме должно быть четным. Другими словами, число $n - r$ вершин нечетной степени должно быть четным.

1.2. Подграфы и дополнения

Рассмотрим граф $G=(V, E)$. $G'=(V', E')$ называется *подграфом* G, если V' и E' являются соответственно такими подмножествами V и E, что ребро (v_i, v_j) содержится в E' только в том случае, если v_i и v_j содержатся в V'. G' называется *собственным подграфом* G, если E' — собственное подмножество E или V' — собственное подмножество V. Если все вершины графа G присутствуют в подграфе G' графа G, тогда G' называется *остовным подграфом* G. Например,

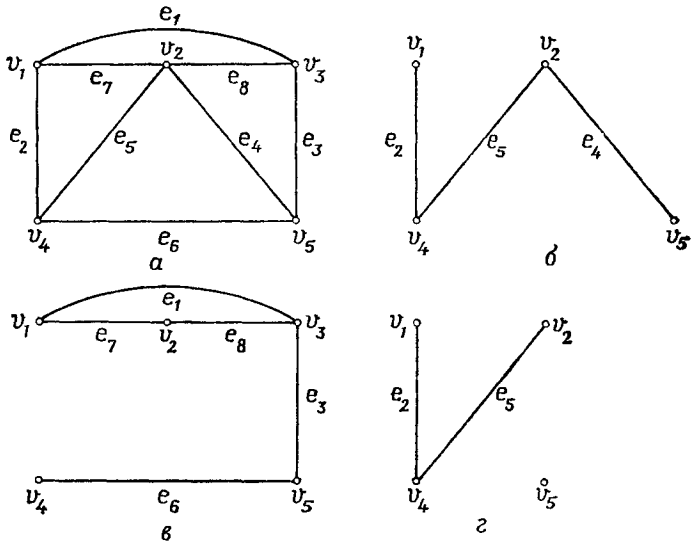


Рис. 1.2. Граф и некоторые его подграфы.
а)—граф G; *б*—подграф G'; *в*—подграф G''; *г*—подграф G'''.

рассмотрим показанный на рис. 1.2, *а* граф G. Граф G', изображенный на рис. 1.2, *б*, является подграфом G. Множество его вершин — $\{v_1, v_2, v_4, v_5\}$. Он является собственным подграфом G. Граф G'' на рис. 1.2, *в* является остовным подграфом G.

Некоторые из вершин подграфа могут быть изолированными. Например, показанный на рис. 1.2, *г* подграф G''' является подграфом с изолированной вершиной.

Если подграф $G'=(V', E')$ графа G не содержит изолированных вершин, тогда по определению подграфа каждая вершина V' является концевой вершиной некоторого ребра E. Таким образом, в

этом случае E однозначно определяет V' и, следовательно, подграф G'. Подграф G' называется *порожденным подграфом графа G на множестве ребер E'* (или просто *реберно-порожденным подграфом* графа G) и обозначается как $\langle E' \rangle$.

Заметим, что множество вершин V' графа $\langle E' \rangle$ является наименьшим подмножеством V, содержащим все концевые вершины ребер в E'. Подграфы G' и G'' на рис. 1.2, *б*, *в* являются реберно-порожденными подграфами графа G на рис. 1.2, *а*, тогда как G''', показанный на рис. 1.2, *г*, не является реберно-порожденным подграфом.

Определим теперь вершинно-порожденный подграф. Пусть V' — подмножество множества вершин V графа $G=(V, E)$. Тогда подграф $G' = (V', E')$ называется *порожденным подграфом графа G на множестве вершин V'* (или просто *вершинно-порожденным подграфом графа G*), если E' является таким подмножеством E, что ребро $(v_i, v_j) \in E$ входит в E' тогда и только тогда, когда v_i и v_j входят в V'. Другими словами, если v_i и v_j принадлежат V', то каждое ребро E, имеющее в качестве концевых вершин v_i и v_j, должно входить в E'. Заметим, что в этом случае V' полностью определяет E' и, таким образом, подграф G'. Следовательно, вершинно-порожденный подграф $G'=(V', E')$ можно обозначить как $\langle V' \rangle$. Например, граф, представленный на рис. 1.3, является вершинно-порожденным подграфом графа G на рис. 1.2, *а*.

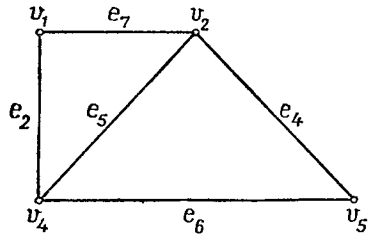


Рис. 1.3. Вершинно-порожденный подграф графа G, изображенного на рис. 1.2, *а*.

Заметим, что множество ребер E' вершинно-порожденного подграфа на множестве V' является таким наибольшим подмножеством E, что концевые вершины всех его ребер принадлежат V'.

Подграф G' графа G называется *максимальным подграфом* по отношению к некоторому свойству P, если G' обладает свойством P и G' не является собственным подграфом никакого другого подграфа графа G, обладающего свойством P.

Подграф G' графа G называется *минимальным подграфом графа G* по отношению к некоторому свойству P, если G' обладает свойством P и никакой подграф графа G, обладающий свойством P, не является собственным подграфом графа G'.

Максимальное и *минимальное подмножества* некоторого множества по отношению к свойству определяются аналогичным образом.

Например, множество вершин V' реберно-порожденного подграфа $\langle E' \rangle$ графа $G=(V, E)$ является минимальным подмножеством V, содержащим концевые вершины всех ребер E'. С другой стороны, множество ребер E' вершинно-порожденного подграфа $\langle V' \rangle$ явля-

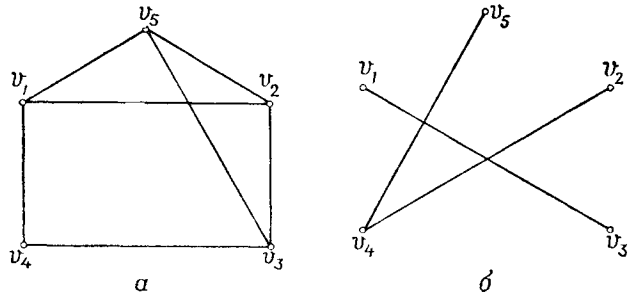


Рис. 1.4. Граф и его дополнение.
а—граф G; *б*—граф $\overline{G}$, дополняющий граф G.

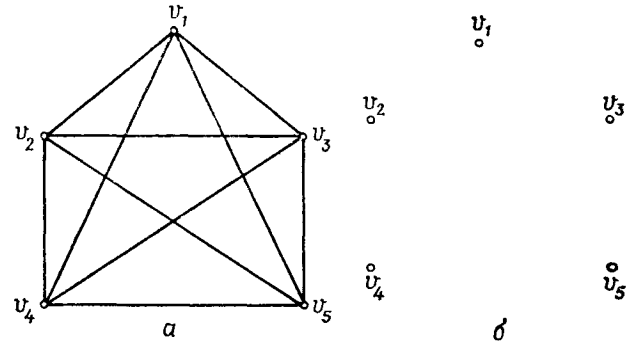


Рис. 1.5. Другой пример графа и его дополнения.
а—граф G; *б*—граф $\overline{G}$ дополняющий граф G.

ется таким максимальным подмножеством E, что концевые вершины всех его ребер принадлежат V'.

Позднее мы увидим, что «компонента» (разд. 1.4) графа G является максимальным «связным» подграфом графа G, а «остовное дерево» (гл. 2) связного графа G является минимальным «связным» остовным подграфом графа G.

Определим теперь понятие «дополнение графа».

Граф $\overline{G}=(V, E')$ называется дополнением простого графа $G=(V, E)$, если ребро (v_i, v_j) входит в E' в том и только в том случае, если оно не входит в E. Другими словами, две вершины v_i и v_j смежны в $\overline{G}$ тогда и только тогда, когда они не смежны в G. На рис. 1.4 представлены граф и его дополнение. В качестве другого примера рассмотрим граф, изображенный на рис. 1.5, *а*. В этом графе между каждой парой вершин имеется ребро. Следовательно, в дополнении $\overline{G}$ графа G вообще не будет ребер, т. е. $\overline{G}$ будет содержать только изолированные вершины. Граф $\overline{G}$ изображен на рис. 1.5, *б*.

Пусть $G'=(V', E')$ является подграфом графа $G=(V, E)$. Подграф $G''=(V, E-E')$ графа G называется дополнением G' в G. Например, подграф G'' на рис. 1.2 является дополнением G' в графе G.

Следующий пример иллюстрирует некоторые из рассмотренных понятий.

Предположим, что мы хотим доказать следующее утверждение:

В любой группе из шести человек трое либо обоюдно знакомы, либо обоюдно незнакомы.

Представляя людей вершинами графа, а наличие знакомства между ними — ребрами, соединяющими соответствующие вершины, можно увидеть, что приведенное выше утверждение можно сформулировать следующим образом:

В любом простом графе G с шестью вершинами имеются три либо попарно смежные, либо попарно несмежные вершины.

Используя определение дополнения графа, замечаем, что это утверждение эквивалентно следующему:

Для любого графа G с шестью вершинами справедливо, что G или $\overline{G}$ содержит три попарно смежные вершины.

Для доказательства этого проделаем следующее. Рассмотрим произвольную вершину v простого графа G с шестью вершинами. Заметим, что, если v не смежна с тремя вершинами в G, она должна быть смежна с тремя вершинами в $\overline{G}$. Поэтому, не нарушая общности, допустим, что v смежна в G с некоторыми тремя вершинами v_1, v_2, v_3. Если две из этих вершин, например v_1 и v_2, смежны в G, то вершины v, v_1 и v_2 попарно смежны в G, и утверждение доказано.

Если в G никакие две из трех вершин v_1, v_2 и v_3 не смежны, тогда они попарно несмежны. Следовательно, по определению дополнения вершины v_1, v_2 и v_3 попарно смежны в $\overline{G}$, и утверждение снова доказано.

1.3. Маршруты, цепи, пути и циклы

Маршрут в графе $G=(V, E)$ представляет собой конечную чередующуюся последовательность вершин и ребер $v_0, e_1, v_1, e_2, \ldots$ $\ldots, v_{k-1}, e_k, v_k$, начинающуюся и кончающуюся на вершинах, причем v_{i-1} и v_i являются концевыми вершинами ребра e_i, $1 \leqslant i \leqslant k$. С другой стороны, маршрут можно рассматривать как конечную последовательность таких вершин $v_0, v_1, v_2, \ldots, v_k$, что (v_{i-1}, v_i), $1 \leqslant i \leqslant k$,— ребро графа G. Такой маршрут обычно называется v_0—v_k-маршрутом, а v_0 и v_k — *концевыми* или *терминальными вершинами* марш-

рута. Все другие вершины маршрута называются *внутренними*. Заметим, что ребра и вершины в маршруте могут появляться более одного раза.

Маршрут называется *открытым*, если его концевые вершины различны, в противном случае он называется *замкнутым*.

В графе на рис. 1.6 последовательность v_1, e_1, v_2, e_2, v_3, e_8, v_6, e_9, v_5, e_7, v_3, e_{11}, v_6 является открытым маршрутом, а последовательность v_1, e_1, v_2, e_2, v_3, e_7, v_5, e_3, v_2, e_4, v_4, e_5, v_1 — замкнутым.

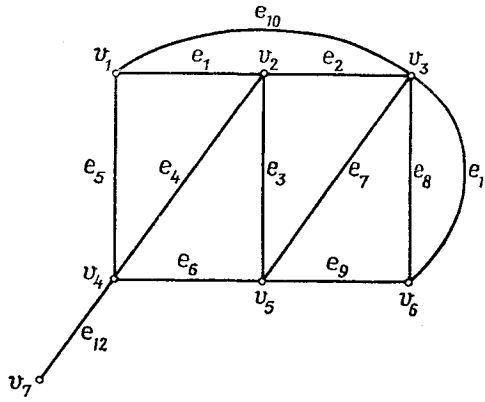


Рис. 1.6. Граф G.

Маршрут называется *цепью*, если все его ребра различны. Цепь называется *открытой*, если ее концевые вершины различны, в противном случае она называется *замкнутой*. На рис. 1.6 цепь v_1, e_1, v_2, e_2, v_3, e_8, v_6, e_{11}, v_3 — открытая, а v_1, e_1, v_2, e_2, v_3, e_7, v_5, e_3, v_2, e_4, v_4, e_5, v_1 — замкнутая.

Открытая цепь называется *путем*, если все ее вершины различны.

Замкнутая цепь называется *циклом*, если различны все ее вершины, за исключением концевых.

Например, на рис. 1.6 последовательность v_1, e_1, v_2, e_2, v_3 является путем, а последовательность v_1, e_1, v_2, e_3, v_5, e_6, v_4, e_5, v_1 — циклом.

Ребро графа G называется *циклическим*, если в графе G существует цикл, содержащий ребро. В противном случае ребро называется *нециклическим*. На рис. 1.6 все ребра, за исключением e_{12}, циклические.

Число ребер в пути называется *длиной пути*. Аналогично определяется *длина цикла*.

Необходимо указать следующие свойства путей и циклов.

1. Степень каждой неконцевой вершины пути равна 2, концевые вершины имеют степень, равную 1.

2. Каждая вершина цикла имеет степень 2 или другую четную степень. Обращение этого утверждения, а именно то, что ребра подграфа, в котором каждая вершина имеет четную степень, образуют цикл,— неверно. Более общий вопрос обсуждается в гл. 3.

3. Число вершин в пути на единицу больше числа ребер, тогда как в цикле число ребер равно числу вершин.

1.4. Связность и компоненты графа

Важным понятием в теории графов является связность. Две вершины v_i и v_j называются *связанными* в графе G, если в нем существует путь $v_i - v_j$. Вершина связана сама с собой.

Граф G называется *связным*, если в нем существует путь между каждой парой вершин. Например, граф, представленный на рис. 1.6, связный.

Рассмотрим несвязный граф $G=(V, E)$. Тогда множество вершин V графа G можно разбить [1] на такие подмножества $V_1, V_2, \ldots V_p$, что вершинно-порожденные подграфы $\langle V_i\rangle$, $i=1, 2, \ldots, p$, связны, и никакая вершина подмножества V_i не связана ни с какой

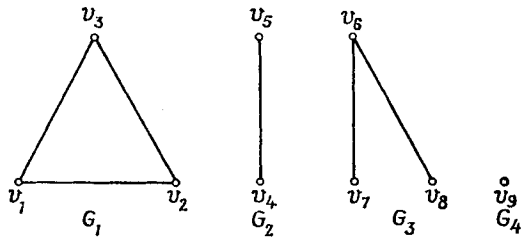


Рис. 1.7. Граф G с компонентами G_1, G_2, G_3 и G_4.

вершиной подмножества V_j, $j\neq i$. Подграфы $\langle V_i\rangle$, $i=1, 2, \ldots, p$, называются *компонентами* графа G. Легко видеть, что компонентой графа G является максимально связный подграф графа G, т. е. компонента графа G не является собственным подграфом любого другого связного подграфа графа G.

Например, граф G на рис. 1.7 не связен. Его четыре компоненты G_1, G_2, G_3, G_4 имеют множества вершин $\{v_1, v_2, v_3\}$, $\{v_4, v_5\}$, $\{v_6, v_7, v_8\}$, $\{v_9\}$ соответственно.

Отметим, что изолированную вершину также следует рассматривать как компоненту, поскольку по определению вершина связана сама с собой. Кроме того, следует отметить, что если граф G

---

[1] Говорят, что множество V *разбито* на подмножества $V_1, V_2, \ldots, V_p$, если $V_1 \cup V_2 \cup \ldots \cup V_p = V$ и $V_i \cap V_j = \varnothing$ для любого i и j, $i\neq j$. $\{V_1, V_2, \ldots, V_p\}$ называется *разбиением* V.

связен, то он имеет только одну компоненту, которая является графом G.

Теперь рассмотрим некоторые свойства связных графов.

Теорема 1.3. В связном графе любые два пути максимальной длины имеют общую вершину.

Доказательство. Рассмотрим произвольные пути максимальной длины P_1 и P_2 в связном графе G. Обозначим через P_1 последовательность вершин $v_0, v_1, v_2, \ldots$ $\ldots, v_k$, а через P_2 — последовательность $v_0', v_1', v_2', \ldots, v_k'$.

Предположим, что P_1 и P_2 не имеют общей вершины. Поскольку граф G связен, то для некоторых i, $0 \leqslant i \leqslant k$, и j, $0 \leqslant j \leqslant k$, существует $v_i - v_j'$-путь P_a, такой, что все вершины P_a, кроме v_i и v_j', отличны от вершин P_1 и P_2. Пути P_1, P_2 и P_a могут быть такими, как показано на рис. 1.8. Пусть t_1 — длина $v_0 - v_i$-пути P_{11}, t_2 — длина $v_i - v_k$-пути P_{12}, t_1' — длина $v_0' - v_j'$-пути P_{21}, t_2' — длина $v_j' - v_k'$-пути P_{22}, t_a — длина пути P_a. Пути P_{11}, P_{12}, P_{21} и P_{22} также представ-

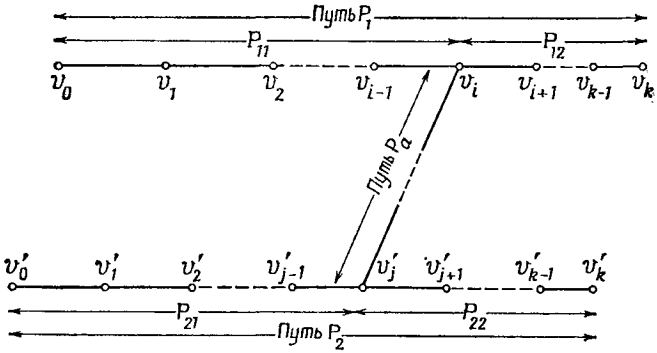


Рис. 1.8. Пути P_1, P_2 и P_a.

лены на рис. 1.8. Отметим, что $t_1+t_2=t_1'+t_2'$ — максимальная длина пути в графе G, а $t_a>0$. Не нарушая общности, предположим, что $t_1 \geqslant t_2$ и $t_1' \geqslant t_2'$, поэтому $t_1+t_1' \geqslant t_1+t_2=t_1'+t_2'$. Легко видеть, что пути P_{11}, P_a и P_{21} вместе составляют путь $v_0 - v_0'$, длина которого равна $t_1+t_1'+t_a>t_1+t_2$, так как $t_a>0$. Это противоречит тому, что t_1+t_2 является максимальной длиной пути в графе G.

Следующая теорема будет часто использоваться в обсуждениях гл. 2. В дальнейшем будем заменять $\{x\}$ на x всякий раз, когда ясно, что мы ссылаемся на множество, а не на его элемент.

Теорема 1.4. Если граф $G=(V, E)$ связен, то граф $G'=(V, E-e)$, получающийся после удаления циклического ребра e, тоже связен. Доказательство этой теоремы оставим в качестве упражнения.

1.5. Операции над графами

В этом разделе мы введем несколько операций над графами. Первые три операции, включающие два графа, бинарные, а остальные четыре — унарные, т. е. определены на одном графе.

Рассмотрим графы $G_1=(V_1, E_1)$ и $G_2=(V_2, E_2)$.

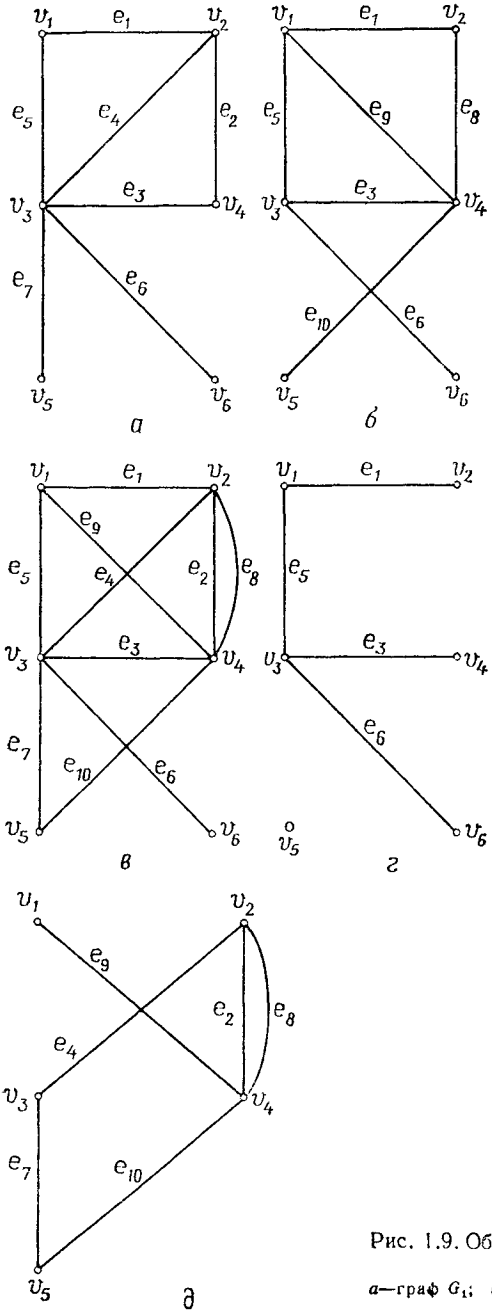


Рис. 1.9. Объединение, пересечение и кольцевая сумма графов.
а—граф G_1; *б*—граф G_2; *в*—$G_1 \cup G_2$; *г*—$G_1 \cap G_2$; *д*—$G_1 \oplus G_2$

Объединение графов G_1 и G_2, обозначаемое как $G_1 \cup G_2$, представляет собой такой граф $G_3 = (V_1 \cup V_2, E_1 \cup E_2)$, что множество его вершин является объединением V_1 и V_2, а множество ребер — объединением E_1 и E_2. Например, графы G_1 и G_2 и их объединение представлены на рис. 1.9, *а*, *б*, *в*.

Пересечение графов G_1 и G_2, обозначаемое как $G_1 \cap G_2$, представляет собой граф $G_3 = (V_1 \cap V_2, E_1 \cap E_2)$. Таким образом, множество вершин G_3 состоит только из вершин, присутствующих одновременно в графах G_1 и G_2, а множество ребер G_3 состоит только из ребер, присутствующих одновременно в G_1 и G_2. Пересечение графов G_1 и G_2 (рис. 1.9, *а*, *б*) показано на рис. 1.9, *г*. *Кольцевая сумма* двух графов G_1 и G_2, обозначаемая как $G_1 \oplus G_2$, представляет собой граф G_3, порожденный на множестве ребер $E_1 \oplus E_2$. Другими словами, граф G_3 не имеет изолированных вершин и состоит только из ребер, присутствующих либо в G_1, либо в G_2, но не в обоих графах одновременно. Кольцевая сумма графов (рис. 1.9, *а*, *б*) показана на рис. 1.9, *д*.

Легко убедиться в том, что три рассмотренные операции коммутативны, т. е. $G_1 \cup G_2 = G_2 \cup G_1$, $G_1 \cap G_2 = G_2 \cap G_1$, $G_1 \oplus G_2 = G_2 \oplus G_1$.

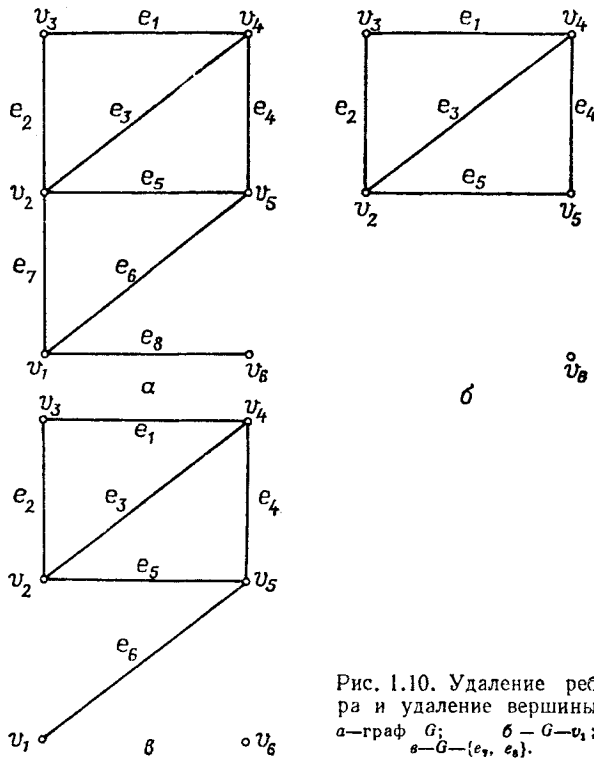


Рис. 1.10. Удаление ребра и удаление вершины.
а—граф G; *б* — $G - v_1$; *в*—$G - \{e_7, e_8\}$.

Заметим также, что эти операции бинарны, т. е. определены по отношению к двум графам. Очевидно, определение этих операций можно расширить на большее число графов.

Теперь рассмотрим унарные операции на графе.

Удаление вершины. Если v_i — вершина графа $G=(V, E)$, то $G - v_i$ — порожденный подграф графа G на множестве вершин $V - v_i$, т. е. $G—v_i$ является графом, получившимся после удаления из графа G вершины v_i и всех ребер, инцидентных этой вершине.

Удаление ребра. Если e_i — ребро графа $G=(V, E)$, то $G - e_i$ — подграф графа G, получающийся после удаления из G ребра e_i. Заметим, что концевые вершины ребра e_i не удаляются из G. Удаление из графа множества вершин или ребер определяется как последовательное удаление отдельных вершин или ребер.

Если $G_1 = (V', E')$ — подграф графа $G=(V, E)$, то через $G—G_1$ будем обозначать граф $G'=(V, E—E')$. Таким образом, $G—G_1$ — дополнение подграфа G_1 в G.

Удаление вершины и удаление ребра показано на рис. 1.10.

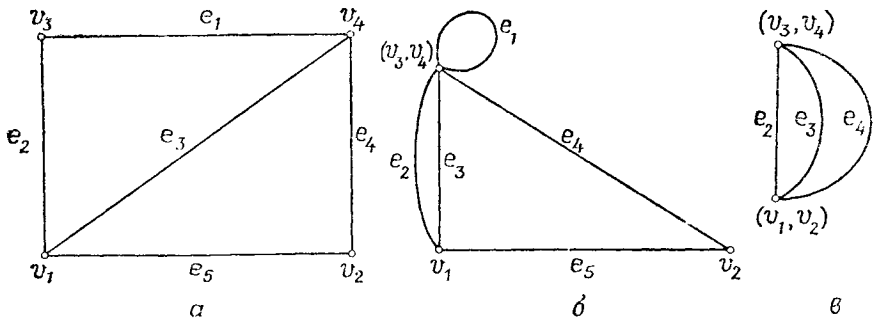
Рассматриваемая далее операция замыкания, или отождествления, хорошо знакома инженерам-электрикам.

Замыкание или отождествление. Говорят, что пара вершин v_i и v_j в графе G *замыкается* (или *отождествляется*), если они заменяются такой новой вершиной, что все ребра в графе G, инцидентные v_i и v_j, становятся инцидентными новой вершине.

Например, результат замыкания вершин v_3 и v_4 в графе рис. 1.11, *а* представлен на рис. 1.11, *б*.

Стягивание. Под стягиванием мы подразумеваем операцию удаления ребра e и отождествление его концевых вершин. Граф G является *стягиваемым* графом к графу H, если H можно получить из G последовательностью стягиваний.

Граф, изображенный на рис. 1.11, *в*, получен стягиванием ребер e_1 и e_5 в графе G (рис. 1.11, *а*).



а *б* *в*

Рис. 1.11. Операции отождествления и стягивания в графе.

а—граф G; *б*—граф G после отождествления v_3 и v_4; *в*—граф G после стягивания e_1 и e_5.

1.6. Специальные графы

В этом разделе мы рассмотрим специальные классы графов, часто встречающиеся в теории графов.

Полный граф G — простой граф, в котором каждая пара вершин смежна. Если полный граф G имеет n вершин, то он обозначается через K_n. Легко видеть, что K_n имеет $n(n-1)/2$ ребер. В качестве примера на рис. 1.12 представлен граф K_5.

Граф G называется *однородным*, если в нем все вершины имеют равную степень. Если граф G однороден и $d(v_i)=r$ для всех вершин v_i в G, то G называют r-однородным графом. 4-однородный граф

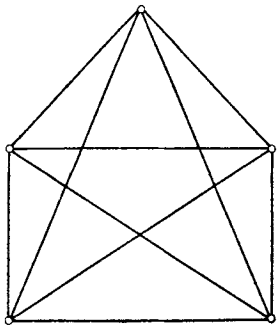


Рис. 1.12. Граф K_5.

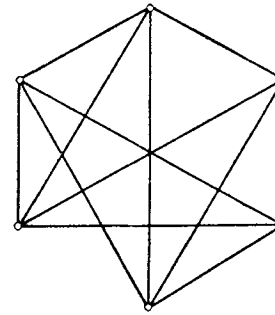


Рис. 1.13. 4-однородный граф.

представлен на рис. 1.13. Отметим, что K_n является $(n-1)$-однородным графом.

Граф $G=(V, E)$ называется *двудольным* графом, если множество его вершин V можно разбить на два таких подмножества V_1 и V_2, что каждое ребро, принадлежащее E, имеет одну концевую вершину в подмножестве V_1, а другую — в V_2; (V_1, V_2) называют *двудольным* разбиением G.

Если в простом двудольном графе G с разбиением (V_1, V_2) для каждой вершины v_i в V_1 и v_j в V_2 существует ребро (v_i, v_j), то G называют *полным двудольным* графом и обозначают через $K_{m,n}$, если V_1 содержит m вершин, а V_2 — n.

Двудольный граф и полный двудольный граф $K_{3,4}$ представлены на рис. 1.14.

Граф $G=(V, E)$ называется *k-дольным*, если V можно разбить на k таких подмножеств $V_1, V_2, \ldots, V_k$, что каждое ребро графа G имеет одну концевую вершину в некотором подмножестве V_i, а другую — в некотором подмножестве V_j, $i \neq j$. *Полный k-дольный граф* G — простой k-дольный граф с разбиением множества вершин

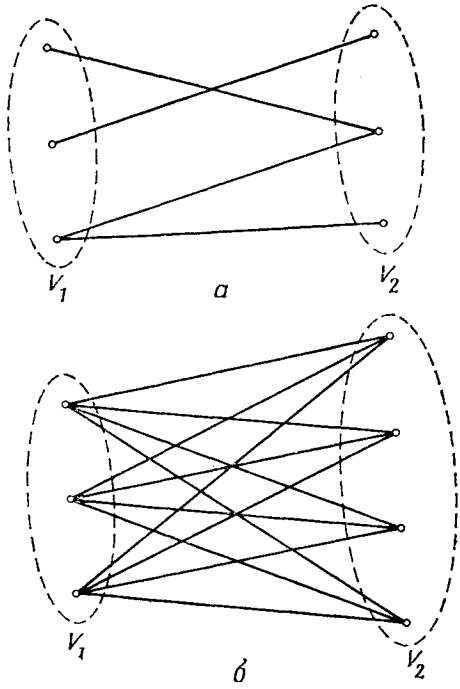


Рис. 1.14.
а—двудольный граф; *б*—полный двудольный граф $K_{5,4}$

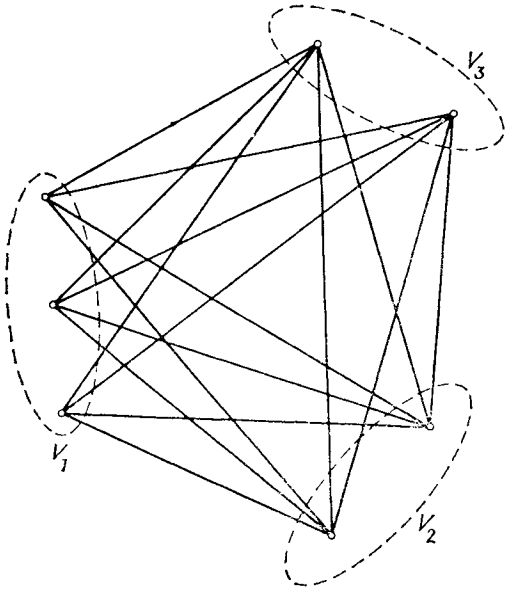


Рис. 1.15. Полный 3-дольный граф.

$\{V_1, V_2, \ldots, V_k\}$, обладающий таким свойством, что для каждой вершины v_i в V_r и v_j в V_s, $r \neq s$, $1 \leqslant r, s \leqslant k$, (v_i, v_j) является ребром графа G. Полный 3-дольный граф представлен на рис. 1.15.

1.7. Точки сочленения и разделимые графы

Вершина v_i графа G является *точкой сочленения* графа G, если граф $G-v_i$ состоит из большего числа компонент, чем G. Если граф G связный, то $G-v_i$ будет содержать по крайней мере две компоненты, т. е. $G-v_i$ будет несвязным. Согласно этому определению, изолированная вершина не может быть точкой сочленения.

Граф, имеющий одну вершину, будем называть *тривиальным*. Таким образом, тривиальный граф не имеет точек сочленения.

Неразделимый граф — это связный граф без точек сочленения. Все другие графы — *разделимые* (отметим, что несвязный граф также является разделимым графом).

Граф G, представленный на рис. 1.16, *а*, является разделимым. У него три точки сочленения: v_1, v_2 и v_3.

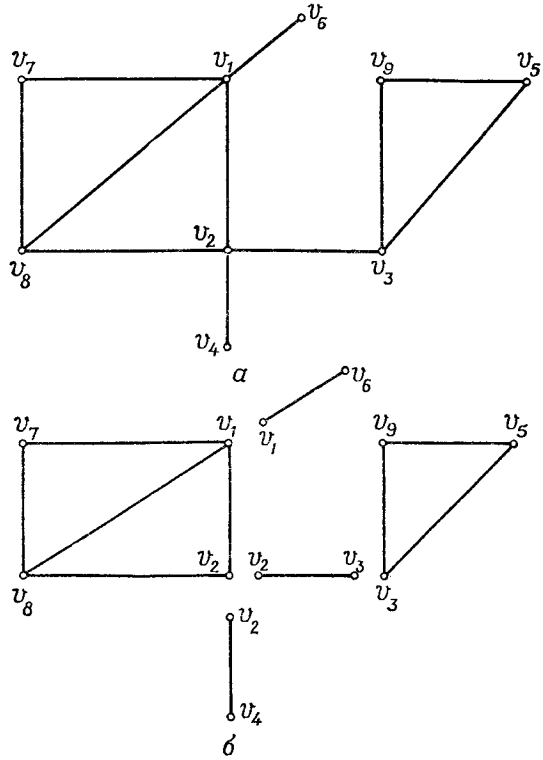


Рис. 1.16. Разделимый граф и его блоки.
а—разделимый граф G; *б*—блоки графа G.

Блоком разделимого графа G является максимальный неразделимый подграф графа G. Блоки разделимого графа G (рис. 1.16, *а*) изображены на рис. 1.16, *б*.

Следующая теорема дает эквивалентное введенному определение точки сочленения.

Теорема 1.5. Вершина v является точкой сочленения связного графа G тогда и только тогда, когда существуют такие две вершины u и w, отличные от v, что вершина v лежит на всяком u—w-пути.

Доказательство.

Необходимость: Если v — точка сочленения в графе G, тогда по определению граф G—v — несвязный. Пусть G_1 — одна из компонент G—v, и пусть V_1 — множество вершин G_1, а V_2 — дополнение V_1 в V—v. Пусть вершина u находится в V_1, а вершина w — в V_2. Рассмотрим в графе G произвольный u—w-путь. Если точка сочленения v не лежит на этом пути, то данный путь находится и в G—v, т. е. вершины u и w связаны в G—v. Однако это противоречит тому, что u и w находятся в разных компонентах G—v. Следовательно, вершина v лежит на любом u—w-пути.

Достаточность: Если v лежит на u—w-пути, то вершины u и w не связаны в G—v. Таким образом, граф G—v — несвязный. Следовательно, по определению v — точка сочленения.

Как следствие теоремы 1.5 можно дать эквивалентное определение разделимого графа:

Связный граф G является разделимым тогда и только тогда, когда в графе G существует такая вершина v, что она является единственной общей вершиной для двух собственных нетривиальных подграфов G_1 и G_2, объединение которых равно G.

Предположим, что граф G — разделимый. Возможно ли, чтобы все вершины графа G были точками сочленения? В следующей теореме мы докажем, что ответ на этот вопрос отрицателен.

Теорема 1.6. Любой нетривиальный связный граф содержит по крайней мере две вершины, которые не являются точками сочленения.

Доказательство. Докажем теорему индукцией по числу вершин графа. Теорема верна для любого связного графа с двумя вершинами (ни одна из этих вершин не является точкой сочленения). Предположим, что теорема верна для любого нетривиального связного графа, имеющего менее n вершин, где $n>2$. Пусть G — связный граф, имеющий n вершин. Если G не имеет ни одной точки сочленения, то теорема доказана.

Теперь предположим, что в графе G v есть точка сочленения. Далее, пусть G_1, G_2, . . ., G_k являются k-компонентами графа G—v.

Если какой-либо граф G_i является тривиальным, то его единственная вершина не является точкой сочленения. Рассмотрим произвольную нетривиальную компоненту G_j. По предположению индукции, G_j содержит вершины v_1 и v_2, которые не являются точками сочленения. Очевидно, что если одна из этих вершин не смежна с v в графе G, то эта вершина не является точкой сочленения в G. С другой стороны, если обе вершины v_1 и v_2 смежны с v в G, то ни v_1, ни v_2 не будут точками сочленения в G.

Таким образом, каждая компонента G—v имеет по крайней мере одну вершину, которая не является точкой сочленения в G.

Следовательно, граф G имеет по крайней мере две вершины, не являющиеся точками сочленения.

1.8. Изоморфизм и 2-изоморфизм

Графы на рис. 1.17 кажутся различными. Однако, если один из них перерисовать соответствующим образом, он будет очень похож на другой. Поэтому эти два графа в каком-то смысле «эквива-

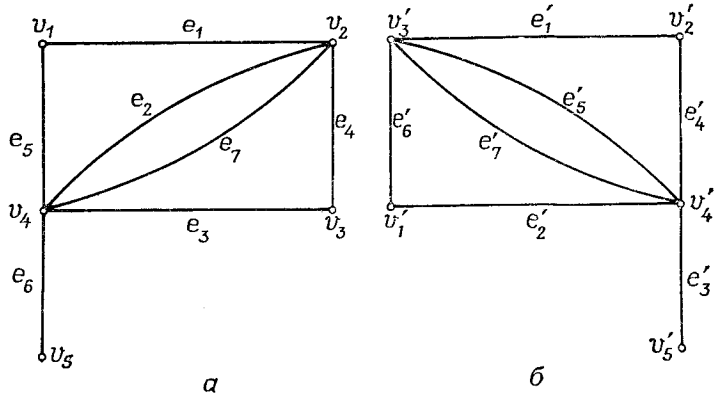


Рис. 1.17. Изоморфные графы.
$а$ — G_1; $б$ — G_2.

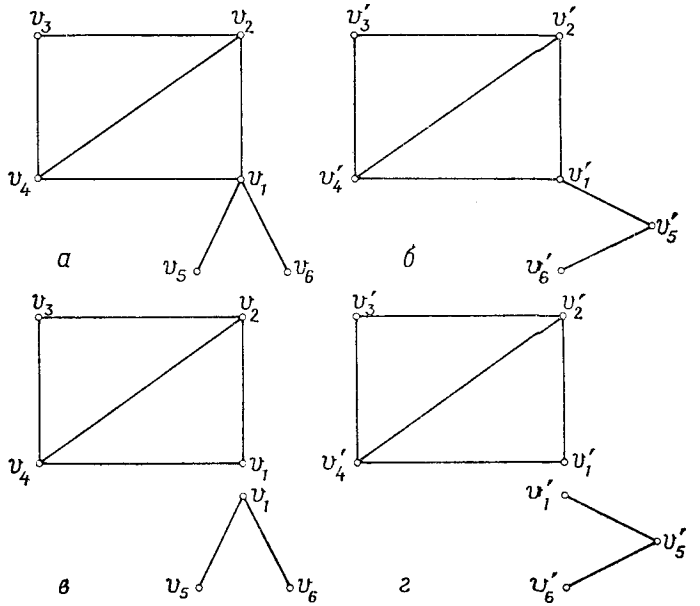


Рис. 1.18. 1-изоморфизм.
$а$ — G_1; $б$ — G_2; $в$ — граф после расщепления v_1 в G_1; $г$ — граф после расщепления v'_1 в G_2.

лентны». Определим эту эквивалентность более точно: говорят, что два графа G_1 и G_2 *изоморфны*, если существует такое взаимно-однозначное отображение между множествами их вершин и ребер, что соответствующие ребра графов G_1 и G_2 инцидентны соответствующим вершинам этих графов. Другими словами, если вершины v_1 и v_2 в G_1 соответствуют вершинам v_1' и v_2' в G_2, то ребро в G_1, имеющее концевые вершины v_1 и v_2, должно соответствовать ребру в G_2, имеющему концевые вершины v_1' и v_2', и наоборот.

Согласно данному определению графы, представленные на рис. рис. 1.17, изоморфны. Соответствие между множествами их вершин и ребер следующее:
соответствие вершин: $v_1 \leftrightarrow v_2'$, $v_2 \leftrightarrow v_3'$, $v_3 \leftrightarrow v_1'$, $v_4 \leftrightarrow v_4'$, $v_5 \leftrightarrow v_5'$;
соответствие ребер: $e_1 \leftrightarrow e_1'$, $e_3 \leftrightarrow e_2'$, $e_5 \leftrightarrow e_4'$, $e_2 \leftrightarrow e_5'$, $e_4 \leftrightarrow e_6'$, $e_6 \leftrightarrow e_3'$.

Рассмотрим разделимые графы G_1 и G_2, представленные на рис. 1.18, *а*, *б*. Эти графы не изоморфны. Предположим, что мы так «расщепили» точку сочленения v_1 в G_1 на две вершины, что получились два реберно-непересекающихся графа на рис. 1.18, *в*[1]. Если мы выполним подобное расщепление в точке сочленения v_1' в G_2, то получим два реберно-непересекающихся графа, приведенных на рис. 1.18, *г*. Легко убедиться, что графы на рис. 1.18, *в*, *г* изоморфны. Таким образом, графы G_1 и G_2 стали изоморфными после расщепления точек сочленения. Такие графы называются *1-изоморфными*.

Далее определим 2-изоморфизм как более общий тип изоморфизма. Два графа G_1 и G_2 являются 2-изоморфными, если они становятся изоморфными после применения следующих операций:
1. «*Расщепление*» точки сочленения в G_1 и (или) в G_2 на две вершины, такое, что получаются два реберно-непересекающихся графа.
2. Если один из графов, например G_1, состоит из двух подграфов G_1' и G_1'', имеющих точно две общие вершины v_1 и v_2, то выполнение перестановки этих вершин осуществляется в одном из подграфов. (Геометрически эта операция эквивалентна такому «*переворачиванию*» одного из подграфов G_1' и G_1'', что общие вершины в нем меняются местами.)

Рассмотрим графы G_1 и G_2 на рис. 1.19, *а*, *б*. После расщепления вершины v_2 в G_1 и выполнения операции «переворачивания» с переменой мест вершин v_1 и v_2 мы получим граф G_1', представленный на рис. 1.19, *в*. Расщепление вершины v_2' в G_2 приводит к графу G_2', изображенному на рис. 1.19, *г*. Графы G_1' и G_2' изоморфны. Следовательно, графы G_1 и G_2 являются 2-изоморфными.

Важный результат по 2-изоморфным графам формулируется в следующей теореме:

---

[1] Графы называются реберно-непересекающимися, если они не имеют общего ребра, и вершинно-непересекающимися, если не имеют общей вершины.

Теорема 1.7. Два графа G_1 и G_2 являются 2-изоморфными тогда и только тогда, когда существует взаимно-однозначное соответствие между множествами их ребер, такое, что циклы в одном графе соответствуют циклам в другом.

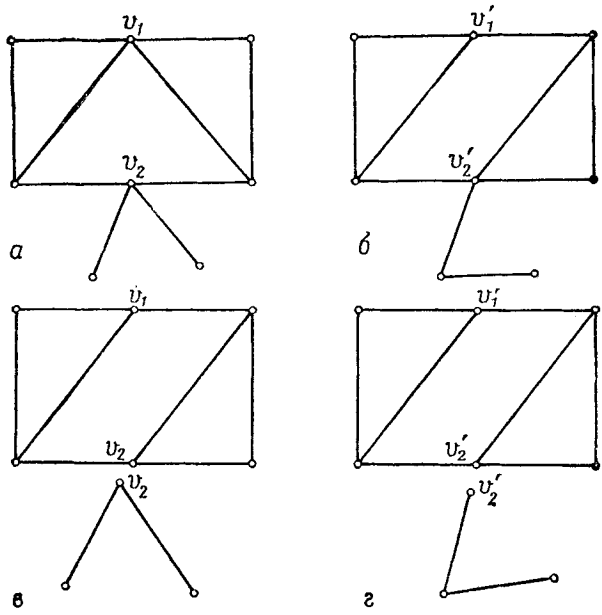


Рис. 1.19. 2-изоморфизм.
а — G_1; *б* — G_2; *в* — G_1'; *г* — G_2'.

Тот факт, что циклы в G_1 будут соответствовать циклам в G_2, когда G_1 и G_2 2-изоморфны, достаточно очевиден. Однако доказательство обратного слишком длинно, чтобы обсуждать его здесь. Оно приведено в оригинальной работе [1.1], посвященной 2-изоморфным графам.

1.9. Замечания, касающиеся литературы

Некоторые из тем, затронутых в этой части книги, рассматриваются в работах [1.2, 1.3]. В работе [1.2] рассматриваются также гиперграфы и матроиды. Хорошей работой, затрагивающей большое число неразрешенных проблем теории графов, является монография [1.4]. В работе [1.5] дается изящное введение в теорию графов и матроидов. Рекомендуются и другие книги по теории графов [1.6—1.8]. Полезны также книги и монографии, связанные с такими специальными темами, как перечисление графов, алгоритмы и т. д. Эти книги рекомендуются в соответствующих главах.

Упражнения

1.1. Пусть G — такой граф на n вершинах и m ребрах, что вершины имеют степень k или $k+1$. Докажите, что если G имеет n_k вершин степени k и n_{k+1} вершин степени $k+1$, то $n_k = (k+1)n-2m$.

1.2. Доказать или опровергнуть:
а) объединение любых двух различных замкнутых маршрутов, соединяющих две вершины, содержит цикл; б) объединение любых двух различных путей, соединяющих две вершины, содержит цикл.

1.3. Докажите, что если в графе G существуют пути между вершинами a и b, а также между b и c, то существует путь между a и c.

1.4. Пусть P_1 и P_2 — два различных пути между двумя вершинами графа. Докажите, что $P_1 \oplus P_2$ является циклом или объединением нескольких реберно-непересекающихся циклов в графе.

1.5. Докажите, что замкнутая цепь, все вершины которой имеют степень два, является циклом.

1.6. Покажите, что если два различных цикла графа G содержат ребро e, то в G существует цикл, не содержащий e.

1.7. Покажите, что в простом графе G с $\delta(G) \geqslant k$ существует путь по крайней мере длины k. Покажите также, что G имеет цикл длины по крайней мере $k+1$, если $k \geqslant 2$.

1.8. Докажите, что граф G является связным тогда и только тогда, когда для каждого разбиения (V_1, V_2) множества V с непустыми V_1 и V_2 существует ребро в G, соединяющее вершину из V_1 с вершиной из V_2.

1.9. Докажите, что простой граф G с n вершинами и k компонентами может иметь, самое большее, $(n-k)(n-k+1)/2$ ребер. Используя этот результат, докажите, что граф G должен быть связным, если он имеет больше $(n-1)(n-2)/2$ ребер.

1.10. Докажите, что если связный или несвязный граф G имеет точно две вершины нечетной степени, то должен существовать путь, соединяющий эти две вершины.

1.11. Докажите, что если простой граф G не является связным, то связно его дополнение $\overline{G}$.

1.12. Пусть G — такой граф на n вершинах и m ребрах, что $m < n-1$. Докажите, что G — несвязный граф.

1.13. Докажите, что если для графа G на n вершинах и m ребрах выполняется неравенство $m \geqslant n$, то G содержит циклическое ребро.

1.14. Докажите, что простой граф G на n вершинах является связным, если $\delta(G) \geqslant (n-1)/2$.

1.15. Покажите, что простой граф G, имеющий по крайней мере две вершины, содержит две вершины одинаковой степени.

1.16. Покажите, что если граф $G = (V, E)$ является простым и связным, но неполным, то он имеет такие три вершины u, v, w, что ребра (u, v) и (v, w) содержатся в E, а ребро (u, w) — не содержится.

1.17. Если вершины u и v связаны в графе G, то *расстояние* между u и v, обозначаемое через $d(u, v)$, является длиной наикратчайшего u—v-пути в G. Если в графе G не существует u—v-пути, то определяем расстояние $d(u, v)$ как бесконечное. Максимальное расстояние между двумя вершинами в G называется *диаметром* G. Покажите, что если G имеет диаметр >3, то $\overline{G}$ имеет диаметр <3.

1.18. *Обхватом* графа G является длина наикратчайшего цикла в G. Если G не имеет циклов, то определяем обхват графа G как бесконечный. Покажите, что k-однородный граф обхвата 4 имеет по крайней мере $2k$ вершин.

1.19. Простой граф G называется *самодополнительным*, если он изоморфен своему дополнению $\overline{G}$. Докажите, что число вершин самодополнительного графа должно равняться $4k$ или $4k+1$, где k — целое число.

1.20. Докажите, что простой граф на n вершинах не является двудольным, если он имеет более $n^2/4$ ребер.

1.21. Докажите, что граф является двудольным тогда и только тогда, когда все его циклы четной длины.

1.22. Постройте простой кубический граф на $2n$ ($n \geqslant 3$) вершинах, не имеющий треугольников.
Примечание: граф называется *кубическим*, если он 3-однородный; *треугольник* — это цикл длины 3.
1.23. Докажите, что если v — точка сочленения в простом графе G, то она не является точкой сочленения в $\overline{G}$.
1.24. Докажите, что приведенные ниже свойства графа G на $n \geqslant 3$ вершинах эквивалентны:
а) граф G является неразделимым;
б) любые две вершины в G входят в общий цикл;
в) для любой вершины v и любого ребра e графа G существует цикл, содержащий и v, и e;
г) любые два ребра в G входят в общий цикл;
д) пусть даны две вершины и одно ребро в графе G; тогда существует путь, который соединяет эти две вершины и содержит данное ребро;
е) для любых трех различных вершин в G найдется путь, соединяющий любые две из них и включающий третью;
ж) для любых трех различных вершин в G найдется путь, соединяющий любые две из них и не включающий третью.
1.25. Покажите, что если граф G не содержит циклов четной длины, то каждый блок в G является либо K_1, либо K_2, либо циклом нечетной длины.
1.26. Покажите, что связный граф, не являющийся блоком, имеет два блока с общей вершиной. Пусть $b(v)$ — число блоков графа $G=(V, E)$, включающих вершину v. Покажите, что число блоков графа G равно $p+\sum_{v \in V}(b(v)-1)$, где p — число компонент графа G.
1.27. Пусть $c(B)$ — число точек сочленения связного графа G, являющихся вершинами блока B. Тогда число $c(G)$ точек сочленения графа G определяется по формуле $c(G)-1=\sum_{\text{по всем блокам}}[c(B)-1]$.
1.28. *Мостом* графа G является такое ребро e, что $G-e$ имеет больше компонент, чем G. Докажите:

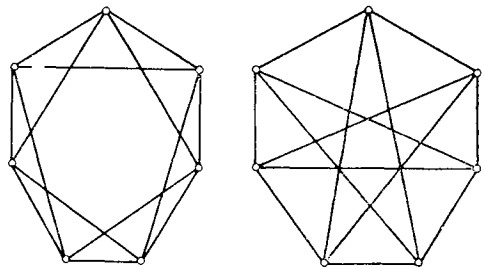


Рис. 1.20.

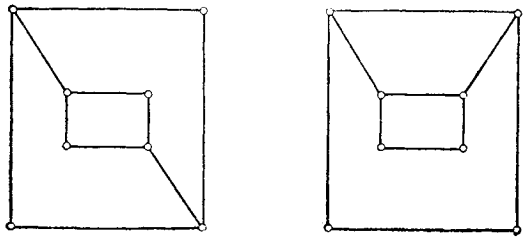


Рис. 1.21.

а) ребро e связного графа G является мостом тогда и только тогда, когда существуют такие вершины u и w, что e лежит на каждом u—w-пути в G;
б) ребро графа G является мостом в G тогда и только тогда, когда в графе G нет ни одного цикла, содержащего это ребро.

1.29. Являются ли графы, изображенные на рис. 1.20, изоморфными? Почему?

1.30. Покажите, что два графа на рис. 1.21 не изоморфны.

1.31. Определите все неизоморфные простые графы порядков 3 и 4.
Примечание: существует только 4 неизоморфных графа с тремя вершинами и 11 — с четырьмя вершинами.

1.32. Докажите, что любые два простых связных графа на n вершинах, каждая степени 2, являются изоморфными.

2. Деревья, разрезающие множества и циклы

Графы, встречающиеся в большинстве приложений, являются связными. Среди связных графов деревья, имеющие самую простую структуру, являются одними из наиболее важных. Однако если так важен связный граф, то и тогда множество ребер, разделяющих граф, имеет не менее важное значение. Таким образом, мы приходим к понятию «разрезающего множества». В этой главе мы ознакомимся с деревьями, разрезающими множествами и некоторыми результатами, связанными с ними, а также выявим связь между этими понятиями и циклами.

2.1. Деревья, остовы и кодеревья

Граф называется *ациклическим*, если он не содержит циклов. *Деревом* называется связный ациклический граф.
Деревом графа G называется связный ациклический подграф графа G. *Остов* графа G — это дерево графа G, содержащее все вершины G. Связный подграф дерева T называется *поддеревом* T. В качестве примера рассмотрим граф G, изображенный на рис. 2.1, *а*.
Графы G_1 и G_2 (рис. 2.1, *б*) являются деревьями графа G, графы G_3 и G_4 (рис. 2.1, *в*) — остовами графа G.
Кодерево T^* остова T графа G является подграфом графа G, содержащего все вершины G и только те ребра G, которые не входят в T. Следует отметить, что кодерево может быть несвязным. Кодеревья G_3^* и G_4^* остовов G_3 и G_4 (рис. 2.1, *в*) представлены на рис. 2.1, *г*. Ребра остова T называются *ветвями* T, а ребра соответствующего кодерева T^* — *хордами* или *связями*.

Остов T однозначно определяет свое кодерево T^*. Таким образом, мы определяем ребра T^* как хорды T. Рассмотрим некоторые свойства дерева. Хотя определение дерева как связного ациклического графа просто для понимания, существует еще несколько других определений дерева, которые рассматриваются в следующей теореме:

Теорема 2.1. Для графа G, имеющего n вершин и m ребер, следующие утверждения эквивалентны:
1. G является деревом. 2. Существует только один путь между любыми двумя вершинами в G. 3. G является связным и $m=n-1$. 4. G — ациклический граф и $m=n-1$. 5. G — ациклический граф, и при соединении ребром произвольных двух несмежных его вершин получается граф, имеющий точно один цикл.

Доказательство.
$1 \Rightarrow 2$. См. упражнение 1.2, *б*.

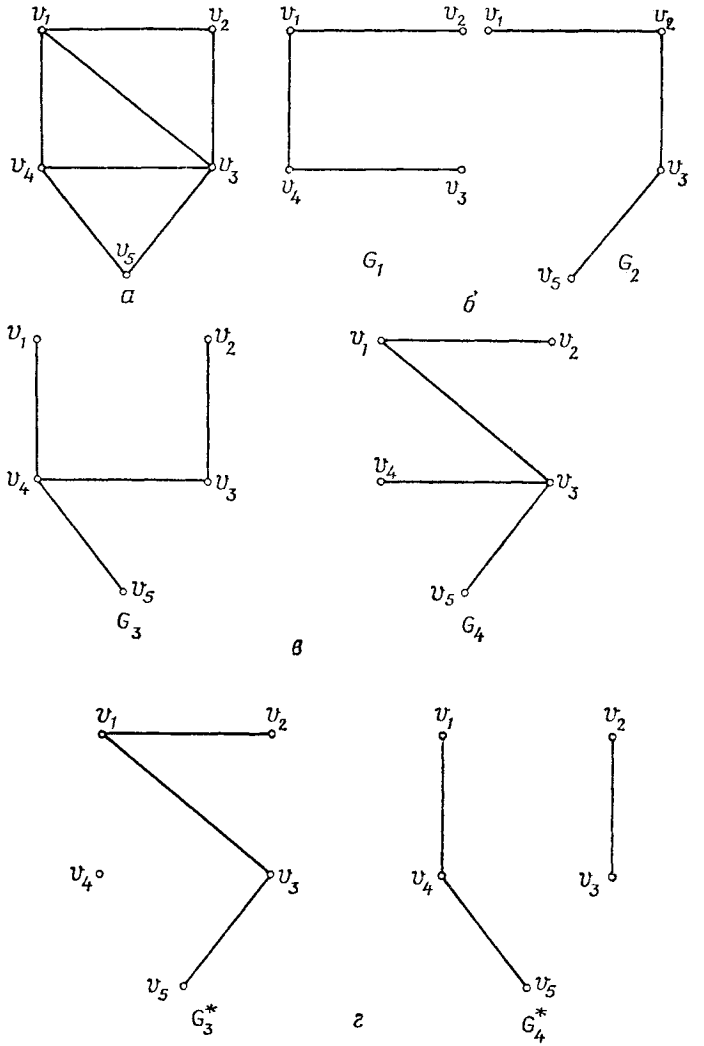


Рис. 2.1. Деревья, остовы и кодеревья.

а — граф G; *б* — деревья G_1 и G_2 графа G; *в* — остовы G_3 и G_4 графа G; *г* — кодеревья G_3^* и G_4^* графа G.

2⇒3. Для начала отметим, что G — связный граф, так как в нем существует путь между двумя любыми вершинами. Докажем, что $m=n-1$, методом индукции по числу вершин в G. Это очевидно для связного графа с одной или двумя вершинами. Предположим, что оно верно и для связного графа с числом вершин, меньшим n.

Рассмотрим произвольное ребро e в G. Это ребро составляет единственный путь между своими концевыми вершинами. Следовательно, в $G-e$ не существует пути между этими вершинами. Поэтому $G-e$ — несвязный граф. Далее, он должен содержать точно две компоненты, иначе граф G не будет связным. Пусть G_1 и G_2 — компоненты $G-e$. Пусть n_1 и m_1 — число вершин и ребер в G_1 соответственно. И пусть n_2 и m_2 аналогично определены для G_2. Тогда имеем $n=n_1+n_2$ и $m=m_1+m_2+1$. Отметим, что G_1 и G_2 удовлетворяют предположению индукции, т. е. что существует только один путь между двумя любыми вершинами в G_1 и G_2. Так как $n_1<n$ и $n_2<n$, то имеем по индуктивному предположению $m_1=n_1-1$ и $m_2=n_2-1$. Следовательно, $m=n_1-1+n_2-1+1=n-1$.

3$\Rightarrow$4. Пусть $G_0=G$. Предположим, что G_0 имеет несколько циклов. Пусть G_1 — граф, полученный после удаления из G_0 циклического ребра, например e_1, т. е. $G_1=G_0-e_1$. Поскольку G_0 — связный граф, то из теоремы 1.4 следует, что G_1 — также связный граф и содержит все n вершин, принадлежащих G_0. Количество ребер в G_1 равно $m-1$.

Если G_1 не является ациклическим, то пусть e_2 будет циклическим ребром в G_1. Вновь граф $G_2=G_1-e_2=G_0-e_1-e_2$ должен быть связным и иметь все n вершин G_1. G_2 имеет $m-2$ ребер. Если G_2 не является ациклическим графом, продолжим выполнение этой операции до тех пор, пока не получим связный ациклический граф G_p. Заметим, что G_p имеет n вершин и $m-p$ ребер.

Поскольку G_p является связным и ациклическим, он должен быть деревом. Следовательно, из предыдущего утверждения теоремы следует, что $m-p=n-1$. Так как по предположению $m=n-1$, то получаем, что $p=0$. Таким образом, граф $G=G_0$ — ациклический граф.

4$\Rightarrow$5. Пусть $G_1, G_2, \ldots, G_p$—p-компоненты графа G, n_i и m_i — число вершин и ребер в компоненте G_i соответственно. Тогда $m=m_1+m_2+\ldots+m_p$ и $n=n_1+n_2+\ldots+n_p$. Каждая компонента G_i является связным, а также ациклическим графом, поскольку этим свойством обладает граф G. Поэтому G_i является деревом. Далее, по утверждению 3 данной теоремы $m_i=n_i-1$ для любого $1\leqslant i\leqslant p$.

Поэтому имеем $m=\sum_{i=1}^{p} m_i=\sum_{i=1}^{p}(n_i-1)=n-p$, однако по предположению $m=n-1$.

Следовательно, в нашем случае $p=1$. Таким образом, G состоит точно из одной компоненты. Следовательно, он связный. Поскольку он также ациклический, то является деревом. Тогда по утверждению 2 теоремы существует единственный путь между двумя различными вершинами в G. Следовательно, если мы добавим в G ребро $e=(v_1, v_2)$, то это ребро вместе с единственным путем между v_1 и v_2 образует в полученном графе точно один цикл.

5$\Rightarrow$1. Предположим, что G не является связным. Рассмотрим любые две вершины v_a и v_b, которые содержатся в различных компонентах G. Тогда v_a и v_b не связаны в G. Добавление в G ребра (v_a, v_b) не образует цикла, так как в G нет пути между v_a и v_b. Однако это противоречит допущению. Следовательно, предположение о том, что G — несвязный граф, неверно. Таким образом, G является связным. А поскольку G еще и ациклический граф, то по определению он должен быть деревом.

Очевидно, что утверждения 1—5 в изложенной выше теореме представляют собой множество необходимых и достаточных условий для того, чтобы граф G был деревом.

Прямым следствием этой теоремы является следующее утверждение:

Следствие 2.1.1. Рассмотрим подграф G' графа G на n вершинах. Пусть G' имеет n вершин и m' ребер. Тогда следующие утверждения эквивалентны:
1. G' является остовом G. 2. Существует единственный путь между двумя любыми вершинами в G'. 3. G' является связным, и $m'=n-1$. 4. G' является ациклическим, и $m'=n-1$. 5. Если G' является ациклическим и любые две несмежные вершины G' соединить ребром, то получающийся граф имеет точно один цикл.
Сформулируем условие, которое не перекрывается предыдущим следствием, но доказывается проще:

Следствие 2.1.2. Подграф G' графа G на n вершинах является остовом графа G тогда и только тогда, когда G' ациклический, связный и имеет $n-1$ ребер.

Теперь должно быть очевидно, что подграф графа G на n вершинах, обладающий любыми тремя из следующих свойств, должен быть остовом G:
1) имеет n вершин, 2) связный, 3) имеет $n-1$ ребер, 4) ациклический.
Тогда возникает вопрос: достаточно ли двух из четырех перечисленных свойств, чтобы определить остов? На этот вопрос мы ответим позднее (см. также упражнение 2.3).

Теорема 2.2. Подграф G' графа G на n вершинах является остовом G тогда и только тогда, когда G' является ациклическим и имеет $n-1$ ребер.

Доказательство. Необходимость следует из утверждения 4 теоремы 2.1. Чтобы доказать достаточность, необходимо показать, что G' — связный граф и содержит n вершин G. Пусть G' состоит из p компонент, причем n_i — число вершин в компоненте G_i. Пусть n' — число вершин в G'. Тогда $n'=\sum_{i=1}^{p} n_i$. Каждый G_i является связным и ациклическим графом, поскольку G' — ациклический граф. Таким образом, каждая компонента G_i — дерево и, следовательно, имеет n_i-1 ребер, т. е. общее число ребер в G' равно $\sum_{i=1}^{p}(n_i-1)=n'-p$. Однако по предположению $n'-p=n-1$. Так как $n'\leqslant n$ и $p\geqslant 1$, ясно, что равенство выполняется тогда и только тогда, когда $n'=n$ и $p=1$. Таким образом, G' — связный граф, имеющий n вершин. А поскольку он еще и ациклический, то по определению является остовом G.

Предположим, что граф G имеет остов T. Тогда G связный потому, что он содержит связный подграф, включающий все его вершины. Докажем обратное, т. е. что связный граф имеет по крайней мере один остов.

Если связный граф G — ациклический, тогда он является своим собственным остовом. Если нет, то пусть e_1 — какое-либо циклическое ребро G. Тогда по теореме 1.4 граф $G_1=G-e_1$ — связный, содержащий все вершины, принадлежащие G. Если G_1 — не ациклический, то будем повторять эту операцию до тех пор, пока не получим связный, ациклический граф G_p, имеющий все вершины G. Тогда граф G_p будет остовом графа G. Результаты приведенного выше обсуждения суммируются в следующей теореме:

Теорема 2.3. Граф G является связным тогда и только тогда, когда он имеет остов.

Поскольку остов T графа G — ациклический, то каждый подграф остова T является ациклическим подграфом графа G. Будет ли тогда верно, что каждый ациклический подграф является подграфом некоторого остова G? Положительный ответ доказывается в следующей теореме:

Теорема 2.4. Подграф G' связного графа G является подграфом некоторого остова G тогда и только тогда, когда G' — ациклический.

Доказательство. Необходимость очевидна. Для доказательства достаточности пусть T — остов графа G. Рассмотрим граф $G_1=T\cup G'$. Ясно, что G' — подграф графа G_1. Граф G_1 — связный и содержит все вершины графа G, поскольку является подграфом графа G_1. Если G_1 — ациклический, то он является остовом графа G, а G' — его подграфом, и тогда теорема доказана. (Отметим, что, если G_1 — ациклический, $G_1=T$ и тогда G' — подграф T.) Предположим, что G_1 содержит цикл C_1. Из того, что G' — ациклический, следует, что не все ребра C_1 содержатся в G'. Таким образом, C_1 должен иметь по крайней мере одно ребро, например e_1, не содержащееся в G'. После удаления из G_1 циклического ребра e_1 получим связный граф $G_2=G_1-e_1$, содержащий все вершины G_1. Заметим, что G_1 является подграфом графа G_2. Если G_2 — ациклический граф, тогда он является требуемым остовом. Если нет, то будем повторять процесс до тех пор, пока не получим остов, подграфом которого является G'. Докажем интересную теорему о минимальном количестве висячих вершин дерева (т. е. вершин степени 1).

Теорема 2.5. В нетривиальном дереве существуют по крайней мере две висячие вершины.

Доказательство. Предположим, что дерево T имеет n вершин. Тогда по теореме 2.1 оно содержит $n-1$ ребер. По теореме 1.1 имеем $\sum_{i=1}^{n} d(v_i)=2\times$число ребер в T. Тогда $d(v_1)+d(v_2)+\ldots+d(v_n)=2n-2$. Это равенство будет выполняться только в том случае, если по крайней мере два члена в левой части будут равны 1, т. е. T имеет по крайней мере две висячие вершины.

2.2. k-деревья, остовные k-деревья, леса

k-Деревом называется ациклический граф, состоящий из k компонент. Очевидно, что каждая компонента k-дерева сама является деревом. Заметим, что 1-дерево совпадает с деревом.

Если k-дерево является остовным подграфом графа G, то оно называется *остовным k-деревом* графа G.

k-Кодеревом T^* остовного k-дерева T графа G является остовный подграф графа G, содержащий точно те ребра, которых нет в T.

Например, граф на рис. 2.2, *б* является 2-деревом графа G, представленного на рис. 2.2, *а*. Остовное 3-дерево T графа G и соответствующее 3-кодерево T^* показаны на рис. 2.2, *в*, *г*.

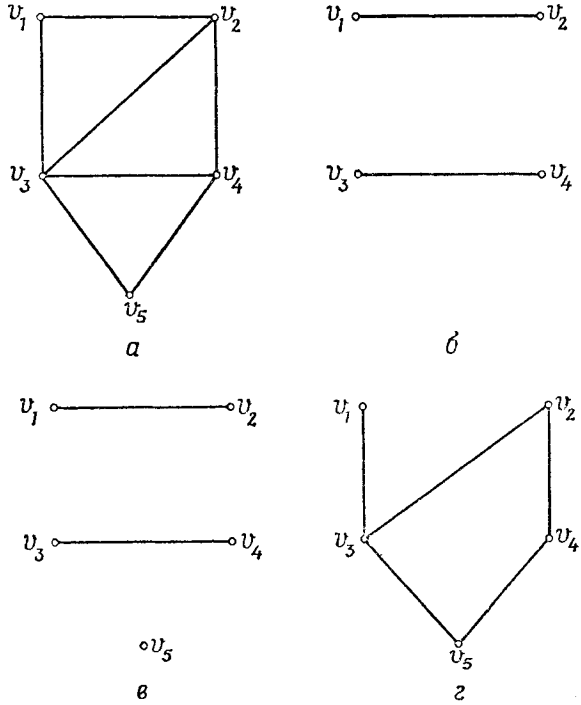


Рис. 2.2. Иллюстрация определения k-дерева, остовного k-дерева и k-кодерева. *а* — граф G; *б* — 2-дерево графа G; *в* — остовное 3-дерево T графа G; *г* — 3-кодерево T^*.

Обозначим k компонент остовного k-дерева графа G на n вершинах через $T_1, T_2, \ldots, T_k$. Если n_i — число вершин в T_i, то $n=n_1+n_2+\ldots+n_k$. Поскольку каждое T_i является деревом, то по теореме 2.1 имеем $m_i=n_i-1$, где m_i — число ребер в T_i. Таким образом, общее число ребер остовного k-дерева T равно $\sum_{i=1}^{k} m_i = \sum_{i=1}^{k} (n_i-1)=n-k$. Если m — число ребер в G, то k-кодерево T^* будет иметь $m-n+k$ ребер.

Лесом графа G называется остовное k-дерево графа G, где k — число компонент в G. Если граф G имеет p компонент, тогда для любого остовного k-дерева графа G $k \geqslant p$. Так как лесом T графа G является остовное k-дерево графа G с $k=p$, необходимо, чтобы каждая компонента T была остовом одной из компонент G. Таким образом, лес T графа G с p компонентами $G_1, G_2, \ldots, G_p$ состоит из p таких компонент $T_1, T_2, \ldots, T_p$, что T_i — есть остов G_i, $1 \leqslant i \leqslant p$.

Ко-лес T^* леса T графа G — это остовный подграф графа G, содержащий точно те ребра G, которые не входят в T.

Отметим, что понятия «лес» и «остов» являются синонимами по отношению к связному графу. Лес T и соответствующий ко-лес T^* графа представлены на рис. 2.3.

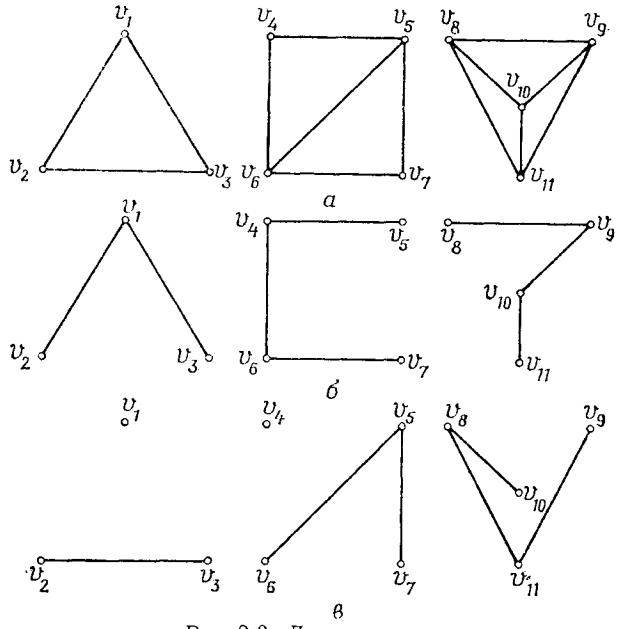


Рис. 2.3. Лес и ко-лес.
а — граф G; *б* — лес T графа G; *в* — ко-лес T^*.

2.3. Ранг и цикломатическое число

Рассмотрим граф G на n вершинах и m ребрах, содержащий k компонент. *Ранг графа* G, обозначаемый через $\rho(G)$, определяется как $n-k$. *Цикломатическое число графа* G, обозначаемое $\mu(G)$, определяется как $m-n+k$. Заметим, что $\rho(G)+\mu(G)=m$. Из определения леса и ко-леса следует, что ранг $\rho(G)$ графа равен числу ребер леса графа G, а цикломатическое число $\mu(G)$ — числу ребер ко-леса графа G. Ранг и цикломатическое число являются одними из наиболее важных характеристик графа. Как мы увидим в гл. 4, они определяют размерность подпространств циклов и разрезов графа.

2.4. Базисные циклы

Рассмотрим остов T связного графа G. Обозначим ветви T через $b_1, b_2, \ldots, b_{n-1}$, а хорды T — через $c_1, c_2, \ldots, c_{m-n+1}$, где m — число ребер, а n — число вершин в графе G. Поскольку T — ациклический, то по теореме 2.1 граф $T \cup c_i$ содержит точно один цикл C_i. Этот цикл состоит из хорды c_i и тех ветвей T, которые принадле-

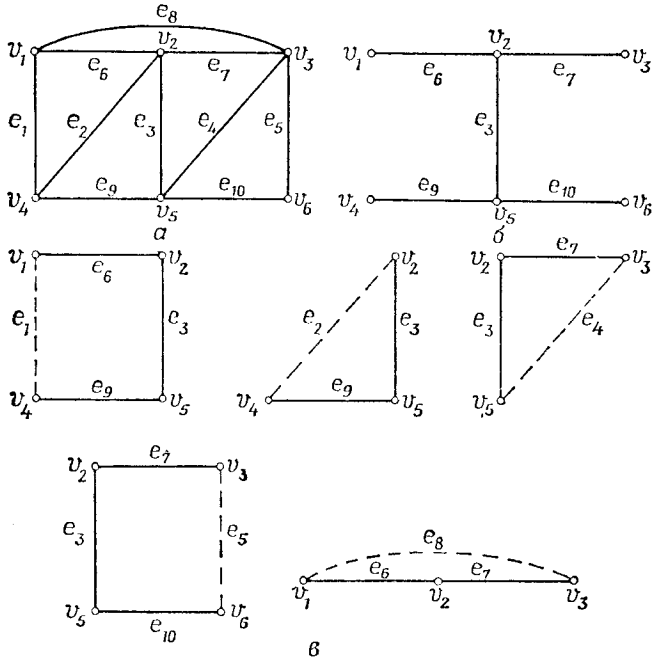


Рис. 2.4. Множество базисных циклов графа G.
а — граф G; *б* — остов T графа G; *в* — пять базисных циклов графа G по отношению к T (хорды обозначены штриховыми линиями).

жат единственному пути между концевыми вершинами c_i. Цикл C_i называется *базисным циклом графа* G относительно хорды c_i остова T.

Множество всех $m - n + 1$ базисных циклов $C_1, C_2, \ldots, C_{m-n+1}$ графа G относительно хорд остова T называется *базисным множеством циклов* графа G относительно T.

Важной особенностью базисного цикла C_i является то, что он содержит только одну хорду, т. е. хорду c_i. Далее, хорда c_i не присутствует ни в одном другом базисном цикле относительно T. Из этих свойств следует, что множество ребер базисного цикла нельзя выразить в виде кольцевой суммы множества ребер некоторых или всех оставшихся базисных циклов. В гл. 4 мы увидим, что каждый цикл графа G можно выразить в виде кольцевой суммы некоторых базисных циклов графа G по отношению к остову графа G. Именно поэтому такие циклы и названы «базисными».

Граф G и множество его базисных циклов представлены на рис. 2.4.

2.5. Разрезающие множества

К разрезающему множеству S связного графа G относится такое минимальное множество ребер графа G, что удаление их из графа G разделяет последний, т. е. граф $G—S$ становится несвязным.

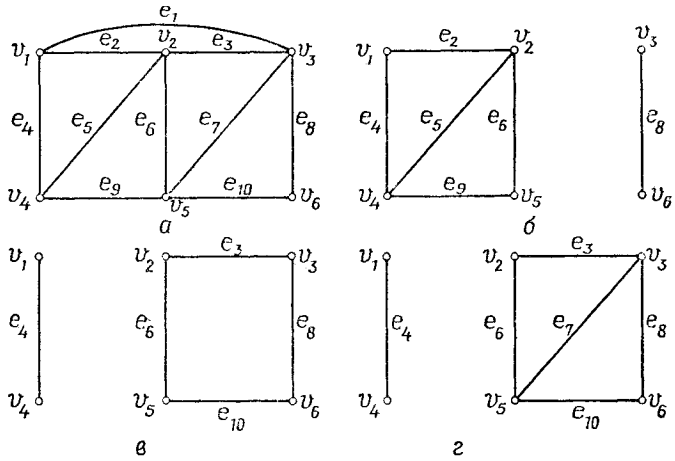


Рис. 2.5. Иллюстрация определения разрезающего множества.
а — граф G; *б* — $G_1 = G - S$, $S = \{e_1, e_3, e_7, e_{10}\}$; *в* — $G_2 = G - S_2$, $S_2 = \{e_1, e_2, e_3, e_7, e_9\}$; *г* — $G_3 = G - S_2'$, $S_2' = \{e_1, e_2, e_5, e_9\}$.

В качестве примера рассмотрим подмножество $S_1 = \{e_1, e_3, e_7, e_{10}\}$ ребер графа G (рис. 2.5, *а*). После удаления S_1 из G получаем $G_1 = G - S$ (рис. 2.5, *б*). Граф G_1 — несвязный. Кроме того, удаление любого собственного подмножества S_1 не может превратить

граф G в несвязный. Таким образом, S_1 — разрезающее множество графа G.

Рассмотрим теперь множество $S_2=\{e_1, e_2 e_5, e_7, e_9\}$. Граф $G_2=G-S_2$ (рис. 2.5, *в*) — несвязный. Однако множество $S_2'=\{e_1, e_2, e_5, e_9\}$, являющееся собственным подмножеством S_2, также превращает граф G в несвязный. Граф $G_3=G-S_2'$ показан на рис. 2.5, *г*. Следовательно, S_2 не будет разрезающим множеством графа.

Заметим, что по определению разрезающего множества, данного выше, если S является разрезающим множеством графа G, то ранги G и $G-S$ отличаются по крайней мере на единицу, т. е. $\rho(G)-\rho(G-S) \geqslant 1$. В работе [2.1] приводится следующее определение разрезающего множества:

Разрезающим множеством S связного графа G является такое минимальное множество ребер G, что удаление S' разбивает граф G на две компоненты, т. е. $\rho(G)-\rho(G-S)=1$. Возникает вопрос: эквивалентны ли эти два определения? Ответ — положителен, а доказательство оставлено в качестве упражнения 2.15.

2.6. Разрез

Определим понятие «разрез» которое связано с понятием «разрезающее множество».

Рассмотрим связный граф G с множеством вершин V. Пусть V_1 и V_2 — два таких непересекающихся подмножества множества V, что $V=V_1 \cup V_2$, т. е. V_1 и V_2 не имеют общих вершин, а вместе содержат все вершины множества V. Тогда множество S всех тех

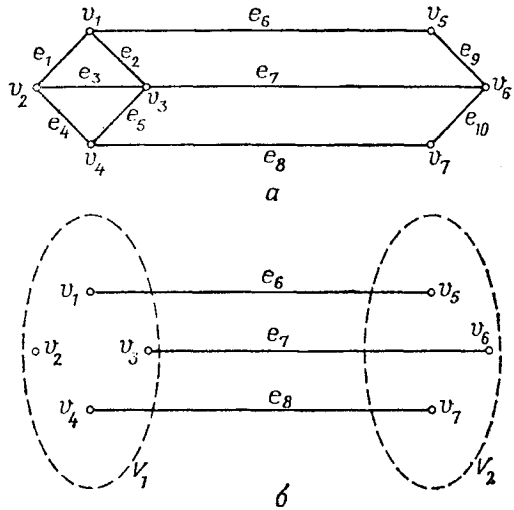


Рис. 2.6. Определение разреза.
а — граф G; *б* — разрез $\langle V_1, V_2\rangle$ графа G.

ребер, которые имеют одну вершину в V_1, а другую — в V_2, называется *разрезом* графа G. Разрез обычно обозначается через $<V_1, V_2>$. В работе [2.2] разрез называется *разделителем* (множество ребер, разделяющих множество вершин V).

В качестве примера рассмотрим граф G, изображенный на рис. 2.6. Если $V_1=\{v_1, v_2 v_3, v_4\}$ и $V_2=\{v_5, v_6, v_7\}$, то разрез $<V_1, V_2>$ — это множество ребер $\{e_6, e_7, e_8\}$.

Отметим, что разрез $<V_1, V_2>$ графа G является минимальным множеством ребер, удаление которых из графа G разбивает исходный граф на два графа G_1 и G_2, являющиеся порожденными подграфами на множествах вершин V_1 и V_2 соответственно. Графы G_1 и G_2 могут быть несвязными. Если оба эти графа связные, то $<V_1, V_2>$ по-прежнему минимальное множество ребер, разделяющих граф G точно на две компоненты. Тогда по определению $<V_1, V_2>$— есть разрезающее множество графа G.

Предположим, что для разрезающего множества S графа G величины V_1 и V_2 являются соответственно множествами вершин двух компонент G_1 и G_2 графа G—S. Тогда S есть разрез $<V_1, V_2>$. Таким образом, имеет место следующая теорема:

Теорема 2.6. 1. Разрез $\langle V_1, V_2\rangle$ связного графа G есть разрезающее множество графа G, если соответствующие подграфы G, порожденные на множествах вершин V_1 и V_2,— связные. 2. Если S — разрезающее множество связного графа G, а V_1 и V_2 — множества вершин двух компонент G—S, то $S=\langle V_1, V_2\rangle$.

Любой разрез $\langle V_1, V_2\rangle$ в связном графе G содержит разрезающее множество, так как удаление $\langle V_1, V_2\rangle$ из графа G переводит последний в несвязный граф. Фактически мы можем доказать, что разрез в графе G является объединением нескольких реберно-непересекающихся разрезающих множеств графа G. Формально мы утверждаем это в следующей теореме:

Теорема 2.7. Разрез в связном графе G есть объединение нескольких реберно-непересекающихся разрезающих множеств графа G. Доказательство этой теоремы несложно, поэтому оставим его в качестве упражнения 2.19.

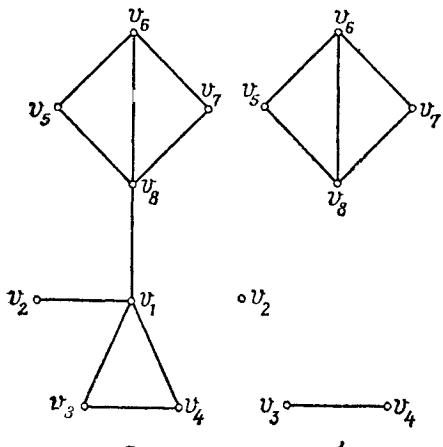


Рис. 2.7. Иллюстрация теоремы 2.8. *а* — граф G; *б* — подграф графа G, порожденный на множестве вершин $\{v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$.

Рассмотрим вершину v_1 в связном графе G. Множество ребер, инцидентных вершине v_1, образует разрез $\langle v_1, V-v_1\rangle$. Удаление этих ребер разбивает граф G на два подграфа. Один из них, содержащий только одну вершину v_1, является связным по определению. Другой является порожденным подграфом G' графа G на множестве вершин $V-v_1$. Поэтому разрез $\langle v_1, V-v_1\rangle$ является разрезающим множеством тогда и только тогда, когда подграф G' является связным. Но подграф G' является связным в том и только в том случае, когда v_1 не является точкой сочленения (разд. 1.7). Таким образом, имеем следующую теорему:

Теорема 2.8. Множество ребер, инцидентных вершине v в связном графе G, есть разрезающее множество тогда и только тогда, когда v не является точкой сочленения в графе G. В качестве примера рассмотрим разделимый граф G, представленный на рис. 2.7, *а*. v_1 — точка сочленения графа G. Подграф графа G, порожденный на множестве вершин $V-v_1=\{v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$, изображен на рис. 2.7, *б*. Этот подграф состоит из трех компонент и не связен. Поэтому ребра, инцидентные точке сочленения v_1, не образуют разрезающего множества графа G.

2.7. Базисные разрезающие множества

В разд. 2.4 было показано, что остов связного графа можно использовать для получения множества базисных циклов в графе. В данном разделе мы показываем, как с помощью остова можно определить множество базисных разрезающих множеств.

Рассмотрим остов T связного графа G. Пусть b — ветвь T. Удаление ветви b разбивает T на две компоненты T_1 и T_2. Следует отметить, что T_1 и T_2 являются деревьями графа G. Пусть V_1 и V_2 обозначают соответственно такие множества вершин T_1 и T_2, что V_1 и V_2 вместе содержат все вершины графа G.

Пусть G_1 и G_2 будут соответственно порожденными подграфами графа G на множествах вершин V_1 и V_2. Не трудно видеть, что T_1 и T_2 являются остовами G_1 и G_2. Следовательно, по теореме 2.3 G_1 и G_2 — связные графы. В теореме 2.6 доказывается, что разрез $\langle V_1, V_2\rangle$ является разрезающим множеством графа G. Это разрезающее множество называется *базисным множеством* графа G по отношению к ветви b остова T графа G. Множество всех $n-1$ базисных разрезающих множеств по отношению к n — 1-ветви остова T связного графа G называется *базисным множеством разрезающих множеств* графа G по отношению к остову T.

Следует отметить, что разрезающее множество $\langle V_1, V_2\rangle$ содержит точно одну ветвь, т. е. ветвь b остова T. Все другие ребра $\langle V_1, V_2\rangle$ являются хордами T. Это следует из того, что $\langle V_1, V_2\rangle$ не содержит ни одного ребра из остовов T_1 и T_2. Далее, ветвь b не присутствует ни в одном базисном разрезающем множестве по отношению к T. Из этого следует, что множество ребер базисного разрезающего множества нельзя представить в виде кольцевой суммы множеств ребер нескольких или всех базисных разрезающих множеств. В гл. 4 мы покажем, что каждое разрезающее множество

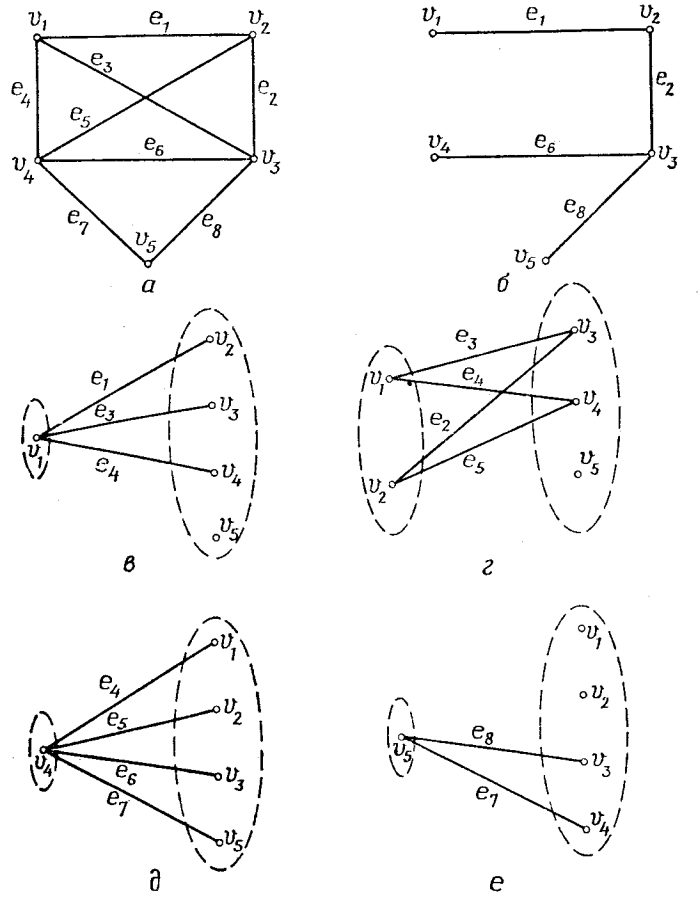


Рис. 2.8. Множество базисных разрезающих множеств графа.

а — граф G; *б* — остов T графа G; *в* — базисное разрезающее множество по отношению к ветви e_1; *г* — базисное разрезающее множество по отношению к ветви e_2; *д* — базисное разрезающее множество по отношению к ветви e_6; *е* — базисное разрезающее множество по отношению к ветви e_8.

графа G можно представить кольцевой суммой нескольких базисных разрезающих множеств по отношению к остову T графа G.

Граф G и множество его базисных разрезающих множеств представлены на рис. 2.8.

2.8. Остовы, циклы и разрезающие множества

В этом разделе мы обсудим некоторые интересные результаты, связывающие разрезающие множества и циклы с остовами и кодеревьями соответственно. Эти результаты выявляют двойственную

природу циклов и разрезающих множеств. Они приводят к альтернативным характеризациям разрезающих множеств и циклов в терминах остовов и кодеревьев соответственно.

Очевидно, что удаление разрезающего множества S из связного графа G разрушает все остовы графа G. Нетрудно показать, что разрезающее множество является минимальным множеством ребер, удаление которого из графа G вызывает разрушение всех остовов в графе. Однако результат, обратный этому, не очевиден. Первые теоремы данного параграфа затрагивают эти и некоторые другие вопросы, относящиеся к циклам.

Теорема 2.9. Разрезающее множество связного графа G содержит по крайней мере одну ветвь каждого остова графа G.

Доказательство. Предположим, что разрезающее множество S графа G не содержит ни одной ветви остова T графа G. Тогда граф $G—S$ будет содержать остов T и, следовательно, по теореме 2.3 $G—S$ — связный граф. Но это противоречит тому, что S — разрезающее множество графа S.

Теорема 2.10. Цикл связного графа G содержит по крайней мере одно ребро каждого кодерева графа G.

Доказательство. Пусть цикл C графа G не содержит ни одного ребра кодерева T^* остова T графа G. Тогда граф $G—T^*$ будет содержать цикл C. Так как $G—T^*$ совпадает с остовом T, то, следовательно, остов T содержит цикл. Но это противоречит определению остова.

Теорема 2.11. Множество ребер S связного графа G является разрезающим множеством G тогда и только тогда, когда S — минимальное множество ребер, содержащих по крайней мере одну ветвь каждого остова графа G.

Доказательство.

Необходимость. Если множество ребер S графа G является разрезающим множеством, то по теореме 2.9 оно содержит по крайней мере одну ветвь каждого остова графа G. Если оно не является таким минимальным множеством, то собственное подмножество S' множества S будет содержать по крайней мере одну ветвь каждого остова графа G. Тогда $G—S'$ не будет содержать ни одного остова графа и будет несвязным. Таким образом, удаление собственного подмножества S' разрезающего множества S графа G превращает последний в несвязный граф. Однако это будет противоречить определению разрезающего множества. Следовательно, необходимость доказана.

Достаточность. Пусть S — минимальное множество ребер, содержащее по крайней мере одну ветвь каждого остова графа G. Тогда граф $G—S$ не будет содержать ни одного остова и, следовательно, будет несвязным. Предположим, что S не является разрезающим множеством. Тогда собственное подмножество S' множества S будет разрезающим. Следовательно, из доказательства необходимости этой теоремы S' будет минимальным множеством ребер, содержащим по крайней мере одну ветвь каждого остова графа G. Но это противоречит тому, что S — минимальное такое множество. Следовательно, достаточность доказана.

Эта теорема характеризует разрезающее множество в терминах остовов. Сформулируем подобную характеризацию для циклов с использованием кодеревьев.

Рассмотрим множество ребер C, образующее цикл в графе G. По теореме 2.10 множество C содержит по крайней мере одно ребро каждого кодерева графа G. Покажем, что никакое собственное подмножество C' множества C не обладает этим свойством. Очевидно, что C' не содержит цикл. Тогда по теореме 2.4 можно по-

строить остов T, содержащий подмножество C'. Кодерево T^*, соответствующее T, не имеет общих ребер с C'. Следовательно, для каждого собственного подмножества C' множества C существует по меньшей мере одно кодерево T^*, не имеющее общих ребер с подмножеством C'. В самом деле, это утверждение верно для каждого ациклического подграфа. Таким образом, имеем следующую теорему:

Теорема 2.12. Циклом связного графа G является минимальное множество ребер графа, содержащее по крайней мере одно ребро каждого кодерева графа G. Докажем обратную теорему.

Теорема 2.13. Множество ребер C связного графа G есть цикл в G, если оно является минимальным множеством, содержащим по крайней мере одно ребро каждого кодерева графа G.

Доказательство. Как было показано ранее, множество C не может быть ациклическим, так как для каждого ациклического подграфа G' графа G существует кодерево, не имеющее ни одного общего ребра с G'. Таким образом, множество C содержит по крайней мере один цикл C'. Предположим, что C' является собственным подмножеством множества C. Тогда по теореме 2.12 C' — минимальное множество ребер, содержащее по меньшей мере одно ребро каждого кодерева графа. Однако это противоречит предположению, что C — такое минимальное множество. Следовательно, ни одно собственное подмножество множества C не является циклом. Поскольку множество C не ациклическое, то оно должно быть циклом.

Теоремы 2.12 и 2.13 устанавливают, что множество ребер C связного графа G будет циклом тогда и только тогда, когда оно является минимальным множеством ребер, содержащим по крайней мере одно ребро каждого кодерева графа G. Новые характеризации разрезающего множества и цикла, полученные с помощью теорем 2.11—2.13, выявляют двойственную природу понятий «цикл» и «разрезающее множество». Эту двойственность мы подробнее исследуем в гл. 10 при обсуждении теории матроидов.

Следующая теорема связывает циклы и разрезающие множества без использования понятия «дерево».

Теорема 2.14. Цикл и разрезающее множество связного графа имеют четное число общих ребер.

Доказательство. Пусть C — цикл, а S — разрезающее множество связного графа G, а V_1 и V_2 — множества вершин двух связных подграфов G_1 и G_2 графа $G-S$.

Если C — подграф графа G_1 или G_2, то, очевидно, число ребер, общих в C и S, равно нулю, т. е. четному числу.

Предположим, что C и S имеют несколько общих ребер. Будем передвигаться по циклу C из вершины v_1, принадлежащей множеству V_1.

Поскольку передвижение должно кончиться в вершине v_1, необходимо, чтобы каждый раз мы встречались с ребром множества S, ведущим нас от вершины в множестве V_1 к вершине в множестве V_2; также должно существовать ребро S, переводящее нас из вершины в множестве V_2 назад, к вершине в множестве V_1. Это возможно только в том случае, если C и S имеют четное число общих ребер.

Мы доказали очень важную теорему. Она формулирует соотношение ортогональности между разрезающими множествами и циклами. Это соотношение будет позднее рассмотрено в гл. 4.

Следует отметить, что теорема, обратная теореме 2.14, не совсем верна. Однако в гл. 4 мы покажем, что множество S графа G является разрезающим множеством (циклом) или объединением

нескольких реберно-непересекающихся разрезающих множеств (циклов) тогда и только тогда, когда множество S имеет четное число ребер, входящих в пересечение с каждым циклом (разрезающим множеством).

Базисные циклы и разрезающие множества связного графа были определены относительно остова. И не удивительно, что базисные циклы и разрезающие множества связаны между собой следующим образом:

Теорема 2.15. 1. Базисный цикл по отношению к хорде остова T связного графа состоит точно из тех ветвей T, которым соответствуют базисные разрезающие множества, включающие эту хорду. 2. Базисное разрезающее множество по отношению к ветви остова T связного графа состоит точно из тех хорд остова T, которым соответствуют базисные циклы, включающие эту ветвь.

Доказательство. 1. Пусть C — базисный цикл связного графа G по отношению к хорде c_1 остова T графа G. Пусть цикл C содержит помимо хорды c_1 ветви $b_1, b_2, \ldots, b_k$, принадлежащие остову T.

Предположим, что S_i $(1 \leqslant i \leqslant k)$ — базисное разрезающее множество графа G по отношению к ветви b_i $(1 \leqslant i \leqslant k)$ остова T. Ветвь b_i — единственная общая ветвь как для C, так и для S_i. Хорда c_i — единственная хорда в цикле C. Поскольку C и S_i должны иметь четное число общих ребер, то необходимо, чтобы базисное разрезающее множество S_i включало в себя хорду c_1. Покажем, что никакое другое базисное разрезающее множество остова T не содержит хорды c_1.

Предположим, что базисное разрезающее множество S_{k+1} по отношению к некоторой ветви b_{k+1} остова T содержит хорду c_i. Тогда она будет единственным общим ребром между C и S_{k+1}. Однако это противоречит теореме 2.14. Таким образом, хорда c_1 присутствует только в разрезающем множестве, определяемом ветвями $b_1, b_2, \ldots, b_k$.

2. Доказывается аналогично.

2.9. Замечения, касающиеся литературы

Понятие «дерево» является основным при получении некоторых результатов, связанных с циклами и разрезающими множествами графов. Оказывается, что остовы, циклы и разрезающие множества являются соответственно базами, циклами и коциклами матроида, которые можно определить на множестве ребер графа.

Таким образом, результаты этой главы помогут лучшему пониманию теории матроидов, изложенной в гл. 10.

Теория электрических цепей относится к одной из самых ранних областей применения теории графов. Гл. 2, 4 и 6 образуют основу теоретического изучения электрических цепей с помощью графов. Наиболее полно с этим вопросом можно ознакомиться в работе [2.1]; работы [2.3—2.5] рекомендуются для дальнейшего изучения данной темы.

Некоторые вопросы, связанные с деревьями широко, обсуждаются в литературе. Часть из них рассмотрим в последующих главах книги.

Упражнения

2.1. Покажите, что на шести вершинах существует точно шесть неизоморфных деревьев, а на семи вершинах — 11. Нарисуйте все деревья.
2.2. Покажите, что дерево является двудольным графом.
2.3. Рассмотрим подграф G_s связного графа G на n вершинах. Покажите, что, за исключением пары *б* и *г*, ни одна другая пара из следующих условий не влечет того, что G_s является остовом G:
а) G_s содержит n вершин;
б) G_s содержит $n-1$ ребер;
в) G_s — связный подграф;
г) G_s не содержит циклов.
2.4. Докажите, что любое висячее ребро (ребро, инцидентное висячей вершине) связного графа G содержится в каждом его остове.
2.5. Докажите, что каждое ребро связного графа G является ветвью какого-либо остова этого графа.
2.6. Докажите, что в дереве каждая вершина, степень которой больше единицы, является точкой сочленения.
2.7. Докажите, что каждое ребро неразделимого графа G является хордой некоторого кодерева графа G.
2.8. Докажите или опровергните: любые два ребра неразделимого графа содержатся в некотором базисном цикле.
2.9. При каких условиях любые два ребра графа G могут быть хордами некоторого кодерева графа G?
2.10. Докажите, что неразделимый граф имеет цикломатическое число, равное единице, в том и только в том случае, если он является циклом.
2.11. Покажите, что цикломатическое число любого графа не может быть отрицательным. Приведите пример графа, цикломатическое число которого равно 0.
2.12. Рассмотрите следующие две операции над графами:
а) Если вершине v_a инцидентны только два ребра $e_1=(v_1, v_a)$ и $e_2=(v_2, v_a)$, то заменить e_1 и e_2 единственным ребром, соединяющим v_1 и v_2.
б) Заменить любое ребро (v_1, v_2) двумя ребрами (v_1, v_a) и (v_2, v_a), где v_a — новая вершина, не входящая в граф G. Докажите, что цикломатическое число графа является инвариантным относительно этих операций.
2.13. Связный граф G является *минимально-связным*, если для каждого ребра e графа G граф $G-e$ — несвязный. Докажите, что связный граф является деревом тогда и только тогда, когда он минимально-связный.
2.14. Докажите, что подграф G_s связного графа G является его остовом тогда и только тогда, когда он является максимальным подграфом графа G, не содержащим ни одного цикла.
2.15. Покажите, что разрезающее множество связного графа G является таким минимальным множеством S ребер графа G, что удаление S разбивает граф G точно на две компоненты, т. е. $\rho(G)-\rho(G-S)=1$.
2.16. Докажите, что каждый связный граф содержит разрезающее множество.
2.17. Докажите, что подграф связного графа G может содержаться в кодереве графа G тогда и только тогда, когда он не включает в себя ни одного разрезающего множества графа G.
2.18. Докажите, что подмножество S ребер связного графа G образует кодерево графа G тогда и только тогда, когда оно является максимальным подмножеством ребер, не содержащим ни одного разрезающего множества графа G.
2.19. Докажите, что разрез в связном графе G является разрезающим множеством или объединением нескольких реберно-непересекающихся разрезающих множеств графа G.
2.20. Докажите, что каждое разрезающее множество неразделимого графа более чем с двумя вершинами содержит по крайней мере два ребра.
2.21. Докажите, что граф G является неразделимым тогда и только тогда, когда любые два ребра графа G принадлежат общему разрезающему множеству.

2.22. Пусть C — цикл в графе G, а a и b — любые ребра цикла C. Докажите, что существует такое разрезающее множество S, что $S \cap C = \{a, b\}$.

2.23. Пусть T_1 и T_2 — остовы связного графа G. Покажите, что если e — произвольное ребро остова T_1, то существует ребро f в остове T_2 с таким свойством, что $(T_1 - e) \cup f$ (граф, полученный из T_1 после замены e на f) тоже является остовом графа G. Покажите также, что остов T_1 можно «трансформировать» в остов T_2 с помощью замены ребер остова T_1 на ребра остова T_2 по одному таким образом, что на каждом шаге замены мы получаем остов графа G.

2.24. а) Пусть C_1 и C_2 — два цикла графа G. Пусть e_1 — ребро, принадлежащее циклам C_1 и C_2, а e_2 — ребро, принадлежащее только циклу C_1. Докажите, что существует такой цикл C_3, что $C_3 \subseteq (C_1 \cup C_2) - e_1$, а e_2 принадлежит циклу C_3. б) Приведите доказательство для того случая, когда в п. а) термин «цикл» заменен термином «разрезающее множество».

Примечание. Это утверждение является одним из постулатов, использованных в работе [2.6] при определении «циклов» матронда (гл. 10).

2.25. Пусть T — произвольное дерево на $k+1$ вершинах. Покажите, что если G — простой граф и $\delta(G) \geqslant k$, то граф G имеет подграф, изоморфный дереву T.

2.26. Покажите, что граф G содержит k реберно-непересекающихся остовов тогда и только тогда, когда для каждого разбиения $(V_1, V_2, \ldots, V_l)$ множества V число ребер, имеющих концевые вершины в разных частях разбиения, равно по крайней мере $k(l-1)$ [2.7, 2.8].

2.27. *Центральная вершина графа* $G = (V, E)$ — это такая вершина u, что значение $\max_{v \in V} \{d(u, v)\}$ является наименьшим из возможных, где $d(u, v)$ — расстояние между u и v. Покажите, что дерево имеет либо только одну центральную вершину, либо две смежные центральные вершины.

2.28. *Граф деревьев* связного графа G на n вершинах — это граф, вершинами которого являются остовы $T_1, T_2, \ldots, T_r$ графа G, причем остовы T_i и T_j смежны тогда и только тогда, когда они имеют точно $n-2$ общих ребра. Покажите, что граф деревьев любого связного графа также является связным.

Примечание: см. упр. 2.23.

3. Эйлеровы и гамильтоновы графы

Многие открытия теории графов были использованы для решения «практических» проблем — задач, головоломок, игр и т. д. К одной из этих задач относится знаменитая задача о кенигсбергских мостах. Ее можно сформулировать следующим образом:

На реке Преголя в Кенигсберге (теперь г. Калининград) было два острова. Они соединялись между собой и с берегами реки семью мостами, как показано на рис. 3.1, *а*. Задача заключалась в том, чтобы, начав двигаться с одного из четырех участков суши (помеченных на рис. 3.1, *а* буквами A, B, C и D), только один раз пройти по каждому мосту и снова вернуться в исходную точку, т. е.

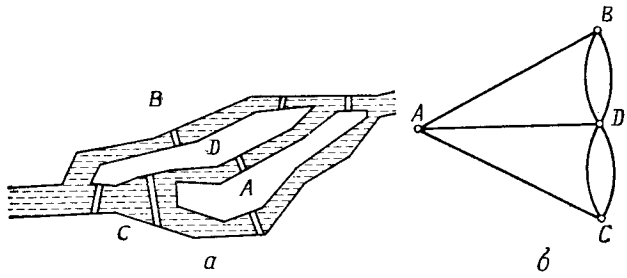


Рис. 3.1.

а — задача о кенигсбергских мостах; *б* — граф задачи о кенигсбергских мостах.

определить замкнутый маршрут через все семь мостов, дважды не пересекая ни один из них.

Многие были убеждены, что у этой задачи нет решения. Однако лишь в 1736 г. великий швейцарский математик Эйлер доказал возможность решения задачи и тем самым заложил основу теории графов. Эйлер был первым, кто показал, что эта задача эквивалентна созданию замкнутой цепи вдоль ребер графа (рис. 3.1, *б*), в котором вершины A, B, C и D представляют собой участки суши, а ребра — мосты, соединяющие эти участки. Затем он обобщил эту задачу и охарактеризовал ее с точки зрения графов, в которых существует такая замкнутая цепь. Такие графы стали называться эйлеровыми. В разд. 3.1 мы рассмотрим эти графы.

В 1859 г. другой известный математик — сэр Уильям Гамильтон — придумал игру, в которой требуется обойти замкнутый кон-

тур всех ребер додекаэдра, минуя каждую вершину лишь один раз. В теории графов эта игра эквивалентна определению остовного цикла, содержащего все 20 вершин (рис. 3.2). Можно проверить, что

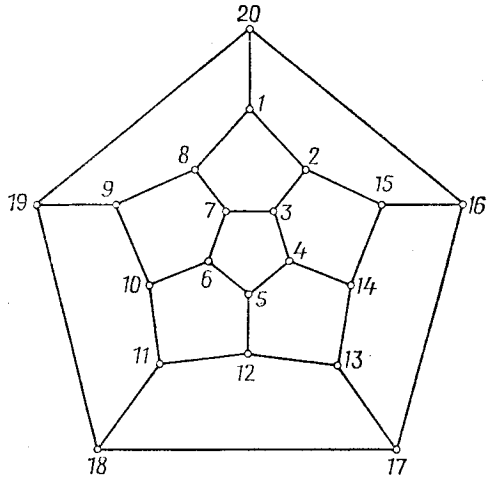


Рис. 3.2. Граф игры Гамильтона.

последовательность вершин 1, 2, . . ., 20, 1 образует такой цикл в графе. Все графы, в которых существует подобный остовный цикл, называются гамильтоновыми. Эти графы мы рассмотрим в разд. 3.2.

3.1. Эйлеровы графы

Эйлеровой цепью в графе G называется замкнутая цепь, содержащая все ребра графа G. К *открытой эйлеровой цепи* относится открытая цепь, содержащая все ребра G. Граф, содержащий эйлерову цепь, называется *эйлеровым графом*.

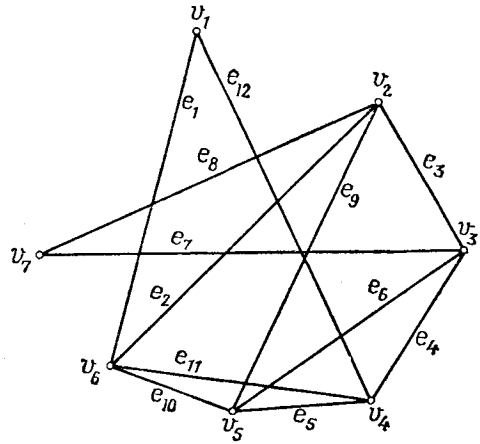
Рассмотрим граф, представленный на рис. 3.3, *а*. Последовательность ребер e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12} образует эйлерову цепь в графе G_1, следовательно, последний является эйлеровым.

В графе G_2 на рис. 3.3, *б* последовательность ребер e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13} образует открытую эйлерову цепь. Однако в графе G_2 эйлеровой цепи нет. Следовательно, граф G_2 не является эйлеровым.

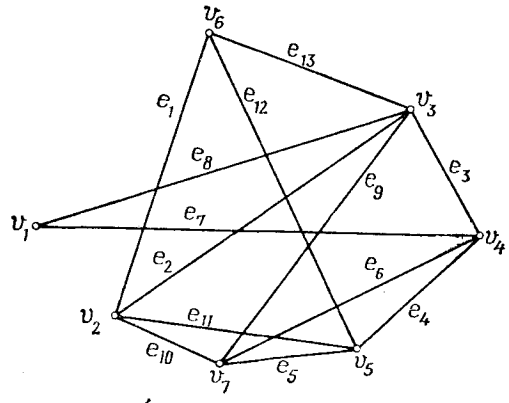
Неэйлеров граф G_3, не обладающий открытой эйлеровой цепью, приведен на рис. 3.3, *в*.

Теперь сформулируем теорему, дающую простую и часто используемую характеризацию эйлеровых графов:

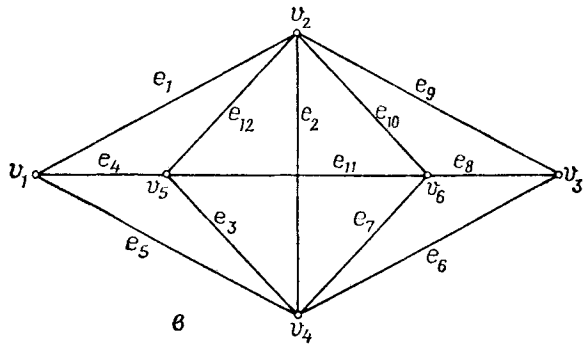
Теорема 3.1. Следующие утверждения эквивалентны для связного графа G:
1. G — эйлеров граф.
2. Степень каждой вершины в графе G четная.



а



б



в

Рис. 3.3.

а — эйлеров граф G_1; *б* — неэйлеров граф G_2, имеющий открытую эйлерову цепь; *в* — неэйлеров граф G_3, не имеющий открытой эйлеровой цепи.

3. G является объединением нескольких реберно-непересекающихся циклов.

Доказательство.

1$\Rightarrow$2. Пусть T — эйлерова цепь в графе G. Предположим, что мы начали движение вдоль T из произвольной вершины v_1, принадлежащей графу G. Представим эйлерову цепь T в виде $v_1=x_1, e_1, x_2, e_2, x_3, \ldots, e_{r+1}, x_r, e_r, x_{r+1}=v_1$, где все ребра различны, а вершины $x_2, \ldots, x_r$ не обязательно различны и некоторые из них могут совпадать с v_1. Тогда очевидно, что пара следующих друг за другом ребер e_i и e_{i+1}, $1\leqslant i\leqslant r-1$, добавляет 2 к степени вершины x_{i+1}. Кроме того, начальное и конечное ребра e_1 и e_r добавляют 2 к степени вершины v_1. Таким образом, все вершины имеют четную степень.

2$\Rightarrow$3. Так как G — связный граф и каждая вершина в нем имеет четную степень, то степень каждой вершины в графе G больше 1. Таким образом, в графе G нет висячих вершин. Следовательно, по теореме 2.5 граф G не является деревом. Это означает, что граф G имеет по крайней мере один цикл, например C_1.

Рассмотрим граф $G_1=G-C_1$. Так как каждая вершина в графе C_1 имеет четную степень, то и каждая вершина, принадлежащая G_1, должна иметь четную степень. Однако возможен случай, когда граф G_1 — несвязный.

Если граф G_1 полностью несвязный, т. е. содержит только изолированные вершины, $G=G_1$, и утверждение 3 доказано. В противном случае граф G_1 имеет по крайней мере один цикл C_2.

Рассмотрим граф $G_2=G_1-C_2=G-C_1-C_2$. Аналогично каждая вершина в графе G_2 имеет четную степень. Если граф G_2 полностью несвязный, тогда $G_2=C_1\cup C_2$. В противном случае будем повторять эту операцию, пока не получим полностью несвязный граф $G_n=G-C_1-C_2-\ldots-C_n$, где $C_1, C_2, \ldots, C_n$ — циклы графа G, попарно не имеющие общих ребер. Тогда $G=C_1\cup C_2\cup\ldots\cup C_n$, и, следовательно, утверждение 3 доказано.

3$\Rightarrow$1. Пусть G — объединение реберно-непересекающихся циклов $C_1, C_2, \ldots, C_n$. Рассмотрим любой из них, например C_1. Так как G — связный граф, то должен существовать по крайней мере один цикл, например C_2, имеющий общую вершину v_1 с циклом C_1. Пусть T_{12} — замкнутая цепь, начинающаяся в вершине v_1 и последовательно проходящая циклы C_1 и C_2. Очевидно, что эта цепь содержит все ребра C_1 и C_2.

Поскольку граф G — связный, цепь T_{12} должна иметь общую вершину v_2 по крайней мере с одним циклом, например C_3, отличным от C_1 и C_2. Замкнутая цепь T_{123}, начинающаяся в v_2 и последовательно проходящая через T_{12} и C_3, содержит все ребра циклов C_1, C_2 и C_3.

Будем повторять эту операцию до тех пор, пока не получим замкнутую цепь $T_{123\ldots n}$, содержащую все ребра циклов $C_1, C_2, \ldots, C_n$. Эта замкнутая цепь и будет являться эйлеровой цепью в графе G. Следовательно, граф G — эйлеров.

По этой теореме граф G_1 (рис. 3.3, *а*) является эйлеровым, так как каждая вершина графа G_1 имеет четную степень, тогда как графы G_2 и G_3 (рис. 3.3, *б*, *в*) содержат несколько вершин нечетной степени и потому не могут быть эйлеровыми.

Нетрудно убедиться в том, что эйлеров граф G_1 является объединением реберно-непересекающихся циклов со следующими множествами ребер: $\{e_4, e_5, e_6\}$, $\{e_2, e_9, e_{10}\}$, $\{e_3, e_7, e_8\}$, $\{e_1, e_{11}, e_{12}\}$.

Из утверждения 3 теоремы 3.1 имеем следующее следствие:

Следствие 3.1.1. Каждая вершина эйлерова графа содержится в некотором цикле.

Хотя в графе, содержащем несколько вершин нечетной степени, эйлерова цепь отсутствует, в таком графе можно построить множество реберно-непересекающихся открытых цепей, которые содержат все ребра. Докажем это в следующей теореме:

Теорема 3.2. Пусть $G=(V, E)$ — связный граф с $2k$ вершинами нечетной

степени, $k \geqslant 1$. Тогда E можно разбить на такие подмножества $E_1, E_2, E_3, \ldots, E_k$, что каждое из них E_i образует в графе G открытую цепь.

Доказательство. Пусть v_i и s_i, $1 \leqslant i \leqslant k$, являются $2k$ вершинами нечетной степени, принадлежащими графу G. Добавим к нему k новых вершин $w_1, w_2, \ldots, w_k$ вместе с $2k$ ребрами (v_i, w_i) и (s_i, w_i), $1 \leqslant i \leqslant k$. В полученном графе G' каждая вершина имеет четную степень, и, следовательно, G' — эйлеров граф.

Заметим, что в любой эйлеровой цепи графа G' ребра (v_i, w_i) и (s_i, w_i) для любого $1 \leqslant i \leqslant k$ появляются последовательно. Удалением этих $2k$ ребер мы получаем k реберно-непересекающихся открытых цепей в графе G, так что каждое ребро графа G присутствует точно в одной из k этих цепей.

Эти открытые цепи и дают необходимое разбиение E.

Следствие 3.2.1. Пусть G — связный граф, имеющий точно две вершины нечетной степени. Тогда он имеет открытую цепь (начинающуюся с одной из вершин нечетной степени и заканчивающуюся в другой), содержащую все ребра G.

Например, граф G_2 (рис. 3.3, *б*) имеет точно две вершины v_6 и v_3 нечетной степени и открытую цепь $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}\}$, содержащую все ребра графа G_2. Эта цепь начинается из вершины v_6 и заканчивается в вершине v_3.

Граф G_3 (рис. 3.3, *в*) имеет 4 вершины нечетной степени и две реберно-непересекающиеся открытые цепи, определяемые следующими множествами ребер: $\{e_1, e_2, e_3, e_4, e_5\}$, $\{e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}\}$.

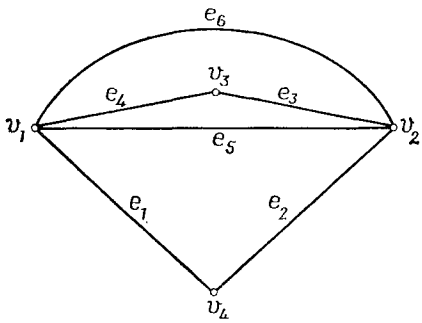


Рис. 3.4. Произвольный эйлеров граф из двух вершин.

Граф называется *произвольно-эйлеровым из вершины* v, если мы, начав двигаться из нее, произвольным образом пройдя вдоль ребер графа G, в итоге получим эйлерову цепь.

Необходимо отметить, что если граф G — произвольно-эйлеров из вершины v, то можно расширить каждую v—v-цепь, не содержащую всех ребер, до эйлеровой цепи графа G. Другими словами, если эйлеров граф G не является произвольно-эйлеровым из вершины v, то должна существовать замкнутая v — v-цепь, содержащая все ребра, инцидентные вершине v, но не включающая в себя все ребра графа G.

В качестве примера рассмотрим эйлеров граф на рис. 3.4. Этот граф является произвольно-эйлеровым из вершин v_1 и v_2. Можно проверить, что для каждой вершины v_i, отличной от вершин v_1 и v_2, существует замкнутая цепь, содержащая все ребра, инцидентные вершине v_i, но не включающая в себя всех ребер графа G. Например, этим свойством обладает замкнутая v_3—v_3-цепь, состоящая из ребер e_4, e_1, e_2 и e_3.

Следующая теорема дает характеризацию графов, являющихся произвольно-эйлеровыми из вершины v.

Теорема 3.3. Эйлеров граф G является произвольно-эйлеровым из вершины v тогда и только тогда, когда каждый цикл графа G содержит эту вершину.

Доказательство.

Необходимость. Пусть G — произвольно-эйлеров граф из вершины v. Предположим, что в графе G существует цикл C, не содержащий v. Рассмотрим граф $G'=G-C$. Каждая вершина в нем имеет четную степень. Граф G' может быть и несвязным. Однако его компонента G'', содержащая вершину v, является эйлеровым графом и содержит все ребра, инцидентные вершине v. Таким образом, в графе G'' существует эйлерова цепь T, начинающаяся и возвращающаяся в вершину v. Эта цепь обязательно состоит из всех ребер, инцидентных вершине v. Поэтому ее нельзя расширить до включения всех ребер цикла C. Это противоречит тому, что G — произвольно-эйлеров граф из вершины v.

Достаточность. Пусть вершина v эйлерова графа G присутствует в каждом его цикле. Предположим, что граф G не является произвольно-эйлеровым из вершины v. Тогда существует замкнутая v—v-цепь T, содержащая все ребра, инцидентные вершине v, но не включающая в себя всех ребер графа G. Далее, существует вершина $u\neq v$, являющаяся концевой вершиной ребра, не содержащегося в цепи T.

После удаления из графа G ребер цепи T получим граф G', в котором v будет изолированной вершиной. Каждая вершина в графе G' имеет четную степень. Таким образом, компонента графа G', содержащая вершину u, является эйлеровым графом. Согласно следствию 3.1.1, существует цикл, содержащий вершину u. Это противоречит предположению, что вершина v находится в каждом цикле графа G.

Нетрудно убедиться в том, что в графе G на рис. 3.4 вершины v_1 и v_2 присутствуют в каждом цикле. Таким образом, G — произвольно-эйлеров граф обеих этих вершин. С другой стороны, для каждой из остальных вершин существует цикл, не содержащий этой вершины.

Граф называется *произвольно-эйлеровым*, если он является таковым для каждой из своих вершин. Из теоремы 3.3 следует, что вершины произвольно-эйлерова графа G находятся точно в одном цикле C графа G и в последнем не существует других циклов. Иначе говоря, G — произвольно эйлеров граф тогда и только тогда, когда он является циклом.

3.2. Гамильтоновы графы

Гамильтоновым циклом в графе G называется цикл, содержащий все вершины графа G. *Гамильтонов путь* в графе G — это путь, содержащий все вершины графа G.

Граф G называется *гамильтоновым*, если он имеет гамильтонов цикл.

Граф G_1, представленный на рис. 3.5, *а*, является гамильтоновым, так как последовательность его ребер e_1, e_2, e_3, e_4, e_5, e_6 образует гамильтонов цикл. Граф на рис. 3.5, *б* имеет гамильтонов путь, состоящий из ребер e_1, e_2, e_3, e_4, но не имеет гамильтонова цикла.

Если эйлерова цепь является замкнутым маршрутом, точно один раз проходящим через каждое ребро, то гамильтонов цикл является

замкнутым маршрутом, точно один раз проходящим через каждую вершину. Таким образом, между эйлеровыми и гамильтоновыми графами имеется большое сходство. Однако это вовсе не означает, что для гамильтоновых графов существует такая же простая и элегантная характеризация, как и для эйлеровых графов. Одна из главных нерешенных проблем теории графов состоит в выработке

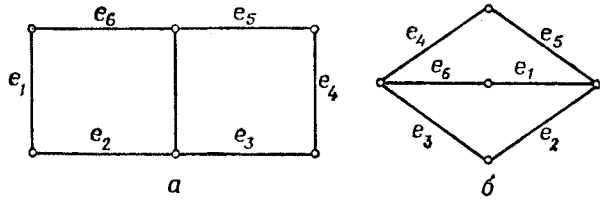


Рис. 3.5.
а — гамильтонов граф; *б* — негамильтонов граф, имеющий гамильтонов путь.

такой характеризации гамильтоновых графов. Однако известно несколько достаточных условий того, что простой граф является гамильтоновым. (Напомним, что простым называется граф, не содержащий параллельных ребер и петель.) Некоторые из таких условий мы рассмотрим в этом разделе.

Последовательность $d_1 \leqslant d_2 \leqslant \ldots \leqslant d_n$ называется *графической*, если существует такой граф G на n вершинах $v_1, v_2, \ldots, v_n$, что степень $d(v_i)$ вершины v_i равна d_i для любого i. В этом случае $(d_1, d_2, \ldots, d_n)$ называется *последовательностью степеней* графа G.

Если $S: d_1 \leqslant d_2 \leqslant \ldots \leqslant d_n$ и $S^*: d_1^* \leqslant d_2^* \leqslant \ldots \leqslant d_n^*$ являются такими графическими последовательностями, что $d_i^* \geqslant d_i$ для $1 \leqslant i \leqslant n$, то говорят, что S^* *мажорирует* S.

Следующий результат получен в работе [3.11].

Теорема 3.4. Простой граф $G = (V, E)$ порядка n, имеющий последовательность степеней $d_1 \leqslant d_2 \leqslant \ldots \leqslant d_n$, является гамильтоновым, если

$$d_k \leqslant k < \frac{1}{2} n \Rightarrow d_{n-k} \geqslant n - k. \tag{3.1}$$

Доказательство. Прежде всего отметим, что если $d_k \leqslant k$, то число вершин, степень которых не превышает k, равно по крайней мере k. Аналогично если $d_{n-k} \geqslant n-k$, то число вершин, степень которых не превышает $n-k$, равно по крайней мере $k+1$. Если графическая последовательность удовлетворяет условию (3.1), то это соотношение будет верно и для последовательности, мажорирующей данную.

Докажем теорему от противного. Пусть существует простой негамильтонов граф, последовательность степеней которого удовлетворяет условию (3.1). Тогда он является остовным подграфом простого максимального негамильтонова графа $G = (V, E)$, в котором последовательность степеней $d_1 \leqslant d_2 \leqslant \ldots \leqslant d_n$ также удовлетворяет условию (3.1).

Пусть u и v — две такие несмежные вершины графа G, что сумма $d(u) + d(v)$ является по возможности наибольшей и $d(u) \leqslant d(v)$. Из того, что G — максималь-

ный негамильтонов граф, следует, что добавление ребра, соединяющего вершины u и v, преобразует граф в гамильтонов.

Таким образом, в графе G существует гамильтонов путь $u=u_1, u_2, u_3, u_4, \ldots, \ldots, u_n=v$, где u и v — конечные вершины (рис. 3.6).

Пусть $S=\{i \mid (u_1, u_{i+1}) \in E\}$, $T=\{i \mid (u_i, u_n) \in E\}$. Не существует $j \in S \cap T$, так как если $j \in S \cap T$ существует, то граф G будет содержать ребра (u_i, u_{j+1}) и (u_j, u_n), а тогда циклическая последовательность вершин u_j, u_{j-1}, $\ldots$, u_1, u_{j+1}, u_{j+2}, $\ldots$, u_n, u_j образует в графе G гамильтонов цикл.

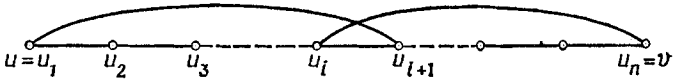


Рис. 3.6.

Так как вершина $u_n=v$ не принадлежит ни S, ни T, из этого следует, что $S \cup T \subseteq \{1, 2, \ldots, n-1\}$. Поэтому $d(u)+d(v)=|S|+|T|<n$ и $d(u)<\frac{1}{2}n$, где $|X|$ обозначает число элементов множества X.

Так как $S \cap T=\varnothing$, то ни одна вершина u_j с $j \in S$ не смежна с вершиной v. Из выбора $d(u)$ и $d(v)$ следует, что для $j \in S$ справедливо неравенство $d(u_j) \leqslant d(u)$. Таким образом, существует по крайней мере $|S|=d(u)$ вершин, степени которых не превышают $d(u)$. Поэтому если мы положим $k=d(u)$, то получим $d_k \leqslant k \leqslant \frac{1}{2}n$ и, следовательно, по условию (3.1) $d_{n-k} \geqslant n-k$. Это означает, что существует по крайней мере $k+1$ вершин, степени которых не превышают $n-k$. Вершина u может быть смежна, самое большее, с k из $k+1$ вершин, так как $d(u)=k$. Таким образом, существует вершина w с $d(w) \geqslant n-k$, не смежная с вершиной u. Но тогда $d(u)+d(w) \geqslant n > d(u)+d(v)$, что противоречит выбору $d(u)$ и $d(v)$.

В следующем следствии мы установим достаточные условия гамильтоновости графа, представленные в работах [3.2—3.5].

Следствие 3.4.1. Простой граф $G=(V, E)$ порядка $n \geqslant 3$ с последовательностью степеней $d_1 \leqslant d_2 \leqslant \ldots \leqslant d_n$ является гамильтоновым, если для него выполнено одно из следующих условий:

1. $1 \leqslant k \leqslant n \Rightarrow d_k \geqslant \frac{1}{2}n$.
2. $(u, v) \notin E \Rightarrow d(u)+d(v) \geqslant n$.
3. $1 \leqslant k < \frac{1}{2}n \Rightarrow d_k > k$.
4. $j<k$, $d_j \leqslant j$, $d_k \leqslant k-1 \Rightarrow d_j+d_k \geqslant n$.

Доказательство. Докажем эти следствия, показав, что все эти условия влекут за собой выполнение (3.1).

$1 \Rightarrow 2$. Очевидно, что любая последовательность степеней, удовлетворяющая условию 1, удовлетворяет также и условию 2.

$2 \Rightarrow 3$. Пусть из условия 2 не следует условие 3. Тогда существует такое t, что $1 \leqslant t < \frac{1}{2}n$ и $d_t \leqslant t$. Теперь предположим, что существует l, для которого $l<t$ и $(v_l, v_t) \in E$. Тогда выражение $d_l+d_t \leqslant 2d_t < 2\frac{1}{2}n=n$ противоречит условию 2. Таким образом, подграф графа G, порожденный на множестве вершин $v_1, v_2, \ldots, \ldots, v_t$, является полным графом.

Так как $d_t \leqslant t$, каждая вершина v_i, $1 \leqslant i \leqslant t$, будет смежной, самое большее, с одной вершиной v_j, $t+1 \leqslant j \leqslant n$. Далее, из $t < \frac{1}{2}n$ следует, что $n-t > t$. Таким образом, существует вершина v_j, $t+1 \leqslant j \leqslant n$, не являющаяся смежной ни с одной

вершиной v_i, $1 \leqslant i \leqslant t$. Поэтому $d_j \leqslant n-t-1$. Но тогда $d_t+d_j \leqslant t+n-t-1<n$. Таким образом, мы имеем $(v_t, v_j) \notin E$ и $d_t+d_j<n$, что противоречит условию 2.

$3 \Rightarrow 4$. Если из условия 3 не следует условие 4, то существует такое $j<k$, что $d_j \leqslant j$, $d_k \leqslant k-1$ и $d_j+d_k<n$. Тогда $d_j<\frac{1}{2}n$. Если теперь мы положим $t=d_j<\frac{1}{2}n$, то получим $d_t=d_{d_j} \leqslant d_j=t$. Поэтому имеем неравенства, противоречащие условию 3: $t<\frac{1}{2}n$ и $d_t<t$.

$4 \Rightarrow$ (3.1). Если это не верно, то существует такое t, что $d_t \leqslant t<\frac{1}{2}n$ и $d_{n-t} \leqslant n-t-1$. Но тогда $d_t+d_{n-t} \leqslant t+n-t-1=n-1$, что противоречит условию 4.

Очевидно, если какая-либо графическая последовательность удовлетворяет условиям теоремы 3.4 или следствия 3.4.1, то тем же условиям удовлетворяет любая графическая последовательность, мажорирующая ее.

Условие Хватала — самое сильное из этих пяти условий и вообще из всех подобных условий. Это означает, что если какая-либо графическая последовательность не удовлетворяет условию Хватала, то она мажорируется последовательностью степеней негамильтонова графа [3.1].

Хотя в общем случае довольно трудно установить, является ли граф негамильтоновым, в определенных случаях это удается сде-

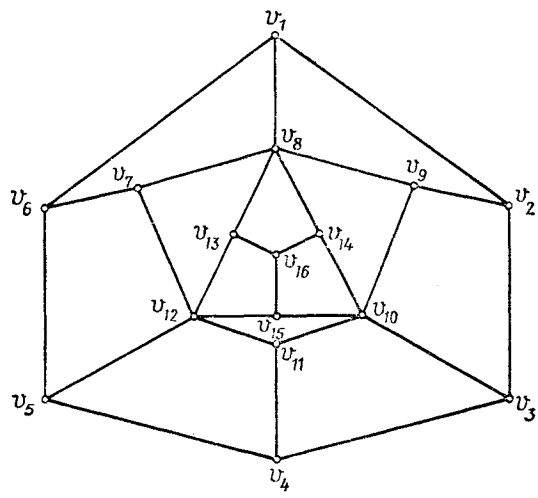


Рис. 3.7. Негамильтонов граф.

лать с помощью изощренных методов. Это иллюстрируется в работе [3,6] на следующем примере.

Рассмотрим граф G, представленный на рис. 3.7. Покажем, что он не содержит гамильтонова пути.

В гамильтонов путь можно включить не более двух ребер, инцидентных какой-либо вершине. В графе G вершина v_8 имеет степень 5, и, следовательно, по меньшей мере три ребра, инцидентных вершине v_8, не войдут в гамильтонов путь. То же самое справедливо и для вершин v_{10} и v_{12}. Так как степени вершин v_2, v_1, v_6, v_{16} равны 3, то по крайней мере одно из ребер, инцидентных каждой из этих вершин, не будет включено в гамильтонов путь. Таким образом, в гамильтонов путь нельзя включить по крайней мере 13 из 27 ребер графа. Следовательно, чтобы составить гамильтонов путь на 16 вершинах графа G, ребер не достаточно. Поэтому граф G гамильтонова пути не содержит.

Граф называется *произвольно-гамильтоновым из вершины* v, если любой путь, начинающийся в вершине v, можно расширить до гамильтонова v — v-цикла. Граф называется *произвольно-гамильтоновым*, если он является таковым из каждой своей вершины.

Следующая теорема полностью характеризует произвольно гамильтоновы графы. Доказательство ее можно найти в работе [3.7].

Теорема 3.5. Простой граф порядка n произвольно-гамильтонов тогда и только тогда, когда он является циклом, либо полным графом, либо полным двудольным графом $K_{n/2,\, n/2}$; последнее возможно, только когда число n — четное.

Закончим этот раздел рассмотрением задачи о коммивояжере. Она заключается в следующем:

Коммивояжеру необходимо посетить несколько городов. Какой маршрут он должен избрать, чтобы, начав двигаться из своего родного города, по одному разу побывать в каждом городе и затем вернуться домой по наикратчайшему пути?

Представим города в виде вершин, а дороги — в виде ребер графа. Длину дороги можно представить весом соответствующего ребра. Если между каждой парой вершин существует дорога, соединяющая их, то задача коммивояжера эквивалентна отысканию наикратчайшего цикла в полном графе, в котором каждому ребру сопоставлен вес [1].

В полном графе порядка n существует $(n-1)!/2$ гамильтоновых циклов. Подход «грубой силы» к задаче о коммивояжере состоит в порождении всех $(n-1)!/2$ гамильтоновых циклов и последующем выборе кратчайшего. Объем затрат при таком подходе слишком велик (даже для ЭВМ) уже для величин n, меньших 50. Для произвольного n алгоритма решения этой проблемы просто не существует.

Для более полного ознакомления с этой задачей можно использовать работы [3.8—3.11].

---

[1] Вес *цикла* — сумма весов ребер этого цикла.

3.3. Замечания, касающиеся литературы

В работе [3.12] единым образом доказывается несколько достаточных условий гамильтоновости графа. В этой книге также обсуждается вопрос разбиения множества ребер графа на пути, циклы и т. д. Этот вопрос затрагивается в работе [3.13].

В разд. 3.2 мы упоминали, что если последовательность степеней графа не удовлетворяет условию Хватала (3.1), то она мажорируется последовательностью степеней негамильтонова графа. Далее, существуют графические последовательности, которые не удовлетворяют условию (3.1), но обязательно являются последовательностями степеней гамильтоновых графов. Более подробно это рассмотрено в работе [3.14]. Обобщение теоремы 3.4 можно найти в работе [3.15].

Для более детального рассмотрения гамильтоновых циклов следует обратиться к работам [3.16, 3.17].

Упражнения

3.1. Пусть G — связный граф, имеющий $2k$ вершин нечетной степени. Докажите, что множество ребер E графа G нельзя разбить на такие подмножества $E_1, E_2, \ldots, E_l$, $l<k$, что каждое E_i образует открытую цепь в графе G.

3.2. Докажите, что если граф G является произвольно-эйлеровым из вершины v, то либо v — единственная точка сочленения, либо граф G не имеет ни одной точки сочленения.

3.3. Пусть G — эйлеров граф на $n\geqslant 3$ вершинах. Докажите, что G — произвольно-эйлеров граф из одной, двух или всех своих вершин или ни из одной из них.

3.4. Докажите, что если граф G — произвольно-эйлеров из вершины v, то $\Delta(G)=d(v)$, где $\Delta(G)$ — максимальная степень в графе G.

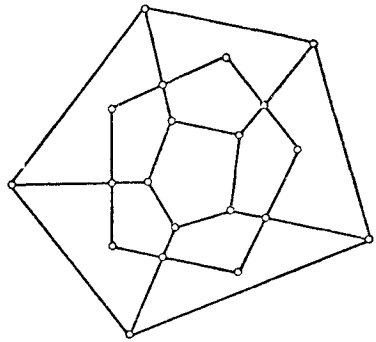


Рис. 3.8.

3.5. Пусть G — произвольно-эйлеров из вершины v. Докажите, что если $d(u)=d(v)$, $u\neq v$, то граф G — произвольно-эйлеров из вершины u.

3.6. Существует ли граф, в котором эйлерова цепь является одновременно и гамильтоновым циклом? Охарактеризуйте такие графы.

3.7. Покажите, что граф на рис. 3.8 не содержит гамильтонова пути.

3.8. Пусть G — простой связный граф на $n>2\delta(G)$ вершинах. Докажите, что длина наиболее длинного пути графа G должна быть больше или равна $2\delta(G)$, где $\delta(G)$ — минимальная степень вершины в графе G [3.2].

3.9. Пусть G — простой граф на n вершинах. Докажите, что он содержит гамильтонов путь, если для каждой пары вершин u и v в графе G выполняется следующее соотношение: $d(u)+d(v)\geqslant n-1$.

3.10. Пусть $d_1\leqslant d_2\leqslant\ldots\leqslant d_n$ — степени вершин простого графа, имеющего $n\geqslant 2$ вершин. Докажите, что если $d_k\leqslant k-1\leqslant\frac{1}{2}n-1\Rightarrow d_{n+1-k}\geqslant n-k$, то граф G имеет гамильтонов путь [3.1].

3.11. Пусть G — простой граф на n вершинах и m ребрах и $n\geqslant 3$, $m\geqslant(n^2-3n+6)/2$. Докажите, что граф G является гамильтоновым [3.3].

3.12. Предположим, что G — простой граф на n вершинах, имеющий такие несмежные вершины u и v, что $d(u)+d(v)\geqslant n$. Пусть G' — граф, полученный при добавлении к графу G ребра (u, v). Покажите, что граф G гамильтонов тогда и только тогда, когда граф G' также гамильтонов.

3.13. Замыкание простого графа G на n вершинах есть граф, полученный последовательным соединением пар несмежных вершин, сумма степеней которых не менее n до тех пор, пока это возможно. Покажите, что замыкание графа является правильно определенным.

Примечание. Простым следствием упражнения 3.12 является то, что граф будет гамильтоновым тогда и только тогда, когда его замыкание — гамильтонов граф. Таким образом, граф G — гамильтонов, если его замыкание является полным графом. Этот результат можно использовать для доказательства некоторых достаточных условий, определяющих гамильтоновы графы [3.13, 3.15].

3.14. Докажите, что максимальное число попарно-непересекающихся гамильтоновых циклов в полном графе K_n равно $[(n-1)/2]$.

Примечание. Символ $[x]$ означает наибольшее целое число, меньшее или равное x.

3.15. Граф G называется гамильтоново-связным, если для каждой пары различных вершин u и v графа G существует гамильтонов u—v-путь. Докажите, что простой граф G порядка $n\geqslant 3$ является гамильтоново-связным, если для каждой пары несмежных вершин u и v графа G выполняется следующее неравенство: $d(u)+d(v)\geqslant n+1$ [3.18].

3.16. Покажите, что граф деревьев (определенный в упражнении 2.28) связного графа является гамильтоновым [3.19, 3.20].

4. Графы и векторные пространства

Современная алгебра является ценным и полезным инструментом при изучении и решении многих научных и инженерных проблем. Очень полезным оказалось определение алгебраической структуры множества объектов, так как сильные и элегантные результаты, полученные для алгебраической структуры, можно затем перенести на эти множества. Теории систем, электрических цепей, кодирования, последовательных и комбинационных переключательных цепей, а также вычислительная техника — лишь немногие примеры, показывающие преимущество такого подхода.

В этой главе мы покажем, что графу можно сопоставить векторное пространство, и детально изучим свойства двух важных векторных подпространств: циклов и разрезающих множеств. В части II книги мы увидим, что эти подпространства фактически определяют пространства напряжений и токов электрической цепи.

В первых двух разделах этой главы мы дадим представление о некоторых элементарных алгебраических понятиях и некоторых результатах, используемых в дальнейшем. Для более детального ознакомления с этими вопросами следует обратиться к работам [4.1—4.3].

4.1. Группы и поля

Рассмотрим конечное множество $S=\{a, b, c,\ldots\}$. Обозначим символом «+» бинарную операцию, определенную на множестве S. Эта операция сопоставляет каждой паре элементов a и b, принадлежащих S, единственный элемент, обозначаемый $a+b$. Множество S называется *замкнутым* относительно операции «+», если элементы a, b, $a+b$ принадлежат S. Говорят, что операция «+» *ассоциативна*, если $a+(b+c)=(a+b)+c$ для любых a, b, c из множества S. Операция «+» называется *коммутативной*, если равенство $a+b=b+a$ справедливо для любых a, b, принадлежащих S.

Определим понятие группы. Множество S, в котором выполняется бинарная операция «+», называемая операцией сложения, является *группой*, если имеют место следующие аксиомы:

1) S замкнуто относительно «+»,

2) «+» является ассоциативной операцией,

3) в множестве S существует такой единственный элемент e, что $a+e=e+a=a$ для любого a, принадлежащего S. Элемент e назы-

вается *нулевым элементом* группы.

4) Для каждого элемента a множества S существует такой единственный элемент b, что $b+a=a+b=e$. Элемент b называется элементом, обратным к a, и наоборот. Очевидно, что нулевой элемент является обратным к самому себе. Группа называется *абелевой*, если операция «+» коммутативна. Обычным примером группы является множество $S=\{\ldots, -2, -1, 0, 1, 2, \ldots\}$ всех целых чисел, на котором определена обычная операция сложения. В этой группе 0 является нулевым элементом, а $-a$ — обратным элементом для всех a, принадлежащих S. Отметим, что эта группа — абелева. Является ли множество S всех целых чисел, на котором определена операция умножения, группой? (Нет. Почему?).

Другим важным примером группы является множество $Z_p=\{0, 1, 2, \ldots, p-1\}$ целых чисел с операцией сложения по модулю p [1]. В этой группе 0 — нулевой элемент. Далее, целое число $p-a$ является обратным к целому числу a, для любого a не равного 0. Естественно, 0 обратен к самому себе. Эта группа также абелева.

Ниже приведена таблица операции сложения для Z_5.

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Введем понятие «поле». Множество F, на котором определены две операции $+$ и $\cdot$, называемые операциями сложения и умножения соответственно, является *полем*, если выполняются следующие аксиомы:

1) F — абелева группа относительно операции «+», нулевой элемент которой обозначается через e.

3) Множество $F-\{e\}$ — абелева группа относительно операции умножения.

3) Операция умножения является дистрибутивной относительно операции сложения, т. е. $a(b+c)=ab+ac$ для любых a, b, c, принадлежащих множеству F.

В качестве примера вновь рассмотрим множество $Z_p=\{0, 1, 2, \ldots, p-1\}$, на котором выполняются операции сложения и умножения по модулю p. Как было показано ранее, множество Z_p является абелевой группой относительно операции сложения по модулю p с нулевым элементом 0. Нетрудно видеть, что множество $Z_p-\{0\}=\{1, 2, \ldots, p-1\}$ будет группой относительно операции умноже-

[1] Если $a=mp+q$, $0\leqslant q\leqslant p-1$, то говорят, что $a=q$ (по модулю p).

ния по модулю p тогда и только тогда, когда p — простое число. Эта группа также является абелевой. Можно проверить, что операция умножения по модулю p дистрибутивна относительно операции сложения по модулю p. Таким образом, множество Z_p будет полем тогда и только тогда, когда p простое число.

Поле Z_p обычно обозначают символом $GF(p)$ и называют *полем Галуа*. Ниже в качестве примера приводится таблица операции умножения для GF (5):

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Особый интерес представляет поле $GF(2)$, т. е. множество целых чисел с операциями сложения и умножения по модулю 2. В этом поле $0+0=0$, $1+0=0+1=1$, $1+1=0$; $0\cdot 0=0$, $1\cdot 0=0\cdot 1=0$, $1\cdot 1=1$.

4.2. Векторные пространства

Рассмотрим множество S с бинарной операцией $\boxplus$. Пусть F — поле, где символы $+$ и $\cdot$ обозначают операции сложения и умножения соответственно. Пусть также определена операция умножения, обозначаемая символом $*$, между элементами F и S. Эта операция сопоставляет каждой упорядоченной паре (α, s), где α принадлежит F, а s принадлежит S, единственный элемент множества S, обозначаемый $\alpha * s$. Множество S называется *векторным пространством* над полем F, если выполнены следующие аксиомы:

1) S — абелева группа относительно операции $\boxplus$.
2) Для любых элементов α и β поля F и любых s_1 и s_2, принадлежащих S, имеем $\alpha*(s_1 \boxplus s_2)=(\alpha*s_1)\boxplus(\alpha*s_2)$ и $(\alpha+\beta)*s_1=(\alpha*s_1)\boxplus(\beta*s_1)$.
3) Для любых элементов α и β, принадлежащих F, и любого элемента s из множества S имеет место равенство $(\alpha\cdot\beta)*s=\alpha*(\beta*s)$.
4) Для любого элемента s множества S выполняется равенство $1*s=s$, где 1 — мультипликативный нулевой элемент поля F.

Приведем важный пример векторного пространства.

Рассмотрим множество W всех n-векторов [1] над полем F. (Заметим, что элементы n-векторов принадлежат F.) Символы $+$ и $\cdot$ будут обозначать операции сложения и умножения в F соответственно, а 0 и 1 — аддитивный и мультипликативный нулевые элементы в F. Пусть $\boxplus$ — операция сложения на множестве W, а $*$ — операция

[1] *n-вектор* над полем F — это вектор-строка из n элементов поля F.

умножения между элементами F и W, определенные следующим образом:

1. Если $\omega_1=(\alpha_1, \alpha_2, \ldots, \alpha_n)$ и $\omega_2=(\beta_1, \beta_2, \ldots, \beta_n)$ — элементы W, то $\omega_1 \boxplus \omega_2=(\alpha_1+\beta_1, \alpha_2+\beta_2, \ldots, \alpha_n+\beta_n)$.
2. Если α принадлежит F, то $\alpha*\omega_1=(\alpha\cdot\alpha_1, \alpha\cdot\alpha_2, \ldots, \alpha\cdot\alpha_n)$.

Нетрудно установить, что W — абелева группа относительно операции $\boxplus$, в которой нулевым элементом считается n-вектор (0, 0, 0, . . ., 0). Таким образом, множество W удовлетворяет первой аксиоме в определении векторного пространства. Легко убедиться в том, что элементы множеств W и F удовлетворяют и трем другим аксиомам этого определения.

Например, множество W, состоящее из восьми 3-векторов, приведенных ниже, является векторным пространством над $GF(2)$: $\omega_0=(000)$, $\omega_1=(001)$, $\omega_2=(010)$, $\omega_3=(011)$, $\omega_4=(100)$, $\omega_5=(101)$, $\omega_6=(110)$, $\omega_7=(111)$. Это векторное пространство будем использовать во всех примерах данного раздела.

Теперь приведем несколько важных определений и результатов (без доказательств), связанных с векторным пространством.

Рассмотрим векторное пространство S над полем F.

Векторы и скаляры. Элементы множества S называются *векторами*, а элементы множества F — скалярами.

Линейная комбинация. Если элемент s множества S можно представить в виде $s=(\alpha_1*s_1)\boxplus(\alpha_2*s_2)\boxplus\ldots\boxplus(\alpha_j*s_j)$, где s_i — векторы, а α_i — скаляры, то говорят, что s является линейной комбинацией $s_1, s_2, \ldots, s_j$.

Линейная независимость. Векторы $s_1, s_2, \ldots, s_n$ — линейно-независимы, если никакой из векторов этого множества нельзя представить в виде *линейной комбинации* остальных.

Базисные векторы. Векторы $s_1, s_2, \ldots, s_n$ образуют базис в векторном пространстве S, если они линейно-независимы и любой вектор в этом пространстве можно представить в виде линейной комбинации этих векторов. Векторы $s_1, s_2, \ldots, s_n$ называются *базисными*. Представление вектора в виде линейной комбинации базисных векторов единственно для данного базиса. Векторное пространство может иметь и несколько базисов. Однако можно доказать, что все базисы имеют одинаковое число векторов.

Размерность. Размерность векторного пространства S, обозначаемая $\dim(S)$, определяется числом базисных векторов в базисе S.

Подпространство. Если S' — подмножество векторного пространства S над полем F, то S' — подпространство пространства S, если S' также векторное пространство над F.

Прямая сумма. Прямой суммой $S_1 \boxplus S_2$ двух подпространств S_1 и S_2 пространства S называется множество всех векторов вида $s_i \boxplus s_j$, где s_i — принадлежит подпространству S_1, а s_j — подпространству S_2. $S_1 \boxplus S_2$ также является подпространством, и размерность его определяется следующим образом: $\dim(S_1 \boxplus S_2) = \dim$

$(S_1)+\dim(S_2)-\dim(S_1 \cap S_2)$. Заметим, что если S_1 и S_2 — подпространства, то $S_1 \cap S_2$ также подпространство. Проиллюстрируем эти определения с помощью векторного подпространства W 3-векторов над полем $GF(2)$. Векторы $\omega_0, \omega_1, \ldots, \omega_7$, принадлежащие подпространству W, определены ранее.

1. ω_1 — линейная комбинация ω_6 и ω_7, поскольку $\omega_1=(1*\omega_6)\boxplus\boxplus(1*\omega_7)$.
2. Векторы ω_3 и ω_4 — линейно-независимы, так как ни один из них нельзя выразить через другой. Следует отметить, что ω_0 и ω_i линейно-независимы для любого i.
3. Векторы ω_1, ω_2, ω_4 образуют в пространстве W базис, поскольку они линейно-независимы, а оставшиеся векторы, как показано ниже, можно представить в виде линейной комбинации этих векторов: $\omega_0=(0*\omega_1)\boxplus(0*\omega_2)\boxplus(0*\omega_4)$, $\omega_3=(1*\omega_1)\boxplus(1*\omega_2)$, $\omega_5=(1*\omega_1)\boxplus(1*\omega_4)$, $\omega_6=(1*\omega_2)\boxplus(1*\omega_4)$, $\omega_7=(1*\omega_1)\boxplus(1*\omega_2)\boxplus(1*\omega_4)$. Можно проверить, что векторы ω_1, ω_3, ω_7 также образуют базис.
4. Размерность векторного пространства W равна 3, так как три вектора образуют его базис.
5. Множества $W'=\{\omega_0, \omega_1, \omega_2, \omega_3\}$ и $W''=(\omega_0, \omega_1, \omega_6, \omega_7\}$ — подпространства пространства W. Базисами W' и W'' являются $\{\omega_1, \omega_2\}$ и $\{\omega_1, \omega_6\}$ соответственно. Таким образом, $\dim(W')=2$ и $\dim(W'')=2$.
6. Если W' и W'' определены как в п. 5, то $W'\boxplus W''=\{\omega_0, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7\}$.
7. Размерность $W'\boxplus W''$ определяется следующим равенством: $\dim(W'\boxplus W')=\dim(W')+\dim(W'')-\dim(W'\boxplus W'')=4-\dim\times(W'\cap W'')$.

Поскольку $W'\cap W''=\{\omega_0, \omega_1\}$, имеем $\dim(W'\cap W'')=1$. Таким образом, получаем $\dim(W'\boxplus W'')=4-1=3$, Того же самого результата можно достичь с учетом $W'\boxplus W''=W$.

Теперь определим изоморфизм между двумя векторными пространствами над одним и тем же полем.

Пусть S и S' — два n-мерных векторных подпространства над полем F. Говорят, что S и S' *изоморфны*, если между ними существует такое взаимно-однозначное соответствие, что справедливы утверждения:

1. Если векторам s_1 и s_2, принадлежащим подпространству S, соответствуют векторы s_1' и s_2' пространства S', то вектору $s_1\boxplus s_2$ соответствует вектор $s_1'\mathbin{\triangle} s_2'$, где $\boxplus$ и $\triangle$ — операции в S и S' соответственно.
2. Для любого α в поле F вектору $\alpha*s$ соответствует вектор $\alpha\mathbin{\triangle} s'$, если вектор s пространства S соответствует вектору s' пространства S', где $*$ и $\triangle$ — операции в S и S' соответственно.

Рассмотрим n-мерное векторное пространство S над полем F и n-мерное векторное пространство W векторов над полем F. Пусть

векторы s_1, s_2,. . ., s_n образуют базис в пространстве S. Сопоставим вектору s из пространства S вектор $\omega=(\alpha_1, \alpha_2,. . ., \alpha_n)$ [1] из пространства W тогда и только тогда, когда $s=(\alpha_1*s_1)\boxplus(\alpha_2*s_2)\boxplus . . . . . \boxplus(\alpha_n*s_n)$. Нетрудно видеть, что это взаимно-однозначное соответствие определяет изоморфизм между пространствами S и W.
Таким образом, сформулируем следующий важный вывод:

Теорема 4.1. Любое n-мерное векторное пространство над полем F является изоморфным векторному пространству W n-векторов над полем F.

Эта теорема является главным звеном, связывающим векторные пространства и матрицы. Она предполагает, что n-мерное векторное пространство над полем F можно исследовать с использованием n-мерного векторного пространства W всех n-векторов над полем F.
Завершим этот раздел определением двух наиболее важных понятий — *скалярное произведение* и *ортогональность*. Пусть $\omega_1=(\alpha_1 \alpha_2. . .\alpha_n)$ и $\omega_2=(\beta_1\beta_2. . .\beta_n)$ — два элемента векторного пространства W n-векторов над полем F. *Скалярным произведением* ω_1 и ω_2, обозначаемым через $<\omega_1, \omega_2>$, называется скаляр, определяемый выражением $<\omega_1, \omega_2> = \alpha_1\cdot\beta_1+\alpha_2\cdot\beta_2+...+\alpha_n\cdot\beta_n$. Например, если $\omega_1=(0\ 1\ 0\ 0\ 1)$ и $\omega_2=(1\ 0\ 1\ 1\ 1)$, то $<\omega_1, \omega_2>= =<\omega_1, \omega_2>=0\cdot1+1\cdot0+0\cdot1+0\cdot1+1\cdot1=0+0+0+0+1=1$. Векторы ω_i и ω_j *ортогональны* относительно друг друга, если $<\omega_i, \omega_j>=0$, где 0 — аддитивный нулевой элемент поля F. Например, векторы $\omega_1=(1\ 1\ 0\ 1\ 1)$ и $\omega_2=(1\ 1\ 1\ 0\ 0)$ ортогональны над полем $GF(2)$, поскольку $<\omega_1, \omega_2>=1\cdot1+1\cdot1+0\cdot1+1\cdot0+1\cdot0=1+1+ +0+0+0=0$. Два подпространства W' и W'' векторного пространства W являются *ортогональными подпространствами* этого же пространства, если каждый вектор одного подпространства ортогонален любому вектору другого подпространства. Два подпространства W' и W'' пространства W называются *ортогональными дополнениями* пространства W, если они ортогональны друг другу и их прямая сумма $W'\boxplus W''$ равна векторному пространству W. В качестве примера вновь рассмотрим 3-мерное векторное пространство W 3-векторов над полем $GF(2)$. В этом векторном пространстве подпространства $W'=\{\omega_0, \omega_1, \omega_2, \omega_3\}$ и $W''=\{\omega_0, \omega_4\}$ ортогональны друг другу. Нетрудно убедиться, что прямая сумма W' и W'' равна W. Следовательно, они являются ортогональными дополнениями.

4.3. Векторное пространство графа

В этом разделе мы рассмотрим соответствие между графом и векторным пространством, а также определим два важных подпространства векторного пространства.

[1] $*\alpha_1, \ldots, \alpha_n$ называется *координатами* вектора S относительно базиса $\{s_1, s_2, \ldots, s_n\}$.

Рассмотрим граф $G=(V, E)$. Набор всех подмножеств множества E, включая и пустое множество $\varnothing$, обозначим через W_G. Сначала покажем, что W_G — абелева группа по отношению к операции $\oplus$ (кольцевой суммы множеств). После соответствующего определения операции умножения над элементами $GF(2)$ и элементами W_G покажем, что W_G является векторным пространством поля $GF(2)$. Нетрудно проверить следующие утверждения:

1) W_G — замкнутое множество по отношению к операции $\oplus$;
2) $\oplus$ — ассоциативная операция;
3) $\oplus$ — коммутативная операция.

Далее, для любого элемента E_i в наборе W_G справедливы равенства $E_i \oplus \varnothing = E_i$ и $E_i \oplus E_i = \varnothing$.

Таким образом, для операции $\oplus$ множество $\varnothing$ — нулевой элемент, а каждый элемент E_i — элемент, обратный самому себе. Следовательно, набор W_G — абелева группа по отношению к операции $\oplus$ и поэтому удовлетворяет первому требованию определения векторного пространства.

Пусть $*$ — операция умножения над элементами $GF(2)$ и W_G — определяется следующим образом:

Для любого E_i в W_G $1*E_i=E_i$ и $0*E_i=\varnothing$. Пользуясь таким определением операции $*$, нетрудно убедиться в том, что элементы W_G удовлетворяют следующим необходимым условиям векторного пространства: для любых элементов α и β, принадлежащих $GF(2)=\{0, 1\}$ любых E_i и E_j множества W_G, справедливы равенства:

1. $(\alpha+\beta)*E_i=(\alpha*E_i)\oplus(\beta*E_i)$.
2. $\alpha*(E_i \oplus E_j)=(\alpha*E_i)\oplus(\alpha*E_j)$.
3. $(\alpha\cdot\beta)*E_i=\alpha*(\beta*E_i)$.
4. $1*E_i=E_i$.

[Отметим, что 1 — нулевой мультипликативный элемент поля $GF(2)$.] Таким образом, W_G — векторное пространство над полем $GF(2)$. Если $E=\{e_1, e_2, \ldots, e_m\}$, то подмножества $\{e_1\}$, $\{e_2\}$. . .$\{e_m\}$ образуют базис для W_G. Следовательно, размерность W_G равна m, т. е. числу ребер в графе G. Из того, что каждый реберно-порожденный подграф графа G соответствует единственному подмножеству множества E и что кольцевой сумме любых 2-реберно-порожденных подграфов по определению (гл. 1) можно поставить в соответствие кольцевую сумму двух соответствующих множеств ребер, следует, что множество всех реберно-порожденных подграфов графа G является векторным пространством над $GF(2)$, если операция умножения $*$ определена следующим образом: для любого реберно-порожденного подграфа G_i графа G $1*G_i=G_i$ и $0*G_i=\varnothing$ — нуль-граф, не имеющий ни вершин, ни ребер. Это векторное пространство также обозначим символом W_G. Заметим, что W_G включает в себя нуль-граф $\varnothing$. Из приведенных рассуждений вытекает следующая теорема:

Теорема 4.2. Для графа G пространство W_G является m-мерным векторным пространством над полем $GF(2)$.

Поскольку в этой главе мы рассматриваем только реберно-порожденные подграфы, то будем называть их просто подграфами, опуская слово «реберно-порожденные». Однако мы можем по-прежнему использовать это уточнение в некоторых местах, чтобы подчеркнуть реберно-порожденную природу рассматриваемых подграфов. Теперь покажем, что следующие подмножества W_G являются подпространствами:

1) W_C — множество всех циклов (включая и нуль-граф $\varnothing$) и объединений реберно-непересекающихся циклов графа G;
2) W_S — множество всех разрезающих множеств (включая и нуль-граф $\varnothing$) и объединений реберно-непересекающихся разрезающих множеств графа G.

Если мы покажем, что W_C и W_S являются замкнутыми по отношению к операции сложения $\oplus$, то из этого и будет следовать, что W_C и W_S — подпространства.

Теорема 4.3. Множество W_C всех циклов и объединений реберно-непересекающихся циклов графа G является подпространством векторного пространства W_G графа G.

Доказательство. По теореме 3.1 граф можно представить в виде объединения реберно-непересекающихся циклов тогда и только тогда, когда каждая вершина в графе имеет четную степень. Следовательно, W_C можно рассматривать как множество всех реберно-порожденных подграфов графа G, все вершины которого имеют четную степень.

Рассмотрим два любых элемента C_1 и C_2 множества W_G. Они являются реберно-порожденными подграфами, вершины которых имеют четную степень. Пусть C_3 — сумма по модулю для элементов C_1 и C_2. Для доказательства этой теоремы необходимо показать, что сумма C_3 принадлежит множеству W_C. Иными словами, надо показать, что каждая вершина в C_3 имеет четную степень.

Рассмотрим любую вершину v, принадлежащую C_3. Очевидно, эта вершина должна присутствовать по крайней мере в одном из подграфов: C_1 или C_2. Обозначим через X_i ($i=1, 2, 3$) множество ребер, инцидентных вершине v в подграфе C_i, а через $|X_i|$ — число ребер в множестве X_i. Таким образом, $|X_i|$ — степень вершины v в подграфе C_i. Заметим, что числа $|X_1|$ и $|X_2|$ — четные и одно из них может равняться нулю. Следовательно, $|X_3| \neq 0$. Поскольку $C_3 = C_1 \oplus C_2$, получаем, что $X_3 = X_1 \oplus X_2$. Поэтому $|X_3| = |X_1| + |X_2| - 2|X_1 \cap X_2|$. Из этого равенства видно, что $|X_3|$ является четной степенью, потому что обе степени $|X_1|$ и $|X_2|$ — четные, другими словами, вершина v в C_3 имеет четную степень. А так как вершину v мы выбрали произвольно, то C_3 принадлежит W_C, и теорема доказана. Назовем W_C *подпространством циклов* графа G.

В качестве примера, иллюстрирующего предыдущую теорему, рассмотрим граф, изображенный на рис. 4.1, *а*. Подграфы C_1 и C_2 (рис. 4.1, *б*, *в*) являются объединениями реберно-непересекающихся циклов G, поскольку каждая вершина этих подграфов имеет четную степень. Эти подграфы принадлежат подпространству W_C пространства W_G. Кольцевая сумма C_1 и C_2 представлена на рис. 4.1, *г* в виде подграфа C_3. Все его вершины имеют четную степень, следовательно, подграф C_3 также принадлежит подпространству W_C.

Покажем теперь, что множество W_S всех разрезающих множеств и объединений реберно-непересекающихся разрезающих множеств графа G является подпространством пространства W_G. По теореме 2.7 разрез является разрезающим множеством или объединением нескольких реберно-непересекающихся разрезающих множеств. Таким образом, каждый разрез графа G принадлежит множеству W_S. Докажем, что каждый элемент последнего является разрезом. Одновременно покажем, что W_S — подпространство пространства W_G.

Теорема 4.4. Сумма двух любых разрезов графа G также является разрезом графа G.

Доказательство. Рассмотрим два любых разреза: $S_1=\langle V_1, V_2\rangle$ и $S_2=\langle V_3, V_4\rangle$ графа $G=(V, E)$. Заметим, что $V_1 \cup V_2=V_3 \cup V_4=V$ и $V_1 \cap V_2=V_3 \cap V_4=\varnothing$. Пусть $A=V_1 \cap V_3$, $B=V_1 \cap V_4$, $C=V_2 \cap V_3$, $D=V_2 \cap V_4$.

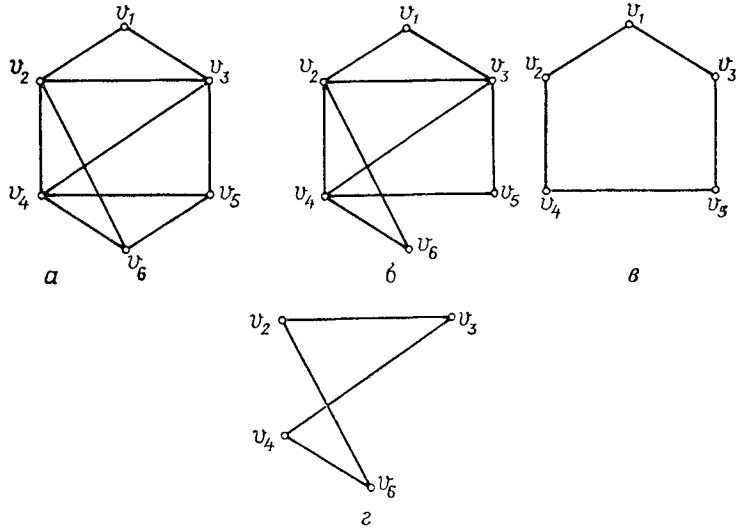


Рис. 4.1.

а — граф G; *б* — подграф C_1 графа G; *в* — подграф C_2 графа G; *г* — подграф $C_1 \oplus C_2$ графа G.

Нетрудно видеть, что множества A, B, C, D взаимно не пересекаются. Тогда $S_1=\langle A \cup B, C \cup D\rangle=\langle A, C\rangle \cup \langle A, D\rangle \cup \langle B, C\rangle \cup \langle B, D\rangle$ и $S_2=\langle A \cup C, B \cup D\rangle=\langle A, B\rangle \cup \langle A, D\rangle \cup \langle C, B\rangle \cup \langle C, D\rangle$.

Следовательно, получим $S_1 \oplus S_2=\langle A, C\rangle \cup \langle B, D\rangle \cup \langle A, B\rangle \cup \langle C, D\rangle$.

Поскольку $\langle A \cup D, B \cup C\rangle=\langle A, C\rangle \cup \langle B, D\rangle \cup \langle A, B\rangle \cup \langle C, D\rangle$, можно записать $S_1 \oplus S_2=\langle A \cup D, B \cup C\rangle$.

Из того, что $A \cup D$ и $B \cup C$ взаимно не пересекаются и вместе включают в себя все вершины V, следует, что $S_1 \oplus S_2$ является разрезом в графе G. Теорема доказана. Так как кольцевая сумма двух реберно-непересекающихся графов совпадает с их объединением, имеет место следующее следствие:

Следствие 4.4.1. Объединение любых двух реберно-непересекающихся разрезов графа G также является его разрезом. Поскольку разрезающее множество является разрезом, то из следствия 4.4.1 очевидно, что W_S — множество всех разрезов графа G. Далее, по теореме 4.4 множество W_S является замкнутым по отношению к операции кольцевой суммы. Таким образом, имеет место следующая теорема:

Теорема 4.5. Множество W_S всех разрезающих множеств и объединений реберно-непересекающих множеств графа G является подпространством векторного пространства W_G графа G. Назовем W_S *подпространством разрезов* графа G. В качестве примера, иллюстрирующего теорему 4.5, рассмотрим разрезы S_1 и S_2 графа G (рис. 4.2): $S_1=\{e_1, e_3, e_4, e_5, e_6, e_7\}=\langle V_1, V_2\rangle$, $S_2=\{e_1, e_2, e_4, e_5, e_8\}=\langle V_3, V_4\rangle$, где $V_1=\{v_1, v_2, v_4\}$, $V_2=\{v_3, v_5, v_6\}$, $V_3=\{v_1, v_4, v_5\}$, $V_4=\{v_2, v_3, v_6\}$.

Тогда $S_1 \oplus S_2=\{e_2, e_3, e_6, e_7, e_8\}$.

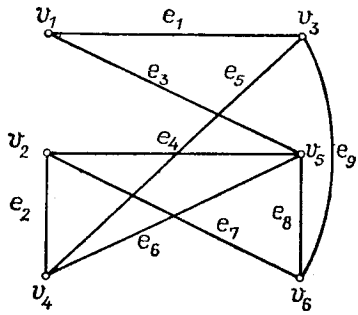


Рис. 4.2.

Если $A=V_1 \cap V_3=\{v_1, v_4\}$, $B=V_1 \cap V_4=\{v_2\}$, $C=V_2 \cap V_3=\{v_5\}$, $D=V_2 \cap V_4=\{v_3, v_6\}$, то множество $S_1 \oplus S_2=\{e_2, e_3, e_6, e_7, e_8\}$ можно записать (как было доказано в теореме 4.4) в таком виде: $S_1 \oplus S_2=\langle A \cup D,\ B \cup C\rangle=\langle\{v_1, v_4, v_3, v_6\}, \{v_2, v_5\}\rangle$.

4.4. Размерность подпространств циклов и разрезов

В этом разделе мы покажем, что размерность подпространств циклов и размерность разрезов соответственно равны цикломатическому числу и рангу графа. Для этого докажем, что множество базисных циклов и множество базисных разрезающих множеств по отношению к некоторому остову связного графа являются соответственно базисами для подпространства циклов и подпространства разрезов графа.

Пусть T — остов связного графа G на n вершинах и m ребрах. Ветви T обозначим через $b_1, b_2, \ldots, b_{n-1}$, а хорды — через $c_1, c_2, \ldots, c_{m-n+1}$. Пусть C_i и S_i — базисный цикл и разрезающее множество по отношению к c_i и b_i соответственно.

По определению каждый базисный цикл включает в себя только одну хорду, не присутствующую ни в одном другом базисном цикле. Таким образом, никакой базисный цикл нельзя представить в виде кольцевой суммы других базисных циклов. Следовательно, базисные циклы $C_1, C_2, \ldots, C_{m-n+1}$ независимы. Аналогично базисные разрезающие множества $S_1, S_2, \ldots, S_{n-1}$ также являются независимыми, так как каждое из них содержит только одну ветвь, не присутствующую ни в одном другом.

Чтобы доказать, что $C_1, C_2, \ldots, C_{m-n+1}(S_1, S_2, \ldots, S_{n-1})$ образуют базис подпространства циклов (разрезов) графа G, необходимо показать, что любой подграф подпространства циклов (разрезов) графа G можно представить в виде кольцевой суммы циклов C_i (разрезов S_i). Рассмотрим любой подграф C подпространства циклов графа G. Пусть он содержит хорды $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$. Пусть

C' — кольцевая сумма базисных циклов $C_{i_1}, C_{i_2}, \ldots, C_{i_r}$. Очевидно, что C' содержит только хорды $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ и не содержит других хорд остова T. Поскольку подграф C также содержит только эти хорды, то $C' \oplus C$ не содержит ни одной хорды. Покажем, что $C' \oplus C$ — пустое множество. Если это неверно, то по предыдущим рассуждениям множество $C' \oplus C$ содержит только ветви и, следовательно, не содержит цикла. С другой стороны, так как множество $C' \oplus C$ — кольцевая сумма циклов, то по теореме 4.3 оно является либо циклом, либо объединением нескольких реберно-непересекающихся циклов. Получаем противоречие. Следовательно, $C' \oplus C$ — пустое множество. Из этого следует, что $C = C' = C_{i_1} \oplus C_{i_2} \oplus \ldots \oplus C_{i_r}$. Другими словами, каждый подграф подпространства циклов графа G можно представить в виде кольцевой суммы циклов C_i. Точно таким же образом мы можем доказать, что каждый подграф подпространства разрезов графа G можно представить в виде кольцевой суммы разрезов S_i. Итак, имеет место следующая теорема:

Теорема 4.6. Пусть G — связный граф на n вершинах и m ребрах. В этом случае

1. Базисные циклы по отношению к остову графа G образуют базис подпространства циклов графа, и, следовательно, размерность подпространства циклов графа G равна $m-n+1$, т. е. цикломатическому числу графа G.
2. Базисные разрезающие множества по отношению к остову графа G образуют базис подпространства разрезов графа G, и, следовательно, размерность подпространства разрезов графа равна $n-1$, т. е. рангу графа G.

Теперь нетрудно видеть, что в случае несвязного графа G множества всех базисных циклов по отношению к хордам леса графа G и разрезающих множеств

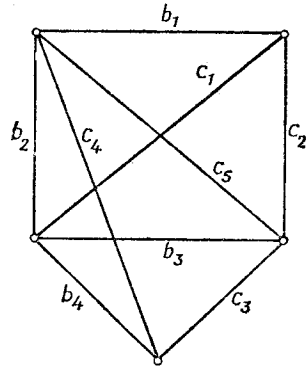


Рис. 4.3.

по отношению к ветвям леса графа G являются базисами подпространств циклов и разрезов графа G соответственно.

Таким образом, получаем следствие из предыдущей теоремы:

Следствие 4.6.1. Если граф на n вершинах и m ребрах имеет p компонент, то

1. Размерность подпространства циклов графа G равна $m-n+p$, т. е. цикломатическому числу графа G.

2. Размерность подпространства разрезов графа G равна $n-p$, т. е. рангу графа G.

В качестве примера рассмотрим граф G, изображенный на рис. 4.3. Ребра b_1, b_2, b_3, b_4 образуют остов графа G. Хордами остова T являются c_1, c_2, c_3, c_4, c_5. Базисные циклы C_1, C_2, C_3, C_4, C_5 по отношению к хордам c_1, c_2, c_3, c_4, c_5 и базисные разрезающие множества S_1, S_2, S_3, S_4 по отношению к ветвям b_1, b_2, b_3, b_4 определяются следующим образом:

$$C_1=\{c_1, b_1, b_2\} \qquad S_1=\{b_1, c_1, c_2\},$$
$$C_2=\{c_2, b_1, b_2, b_3\}, \qquad S_2=\{b_2, c_1, c_2, c_4, c_5\},$$
$$C_3=\{c_3, b_3, b_4\}, \qquad S_3=\{b_3, c_2, c_3, c_5\},$$
$$C_4=\{c_4, b_2, b_4\}, \qquad S_4=\{b_4, c_3, c_4\}.$$
$$C_5=\{c_5, b_2, b_3\}.$$

Рассмотрим сначала подграф C, состоящий из ребер $b_1, b_2, b_3, c_2, c_3, c_4, c_5$. Нетрудно убедиться в том, что в подграфе C каждая вершина имеет четную степень и, следовательно, подграф C принадлежит подпространству циклов графа G. Поскольку хорды c_2, c_3, c_4, c_5 принадлежат подграфу C, из доказательства теоремы 4.6 следует, что подграф C должен равняться кольцевой сумме базисных циклов C_2, C_3, C_4, C_5. Убедимся в том, что это действительно так:

$$C_2 \oplus C_3 \oplus C_4 \oplus C_5 = \{c_2, b_1, b_2, b_3\} \oplus \{c_3, b_3, b_4\} \oplus \{c_4, b_2, b_4\} \oplus \{c_5, b_2, b_3\} = \{b_1, b_2, b_3, c_2, c_3, c_4, c_5\} = C.$$

Теперь рассмотрим разрез S состоящий из ребер b_1, b_3, c_1, c_3, c_5. Снова, поскольку S содержит ветви b_1 и b_3, он должен равняться кольцевой сумме разрезающих множеств S_1 и S_3. Это нетрудно проверить: $S_1 \oplus S_3 = \{b_1, c_1, c_2\} \oplus \{b_3, c_2, c_3, c_5\} = S$.

В предыдущих рассуждениях мы показали, что, имея остов, можно построить базисы подпространств циклов и разрезов графа. Построенные таким образом базисы обычно используются при изучении электрических цепей.

4.5. Связь между подпространствами циклов и разрезов

В данном разделе мы дадим характеризацию подграфов в пространстве циклов графа G в терминах подграфов подпространства разрезов того же графа.

В разд. 2.8 (теорема 2.14) мы доказали, что цикл и разрезающее множество имеют четное количество общих ребер. Используя то, что каждый подграф в подпространстве циклов графа является либо циклом, либо объединением реберно-непересекающихся циклов, а каждый подграф в подпространстве разрезов — разрезающим множеством или объединением реберно-непересекающихся разрезающих множеств, получим следующую теорему как непосредственное следствие теоремы 2.14:

Теорема 4.7. *Любой подграф в подпространстве циклов графа G имеет четное число общих ребер с любым подграфом в подпространстве разрезов того же графа.*

В теореме 4.8 мы докажем обратное утверждение.

Теорема 4.8.

1. *Подграф графа G принадлежит подпространству циклов графа G, если он имеет четное число общих ребер с любым подграфом в подпространстве разрезов графа G.*
2. *Подграф графа G принадлежит подпространству разрезов графа G, если он имеет четное число общих ребер с любым подграфом в подпространстве циклов графа G.*

Доказательство. 1. Не нарушая общности, можно предложить, что G — связный граф. Аналогично доказывается случай, когда граф G несвязный. Пусть T — остов графа G. Обозначим ветви остова T через $b_1, b_2, \ldots$, а хорды — через $c_1, c_2, \ldots$. Рассмотрим любой подграф C графа G, имеющий четное число общих ребер с любым подграфом в подпространстве разрезов графа G. Предположим, не нарушая общности, что подграф C содержит хорды $c_1, c_2, \ldots, c_r$. Обозначим через C' кольцевую сумму базисных циклов $C_1, C_2, \ldots, C_r$ по отношению к хордам $c_1, c_2, \ldots, c_r$. Очевидно, что C' содержит только хорды $c_1, c_2, \ldots, c_r$. Следовательно, $C' \oplus C$ не содержит ни одной хорды. Поскольку C' — кольцевая сумма нескольких циклов графа G, то это множество имеет четное число общих ребер с каждым подграфом в подпространстве разрезов графа G. Из того, что C также обладает этим свойством, следует, что им обладает и $C' \oplus C$. Теперь убедимся, что $C' \oplus C$ — пустое множество. Если это не так, то $C' \oplus C$ состоит только из ветвей. Пусть b_i — любая ветвь в $C' \oplus C$. Тогда b_i — единственное общее ребро между $C' \oplus C$ и базисным разрезающим множеством по отношению к b_i. Однако это невозможно, так как $C' \oplus C$ должно иметь четное число общих ребер с любым разрезающим множеством. Таким образом, $C' \oplus C$ должно быть пустым. Другими словами $C = C' = C_1 \oplus C_2 \oplus \ldots \oplus C_r$ и, следовательно, C принадлежит подпространству циклов графа G.

2. Доказательство этой части теоремы аналогично.

4.6. Ортогональность подпространств циклов и разрезов

Согласно теореме 4.1, каждое n-мерное векторное пространство над полем F изоморфно векторному пространству всех n-векторов над тем же полем. Следовательно, векторное пространство W_G графа G изоморфно векторному пространству всех m-векторов над полем $GF(2)$, где m — число ребер графа G.

Пусть $e_1, e_2, \ldots, e_m$ — ребра графа G. Предположим, что мы сопоставили каждому реберно-порожденному подграфу G_i графа G такой m-вектор w_i, что j-й элемент w_i равен 1 тогда и только тогда, когда ребро e_j принадлежит подграфу G_i. Тогда кольцевая сумма $G_i \oplus G_j$ двух подграфов G_i и G_j будет соответствовать m-вектору $w_i + w_j$, являющемуся суммой по mod 2 векторов w_i и w_j. Легко видеть, что описанное соответствие действительно определяет изоморфизм между W_G и векторным пространством всех m-векторов над полем $GF(2)$. В самом деле, если мы выберем $\{e_1\}, \{e_2\}, \ldots, \{e_m\}$ в качестве базисных векторов для пространства W_G, то элементами w_i будут координаты G_i, связанные с этим базисом.

При определении этого изоморфизма мы опять использовали символ W_G для обозначения векторного пространства всех m-векторов, сопоставленных подграфам графа G. Пусть W_C обозначает подпространство m-векторов, представляющих подграфы в подпространстве циклов графа G, а W_S — подпространство, представляющее подграфы в подпространстве разрезов графа G.

Рассмотрим два таких вектора w_i и w_j, что вектор w_i находится в пространстве W_C, а вектор w_j — в пространстве W_S. Из того, что любой подграф в пространстве W_C имеет четное число общих ребер с произвольным подграфом в пространстве W_S, следует, что скаляр-

ное произведение $\langle w_i, w_j \rangle$ векторов w_i и w_j равно сумме по mod 2 четного числа единиц. Это означает, что $\langle w_i, w_j \rangle = 0$. Иначе говоря, m-векторы в пространстве W_C ортогональны подобным векторам в пространстве W_S. Таким образом, имеет место следующая теорема:

Теорема 4.9. Подпространства циклов и разрезов графа ортогональны.

Рассмотрим теперь прямую сумму $W_C \boxplus W_S$. Мы знаем, что $\dim(W_C \boxplus W_S) = \dim(W_C) + \dim(W_S) - \dim(W_C \cap W_S)$. Поскольку $\dim(W_C) + \dim(W_S) = m$, получаем $\dim(W_C \boxplus W_S) = m - \dim(W_C \cap W_S)$. Теперь ортогональные подпространства W_C и W_S будут также и ортогональными дополнениями W_G тогда и только тогда, когда $\dim(W_C \boxplus W_S) = m$. Иными словами, W_C и W_S будут ортогональными дополнениями в том и только в том случае, если $\dim(W_C \cap W_S) = 0$, т. е. $W_C \cap W_S$ — нулевой вектор (все элементы которого равны нулю). Поэтому мы получаем следующую теорему:

Теорема 4.10. Подпространства W_C и W_S циклов и разрезов графа являются ортогональными дополнениями тогда и только тогда, когда $W_C \cap W_S$ — нулевой вектор.

Пусть W_C и W_S — ортогональные дополнения. Это означает, что каждый вектор в пространстве W_G можно представить кольцевой суммой $w_i + w_j$, где вектор w_i принадлежит пространству W_C, а вектор w_j — пространству W_S. Другими словами, каждый подграф графа G можно представить кольцевой суммой

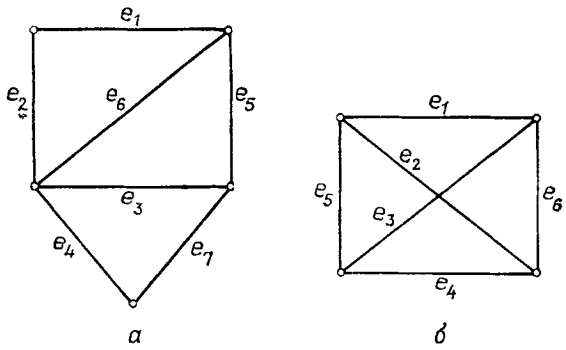


Рис. 4.4.

а — граф G_a; *б* — граф G_b.

двух подграфов, один из которых принадлежит подпространству циклов, а другой — подпространству разрезов. В частности, сам граф G можно представить таким же образом.

Предположим, что W_C и W_S не являются ортогональными дополнениями. Тогда, очевидно, существует такой подграф, который нельзя представить как кольцевую сумму подграфов в пространствах W_C и W_S. Возникает вопрос: можно ли в этом случае представить граф G кольцевой суммой подграфов, принадлежащих пространствам W_C и W_S? Ответом является следующая теорема:

Теорема 4.11. *Любой граф G можно представить в виде кольцевой суммы двух подграфов, один из которых принадлежит подпространству циклов, а другой — подпространству разрезов графа G.* Доказательство этой теоремы можно найти в работах [44, 45].

Завершим этот раздел разбором примера.

Рассмотрим граф G_a, представленный на рис. 4.4, *а*. Нетрудно убедиться, что ни один непустой подграф этого графа не принадлежит пересечению подпро-

странств циклов и разрезов. Следовательно, эти подпространства графа G_a являются ортогональными дополнениями. Поэтому множество базисных циклов и базисных разрезающих множеств по отношению к некоторому остову графа G_a образует базис векторного пространства того же графа. Ниже приводится одно из множеств по отношению к остову, образованному ребрами e_1, e_2, e_3, e_4:

векторы базисных разрезающих множеств

$$S_1 = (1\ 0\ 0\ 0\ 1\ 1\ 0)$$
$$S_2 = (0\ 1\ 0\ 0\ 1\ 1\ 0)$$
$$S_3 = (0\ 0\ 1\ 0\ 1\ 0\ 1)$$
$$S_4 = (0\ 0\ 0\ 1\ 0\ 0\ 1)$$

базисные циклические векторы

$$C_1 = (1\ 1\ 1\ 0\ 1\ 0\ 0)$$
$$C_2 = (1\ 1\ 0\ 0\ 0\ 1\ 0)$$
$$C_3 = (0\ 0\ 1\ 1\ 0\ 0\ 1)$$

Легко видеть, что каждый вектор можно представить в виде кольцевой суммы циклического вектора и вектора разрезающего множества. В частности, вектор (111111), представляющий граф G_a, сам может записываться в виде (1111111)= $= S_1 \oplus S_2 \oplus S_4 \oplus C_1 \oplus C_2 =$ (1101001)$\oplus$(0010110), где (1101001) представляет собой разрез, а (0010110) — цикл в графе G_a. Далее, рассмотрим граф G_b, представленный на рис. 4.4, *б*. В этом графе ребра e_1, e_2, e_3, e_4 образуют разрез и цикл. Следовательно, подпространства циклов разрезов этого графа не являются ортогональными дополнениями. Это означает, что существует некоторый подграф в графе G_b, который нельзя представить суммой подграфов из подпространства циклов и разрезающего подпространства графа G_b. Однако по теореме 4.11 возможно такое разложение для графа G_b. Это справедливо, поскольку $G_b = \{e_1, e_2, e_5\} \oplus \oplus \{e_3, e_4, e_6\}$, где $\{e_1, e_2, e_3\}$ — разрез, а $\{e_3, e_4, e_6\}$ — цикл в графе G_b

4.7. Замечания, касающиеся литературы

Ранней работой по векторным пространствам, связанным с графом, является работа [4, 6], в которой рассматривается вопрос построения графа с заданным множеством циклов. Для дальнейшего изучения этой темы рекомендуются также работы [4.4, 4.5].

Упражнения

4.1. Покажите, что циклы, определяемые следующими множествами, образуют базис подпространства циклов графа, представленного на рис. 4.5: $\{e_1, e_3, e_4, e_6\}$; $\{e_2, e_3, e_5, e_7\}$; $\{e_1, e_2, e_8\}$.

4.2. *Множество инциденций* — множество ребер, инцидентных вершине. Покажите, что любая из $n-1$ множеств инциденций связного графа на n вершинах образуют базис подпространства разрезов графа.

4.3. Покажите, что если ρ — ранг, а μ — цикломатическое число графа G, то

а) число различных базисов для подпространства разрезов графа G равно $\frac{1}{\rho!}(2^\rho - 2^0)(2^\rho - 2^1)(2^\rho - 2^2)\ldots(2^\rho - 2^{\rho-1})$;

б) число различных базисов для подпространства циклов графа G равно $\frac{1}{\mu!}(2^\mu - 2^0)(2^\mu - 2^1)(2^\mu - 2^2)\ldots(2^\mu - 2^{\mu-1})$.

4.4. Для графа G, представленного на рис. 4.6, найдите

а) множество базисных векторов подпространства циклов, которые не являются базисными циклическими векторами по отношению к какому-либо остову графа G;

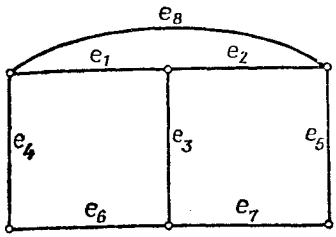


Рис. 4.5.

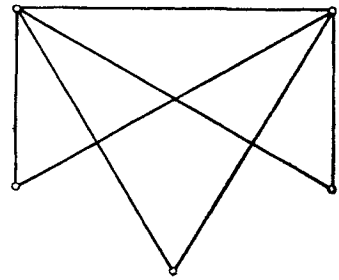


Рис. 4.6.

б) множество базисных векторов подпространства разрезов, которые не соответствуют ни множествам инциденций, ни базисным разрезающим множествам по отношению к какому-либо остову графа G.

4.5. а) Проверьте, являются ли подпространства циклов и разрезов графа G (рис. 4.6) ортогональными дополнениями векторного пространства графа G;

б) представьте граф G кольцевой суммой двух подграфов, один из которых принадлежит подпространству циклов, а другой — подпространству разрезов.

4.6. Покажите, что каждое подмножество множества ребер дерева является разрезом дерева.

4.7. Покажите, что подграф графа имеет четное число ребер, если он принадлежит одновременно подпространствам как циклов, так и разрезов графа.

4.8. Подмножество ребер E' графа G называется независимым, если оно не содержит циклов. Докажите следующее:

а) каждое подмножество независимого множества является независимым;

б) если I и J независимые множества, содержащие k и $k+1$ ребер соответственно, то существует такое ребро e, входящее только в множество J, что $I \cup \{e\}$ — независимое множество.

4.9. Выполните упражнение 4.8, заменив термин «цикл» термином «разрезающее множество».

5. Ориентированные графы

В предыдущих главах мы представили некоторые основные результаты теории неориентированных графов. Однако для описания некоторых ситуаций неориентированных графов недостаточно. Например, при представлении схемы уличного движения графом, ребра которого соответствуют улицам, для указания допустимого направления движения ребрам необходимо присваивать ориентацию. Другим примером является программа для ЭВМ, моделируемая графом, ребра которого представляют поток управления от одних множеств инструкций к другим. В таком представлении программы для указания направления потока управления ребрам также необходимо присвоить ориентацию. Еще одним примером физической системы, для представления которой требуется ориентированный граф, является электрическая цепь. Применения ориентированных графов и соответствующие алгоритмы рассматриваются в гл. 11—15.

В этой главе предлагаются вниманию основные результаты теории ориентированных графов. Обсуждаются вопросы, связанные с существованием ориентированных эйлеровых цепей и гамильтоновых циклов. Рассматриваются также ориентированные деревья и их связь с ориентированными эйлеровыми цепями.

5.1. Основные определения и понятия

Начнем с введения нескольких основных определений и понятий, относящихся к ориентированным графам.

Ориентированный граф $G=(V, E)$ состоит из двух множеств: конечного множества V, элементы которого называются *вершинами*, и конечного множества E, элементы которого называются *ребрами* или дугами. Каждая дуга связана с упорядоченной парой вершин.

Для обозначения вершин используются символы $v_1, v_2, \ldots$, а для обозначения дуг — символы $e_1, e_2, \ldots$. Если $e_l=(v_i, v_j)$, то v_i и v_j называются *концевыми вершинами* e_l, при этом v_i — *начальная вершина*, а v_j — конечная вершина e_l. Все дуги имеющие одну пару начальных и конечных вершин, называются *параллельными*. Дуга называется *петлей*, если инцидентная вершина v_i является одновременно начальной и конечной ее вершиной.

В графическом представлении ориентированного графа вершины изображаются точками или кружками, а ребра (дуги) — отрезками

линий, соединяющими точки или кружки, представляющие их концевые вершины. Кроме того, дугам присваивается ориентация, показываемая стрелкой, направленной от начальной вершины к конечной.

Например, если $V=\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ и $E=\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$ такие, что $e_1=(v_1, v_2)$, $e_2=(v_1, v_2)$, $e_3=(v_1, v_3)$, $e_4=(v_3, v_1)$, $e_5=(v_2, v_4)$, $e_6=(v_3, v_4)$, $e_7=(v_4, v_4)$, $e_8=(v_5, v_6)$, ориентированный граф $G=(V, E)$ можно представить рис. 5.1. В этом графе e_1 и e_2 — параллельные дуги, а e_7 — петля.

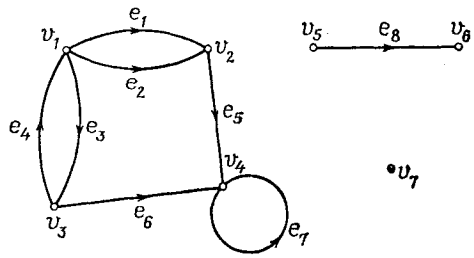


Рис. 5.1. Ориентированный граф.

Говорят, что дуга *инцидентна* своим концевым вершинам. Вершины называются *смежными*, если они являются концевыми для одной дуги. Если дуги имеют общую концевую вершину, то они называются *смежными*.

Дуга называется *исходящей из* своей начальной вершины и *заходящей в* свою конечную вершину. Верщина называется *изолированной*, если она не имеет инцидентных дуг.

Степенью $d(v_j)$ вершины v_j называется число инцидентных ей дуг. *Полустепенью захода* $d^-(v_j)$ вершины v_j является число заходящих в v_j дуг, а *полустепенью исхода* $d^+(v_j)$ — число исходящих из v_j дуг. Символами δ^+ и δ^- обозначают минимальные полустепени исхода и захода ориентированного графа. Аналогично символами Δ^+ и Δ^- обозначают максимальные полустепени исхода и захода соответственно.

Множества $\Gamma^+(v)$ и $\Gamma^-(v)$ для любой вершины определяются следующим образом: $\Gamma^+(v)=\{w \mid (v, w) \in E\}$, $\Gamma^-(v)=\{w \mid (w, v) \in E\}$. Например, в графе на рис. 5.1 $\Gamma^+(v_1)=\{v_2, v_3\}$, $\Gamma^-(v_4)=\{v_2, v_3, v_4\}$.

Заметим, что петля увеличивает полустепени как захода, так и исхода этой вершины. Следующее утверждение является следствием того, что каждая дуга увеличивает на 1 сумму полустепеней как захода, так и исхода ориентированного графа.

Теорема 5.1. В ориентированном графе с m дугами
Сумма полустепеней захода=Сумма полустепеней исхода=m.

Подграфы и *порожденные* подграфы ориентированного графа определяются так же, как и в случае неориентированных графов (разд. 1.2).

Неориентированный граф, получающийся в результате снятия ориентации с дуг ориентированного графа G, называется *лежащим в основе неориентированного графа* G и обозначается через $G_н$.

Ориентированным маршрутом ориентированного графа $G=(V, E)$ называется такая конечная последовательность вершин

$v_0, v_1, \ldots, v_k$, что (v_{i-1}, v_i), $1 \leqslant i \leqslant k$, является дугой графа G. Такой маршрут обычно называется ориентированным $v_0 - v_k$-маршрутом, причем v_0 — *начальная вершина*, v_k — *конечная вершина* маршрута, а все другие вершины — *внутренние*. Начальная и конечная вершины ориентированного маршрута называются его *концевыми вершинами*. Отметим, что дуги, а следовательно, и вершины могут появляться в ориентированном маршруте более одного раза.

Ориентированный маршрут называется *открытым*, если его концевые вершины различны, в противном случае — *замкнутым*.

Ориентированный маршрут называется *ориентированной цепью*, если все его дуги различны. Ориентированная цепь является *открытой*, если ее концевые вершины различны, в противном случае — *замкнутой*.

Открытая ориентированная цепь называется *ориентированным путем*, если различны все ее вершины.

Замкнутая ориентированная цепь называется *ориентированным циклом* или *контуром*, если ее вершины, за исключением концевых, различны.

Говорят, что ориентированный граф *ациклический* или *бесконтурный*, если он не имеет контуров. Например, ациклическим является ориентированный граф на рис. 5.2.

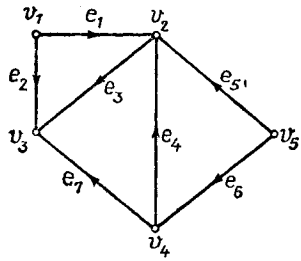


Рис. 5.2. Ациклический ориентированный граф.

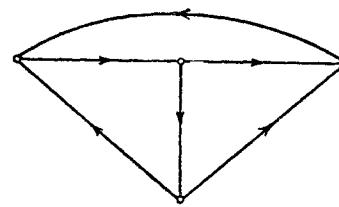


Рис. 5.3. Сильно связный ориентированный граф.

Последовательность вершин ориентированного графа G называется *маршрутом* в G, если она является маршрутом лежащего в основе неориентированного графа $G_н$. Например, последовательность $v_1, v_2, v_3, v_4, v_2, v_3$ в графе на рис. 5.2 является маршрутом, но не ориентированным.

Аналогичным образом определяются *цепь*, *путь* и *цикл* ориентированного графа.

Ориентированный граф называется *связным*, если связным является лежащий в его основе неориентированный граф.

Подграф ориентированного графа G называется *компонентой* графа G, если он является компонентой графа $G_н$.

Вершины v_i и v_j ориентированного графа G называются *сильно связными*, если в G существуют ориентированные пути из v_i в v_j и обратно. Если v_i сильно связна с v_j, то, очевидно, и v_j сильно связна с v_i. Всякая вершина сильно связна сама с собой.

Если вершина v_i сильно связна с вершиной v_j, а v_j — с v_k, то, как легко видеть, верщина v_i сильно связна с вершиной v_k. Следовательно, в этом случае просто говорят, что вершины v_i, v_j и v_k сильно связны.

Ориентированный граф называется *сильно связным*, если сильно связны все его вершины. Например, сильно связным является граф на рис. 5.3.

Максимальный сильно связный подграф ориентированного графа G называется *сильно связной компонентой* графа G. Если ориентированный граф сильно свя́зен, то он имеет единственную сильно связную компоненту, а именно самого себя.

Рассмотрим ориентированный граф $G=(V, E)$. Легко видеть, что всякая его вершина принадлежит точно одной сильно связной компоненте графа G. Следовательно, множества вершин сильно связных компонент образуют разбиение множества вершин V графа G.

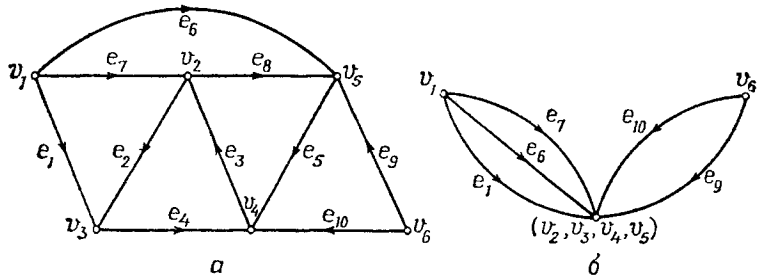


Рис. 5.4. Граф и его конденсация.

Например, ориентированный граф на рис. 5.4, *а* имеет три сильно связные компоненты с множествами вершин $\{v_2, v_3, v_4, v_5\}$, $\{v_1\}$ и $\{v_6\}$, образующими разбиение множества $\{v_1, v_2, v_3, v_4, v_5, v_6\}$ вершин ориентированного графа.

Интересно, что в ориентированном графе могут быть дуги, не входящие ни в какие сильно связные компоненты графа. Например, ни в какие сильно связные компоненты не входят дуги e_1, e_6, e_7, e_9 и e_{10} в графе на рис. 5.4, *а*.

Таким образом, хотя свойство «сильной связности» влечет разбиение множества вершин графа, оно может не порождать разбиение множества дуг.

Объединение, пересечение, сумма по mod 2 и другие операции над ориентированными графами определяются точно так же, как и в случае неориентированных графов (разд. 1.5).

Граф, получающийся в результате стягивания всех дуг сильно связных компонент ориентированного графа G, называется конденсированным графом G_c графа G. Конденсация графа, приведенного на рис. 5.4, *а*, представлена на рис. 5.4, *б*.

Вершины графа G_c соответствуют сильно связным компонентам графа G и называются *конденсированными образами* компонент.

Ранг и *цикломатическое число* ориентированного графа те же, что и у соответствующего неориентированного графа. Это означает, что если ориентированный граф G имеет m дуг, n вершин и p компонент, то ранг ρ и цикломатическое число μ графа G определяются выражениями $\rho=n-p$ и $\mu=m-n+p$.

Теперь определим минимально связные ориентированные графы и изучим некоторые их свойства.

Ориентированный граф G называется *минимально связным*, если он сильно связный, а удаление любой дуги лишает его свойства

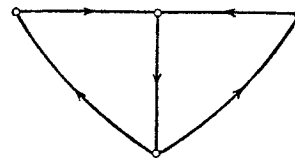


Рис. 5.5. Минимально связный ориентированный граф.

сильной связности. Минимально связным является, например, граф, представленный на рис. 5.5.

Очевидно, что минимально связные графы не могут иметь параллельных дуг и петель.

Мы знаем, что неориентированный граф минимально связен тогда и только тогда, когда он является деревом (упр. 2.13). По теореме 2.5 дерево имеет не менее двух вершин степени 1. Следовательно, минимально связные неориентированные графы имеют по крайней мере две вершины степени 1.

Установим аналогичный результат для ориентированных графов. Степень всякой вершины сильно связного ориентированного графа должна быть не менее 2, поскольку каждая вершина должна иметь исходящую и заходящую дуги. В следующей теореме мы доказываем, что в минимально связном ориентированном графе имеются по крайней мере две вершины степени 2.

Теорема 5.2. Если минимально связный ориентированный граф G имеет более одной вершины, то в нем содержатся по крайней мере две вершины степени 2.

Доказательство. Поскольку граф G — сильно связный и имеет более одной вершины, он должен иметь по крайней мере один контур. Следовательно, для графа G цикломатическое число $\mu \geqslant 1$.

Докажем теорему индукцией по μ.

Если $\mu=1$, то граф G является контуром, следовательно, в этом случае теорема верна.

Пусть теорема верна для всех минимально связных ориентированных графов с $\mu<k$ при $k\geqslant 2$. Рассмотрим минимально связный граф G с $\mu=k$. Покажем, что теорема справедлива и для графа G. Возможны два случая.

Случай 1. Всякий контур G имеет длину 2.

В этом случае любые две смежные вершины соединены точно двумя дугами, имеющими противоположную ориентацию. Пусть G' — простой неориентированный граф на том же множестве вершин, что и граф G, а вершины в нем смежны тогда и только тогда, когда они смежны в графе G. Поскольку граф G — связный, то связен и граф G'. Но так как в графе G нет контуров длины больше 2, то граф G' не имеет циклов. Следовательно, G' — дерево. По теореме 2.5 он имеет по меньшей мере две висячие вершины. В графе G они имеют степень 2, что доказывает теорему для этого случая.

Случай 2. В графе G имеется контур C длиной $l\geqslant 3$. Поскольку граф G — минимально связный, между любыми смежными вершинами контура C имеется единственная дуга, а между любыми не смежными в контуре C вершинами дуг нет.

Пусть G' — граф, получающийся в результате стягивания дуг C. Граф G' имеет $(m-l)$ дуг и $(n-l+1)$ вершин, где m и n — число дуг и вершин графа G соответственно. Поэтому цикломатическое число графа G' равно $(m-l)-(n-l+1)+1=k-1$.

Поскольку граф G' также минимально связный, из индуктивного предположения следует, что в графе G' содержатся хотя бы две вершины v_1 и v_2 степени 2. Если одна из них, например v_1, является конденсированным образом контура C, тогда в нем имеется по крайней мере одна вершина, например v_3, степени 2. В этом случае v_2 и v_3 — вершины степени 2 в графе G.

С другой стороны, если ни v_1, ни v_2 не являются конденсированным образом контура C, то v_1 и v_2 — вершины G степени 2, что доказывает теорему.

Завершаем этот раздел определением свойства «квазисильной связности».

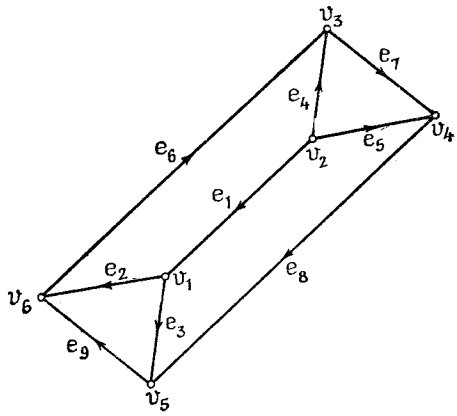


Рис. 5.6. Квазисильно связный граф.

Граф называется *квазисильно связным*, если для любой пары вершин v_1 и v_2 существует вершина v_3, из которой имеются ориентированные пути к вершинам v_1 и v_2. Заметим, что вершина v_3 не обязательно отлична от вершин v_1 и v_2.

Пример квазисильно связного графа представлен на рис. 5.6.

5.2. Графы и отношения

Бинарное отношение на множестве $X=\{x_1, x_2, \ldots\}$ представляет собой набор упорядоченных пар из элементов множества X. Например, если X — множество людей, а R — отношение «является сыном», то упорядоченная пара (x_i, x_j) означает, что x_i является сыном x_j. Этот факт обозначается также x_iRx_j.

Наиболее удачный способ представления бинарного отношения R на множестве X — с помощью ориентированного графа, вершины которого представляют собой элементы множества X, а дуги — упорядоченные пары элементов, определяющие отношение R.

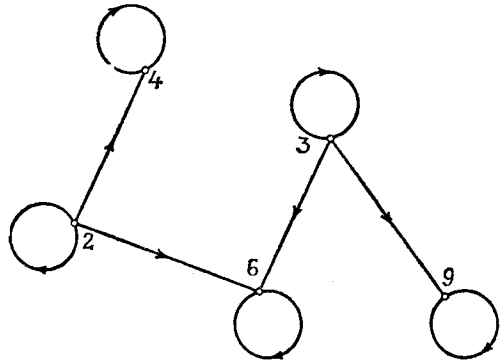


Рис. 5.7. Ориентированный граф, представляющий отношение «является делителем».

Например, на рис. 5.7 приведено представление отношения «является делителем» на множестве $X=\{2, 3, 4, 6, 9\}$.

Рассмотрим множество $X=\{x_1, x_2, \ldots\}$ и отношение R на множестве X:

1) R называется *рефлексивным*, если любой элемент x_i входит в отношение R с самим собой, т. е. для любого x_i x_iRx_i;
2) R называется *симметричным*, если x_iRx_j влечет за собой x_jRx_i;
3) R называется *транзитивным*, если x_iRx_j и x_jRx_k влечет за собой x_iRx_k;
4) R называется *отношением эквивалентности*, если оно рефлексивно, симметрично и транзитивно. Если R — отношение эквивалентности на множестве S, то последнее однозначно разбивается на такие подмножества $S_1, S_2, \ldots, S_k$, что элементы x, y множества S принадлежат S_i тогда, когда xRy. Подмножества $S_1, S_2, \ldots, S_k$ называют классами эквивалентности, порожденными отношением R на множестве S.

Пусть множество X состоит из положительных целых чисел. Тогда

1) отношение «является делителем» рефлексивно и транзитивно;

2) отношение «равно» рефлексивно, симметрично и транзитивно, следовательно, является отношением эквивалентности.

Ориентированный граф, представляющий рефлексивное отношение, называется *рефлексивным ориентированным графом*. Подобным образом определяются *симметричные* и *транзитивные ориентированные графы*. Относительно этих графов можно сделать следующие замечания:

1. В рефлексивном ориентированном графе при каждой вершине имеется инцидентная ей петля.
2. В симметричном ориентированном графе между любыми смежными вершинами имеются две противоположно ориентированные дуги. Поэтому, если ребро поставить в соответствие паре противоположно ориентированных дуг, можно рассматривать неориентированный граф как представление симметричного отношения.
3. Дуга (v_1, v_2) присутствует в транзитивном графе G, если из вершины v_1 имеется ориентированный путь в вершину v_2.

5.3. Ориентированные или корневые деревья

Вершина v ориентированного графа называется *корнем* графа G, если из вершины v существует ориентированный путь ко всем остальным вершинам графа G.

Например, в графе на рис. 5.6 корнем является вершина v_1. Очевидно, что если граф имеет корень, то он квазисильно связный. В следующей теореме мы доказываем, что и из «квазисильной связности» следует существование корня.

Теорема 5.3. Ориентированный граф G имеет корень тогда и только тогда, когда он квазисильно связный.

Доказательство.

Необходимость очевидна.

Достаточность. Рассмотрим вершины $x_1, x_2, \ldots, x_n$ графа G. Поскольку граф G — квазисильно связный, существует вершина y_2, из которой есть ориентированные пути к вершинам x_1 и x_2. По той же причине существует вершина y_3, из которой есть ориентированные пути к вершинам y_2 и x_3. Очевидно, что вершина y_3 связана также ориентированными путями через вершину y_2 с вершинами x_1 и x_2. Продолжая рассуждать подобным образом, придем к тому, что существует вершина y_n, из которой есть ориентированные пути к вершинам y_{n-1} и x_n. Очевидно, что вершина y_n — корень, поскольку она связана через вершину y_{n-1} с вершинами $x_1, \ldots, x_{n-1}$.

Ориентированный граф G называется *деревом*, если лежащий в его основе неориентированный граф также является деревом.

Ориентированный граф G называется *ориентированным* или *корневым* деревом, если он является деревом и имеет корень. Вершины графа G с нулевой полустепенью исхода называются *листьями*.

Примером ориентированного дерева является граф, приведенный на рис. 5.8. Вершина v_1 в нем является корнем, и притом единственным.

В следующей теореме даются эквивалентные характеризации ориентированных деревьев.

Теорема 5.4. Пусть G — ориентированный граф на $n>1$ вершинах. Тогда следующие утверждения эквивалентны:

1) G — ориентированное дерево;
2) В графе G имеется вершина r, из которой есть только один ориентированный путь в любую другую вершину графа G;

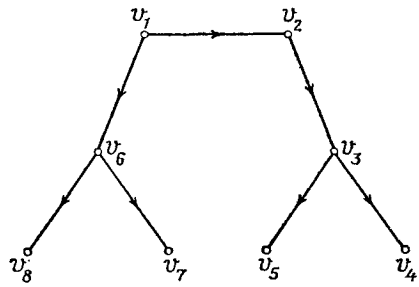


Рис. 5.8. Ориентированное дерево.

3) Граф G — квазисильно связный и теряет это свойство при удалении из него любой дуги;
4) Граф G — квазисильно связный и имеет такую вершину r, что $d^-(r)=0$ и $d^-(v)=1$, $v\neq r$;
5) Граф G не имеет циклов (не обязательно контуров) и содержит такую вершину r, что $d^-(r)=0$ и $d^-(v)=1$, $v\neq r$.

6. Граф G — квазисильно связный без циклов.

Доказательство.

$1\Rightarrow 2$. Корень r графа G обладает требуемыми свойствами.

$2\Rightarrow 3$. Очевидно, что граф G — квазисильно связный. Допустим, что это свойство не теряется при удалении из графа G дуги (u, v). Тогда существует такая вершина z, что из нее имеются ориентированные пути к вершинам u и v, не использующие дугу (u, v). Таким образом, в графе G имеется по два ориентированных пути из вершины z к вершине v, а также из вершины r к вершине v, что противоречит утверждению 2.

$3\Rightarrow 4$. Поскольку граф G — квазисильно связный, он имеет корень r. Поэтому для всякой вершины $v\neq r$ справедливо неравенство $d^-(v)\geqslant 1$. Предположим, что для некоторой вершины v справедливо, что $d^-(v)>1$. Тогда существуют две заходящие в вершину v дуги, например (x, v) и (y, v). Таким образом, существуют два различных пути из вершины r к вершине v, один — через дугу (x, v), а другой — через дугу (y, v). Поэтому при удалении одной из этих дуг получившийся граф будет иметь корень r и, следовательно, останется квазисильно связным. Это противоречит утверждению 3, следовательно, $d^-(v)=1$, $v\neq r$.

Наконец, никакая дуга не может заходить в r, поскольку в противном случае после удаления этой дуги из G граф имел бы все-таки корень r, а значит, был квазисильно связным. Это снова противоречит утверждению 3. Поэтому $d^-(r)=0$.

$4\Rightarrow 5$. Сумма полустепеней захода в графе G равна $n-1$. Поэтому по теореме 5.1 граф G имеет $n-1$ дугу. Поскольку граф G связный, то по теореме 2.1 (п. 3) он является деревом и, следовательно, ациклическим.

$5\Rightarrow 6$. Поскольку сумма полустепеней захода в графе G равна $n-1$, он имеет $n-1$ дугу. Так как граф G не имеет циклов, то по теореме 2.2 он является деревом. Поэтому из вершины r к любой другой вершине в графе G существует единственный путь. Такой путь должен быть ориентированным, иначе хотя бы одна из вершин пути должна иметь полустепень захода, превышающую 1, что противоречит п. 5. Таким образом, вершина r — корень графа G. Поэтому по теореме 5.3 граф G — квазисильно связный.

$6 \Rightarrow 1$. Поскольку граф G — квазисильно связный, то по теореме 5.3 он имеет корень. Однако из того, что он связный и не имеет циклов, следует, что граф G — дерево.

Подграф ориентированного графа G называется *ориентированным остовом* графа G, если он является ориентированным деревом и содержит все вершины G. Например, подграф, образованный дугами e_1, e_2, e_3, e_4 и e_5, является ориентированным остовом графа, изображенного на рис. 5.6.

Мы знаем, что граф G имеет остов в том и только в том случае, когда он связный. Соответствующее утверждение для случая ориентированных графов дается в следующей теореме:

Теорема 5.5. Ориентированный граф G имеет ориентированный остов тогда и только тогда, когда граф G — квазисильно связный.

Доказательство.

Необходимость. Если граф G имеет ориентированный остов, то очевидно, что корень ориентированного дерева будет корнем и в графе G. Поэтому по теореме 5.3 граф G — квазисильно связный.

Достаточность. Пусть граф G — квазисильно связный, не являющийся ориентированным деревом. Тогда по теореме 5.4 существуют дуги, удаление которых из графа G не лишает его свойства квазисильной связности. Поэтому, если из графа G удалить все такие дуги, то получившийся граф будет ориентированным остовом графа G.

Применение ориентированных деревьев в топологическом исследовании электрических цепей рассматривается в гл. 13.

Двоичным деревом является ориентированное дерево, полустепень исхода каждой вершины которого не превышает 2. Двоичным деревом, например, является граф на рис. 5.9. Как оказалось, двоичные

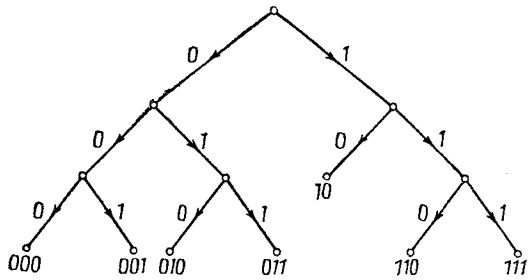


Рис. 5.9. Двоичное дерево.

деревья очень полезны при исследовании многих задач в областях, связанных с применением вычислительной техники, например при анализе алгоритмов, в методах поиска информации и т. д. Одной из часто встречающихся задач является следующая:

Даны n весов $w_1, w_2, \ldots, w_n$ и n длин $l_1, l_2, \ldots, l_n$; построить двоичное такое дерево с n листьями $v_1, v_2, \ldots, v_n$, что

1) $w_i\,(1 \leqslant i \leqslant n)$ — вес вершины v_i;

2) $l_i\,(1 \leqslant i \leqslant n)$ — длина пути из корня дерева в v_i;

3) сумма $\sum_{i=1}^{n} w_i l_i$, называемая *длиной взвешенных путей*,— минимальна.

Более общая задача возникает в описываемом ниже случае.

Множество таких слов, среди которых никакое не является началом другого, называется *префиксным кодом*. Очевидно, что если последовательность букв образована конкатенацией слов (сцеплением цепочек символов) префиксного кода, его можно разложить на отдельные слова префиксного кода путем разбора слева направо. Например, если префиксный код образует набор 000, 001, 01, 10, 11, а последовательность 1011001000 образована из его слов, то она легко раскладывается на слова 10, 11, 001 и 000.

Пусть $S=\{0, 1, 2, \ldots, m-1\}$ — алфавит из m букв, а T — такое ориентированное дерево, что

1) полустепень исхода каждой вершины не превышает m;
2) каждая исходящая из вершины дуга связана с буквой алфавита таким образом, что никакие две дуги не соответствуют одной букве. Тогда всякий лист v можно связать со словом, образованным конкатенацией, букв в том порядке, в котором они появляются на пути к листу v от корня. Легко убедиться в том, что слова, связанные таким образом с листьями, образуют префиксный код. Например, ориентированному дереву на рис. 5.9 соответствует следующий префиксный код: {000, 001, 010, 011, 10, 110, 111}.

Задача состоит в том, чтобы для данного m и длин $l_1, l_2, \ldots, l_n$ построить, используя буквы алфавита $S=\{0, 1, 2, \ldots, m-1\}$, префиксный код, длины слов которого составляют $l_1, l_2, \ldots, l_n$.

Этот и некоторые связанные с ним вопросы рассматриваются в разд. 15.2 и 15.3.

5.4. Ориентированные эйлеровы графы

Ориентированной эйлеровой цепью ориентированного графа G называется замкнутая ориентированная цепь, содержащая все дуги G. *Открытой ориентированной эйлеровой цепью* называется

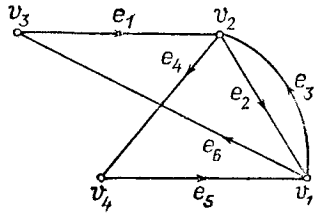


Рис. 5.10. Ориентированный эйлеров граф.

открытая ориентированная цепь, содержащая все дуги графа G. Ориентированный граф, обладающий ориентированной эйлеровой цепью, называется *ориентированным эйлеровым графом*.

Ориентированным эйлеровым графом является граф, изображенный на рис. 5.10, поскольку дуги e_1, e_2, e_3, e_4, e_5, e_6 образуют в графе G ориентированную эйлерову цепь.

В следующей теореме даются простые характеризации ориентированных эйлеровых графов. Теорема доказывается по той же схеме, что и теорема 3.1.

Теорема 5.6. Для связного ориентированного графа G следующие утверждения равносильны:

1) G — ориентированный эйлеров граф;
2) для любой вершины v графа G справедливо равенство $d^-(v)=d^+(v)$;
3) G — объединение нескольких реберно-непересекающихся контуров.

Рассмотрим, например, ориентированный эйлеров граф G на рис. 5.10. Легко проверить, что он обладает свойством, сформулированным в п. 2 теоремы, и является также объединением реберно-непересекающихся контуров $\{e_2, e_3\}$ и $\{e_1, e_4, e_5, e_6\}$.

Легко доказать и следующую теорему:

Теорема 5.7. Связный ориентированный граф содержит открытую ориентированную эйлерову цепь тогда и только тогда, когда выполняются условия:

1) в графе G имеются такие две вершины v_1 и v_2, что $d^+(v_1)=d^-(v_1)+1$ и $d^-(v_2)=d^+(v_2)+1$;
2) для любой вершины v, отличной от v_1 и v_2, справедливо равенство $d^-(v)=d^+(v)$.

Например, условиям этой теоремы удовлетворяет граф на рис. 5.11. Открытой ориентированной эйлеровой цепью графа G является последовательность e_1, e_2, e_3, e_4, e_5, e_6. Обсудим теперь одно интересное применение теоремы 5.6.

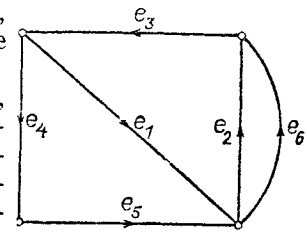


Рис. 5.11. Граф с открытой ориентированной эйлеровой цепью.

Пусть $S=\{0, 1, \ldots, s-1\}$ — алфавит, состоящий из s букв. Очевидно, что, используя буквы этого алфавита S, можно построить s^n различных слов длины n. *Последовательность де Брёйна* — это циклическая последовательность $a_0, a_1, a_2, \ldots, a_{L-1}$ длины $L=s^n$, у которой для любого слова ω длины n существует такое единственное i, что $\omega=a_ia_{i+1}\ldots a_{i+n-1}$, где значения индексов вычисляются по mod L. Последовательности де Брёйна находят применение в теории кодирования и при исследовании сетей связи. Подробную информацию об этих последовательностях можно найти в работах [5.1, 5.2]. Мы же рассмотрим следующую задачу:

Для любых ли $s \geqslant 2$ и целых n существует последовательность де Брёйна над алфавитом $S=\{0, 1, 2, \ldots, s-1\}$? Как мы увидим далее, ответ на этот вопрос положителен.

Покажем, что для любых $s \geqslant 2$ и n существует такой ориентированный эйлеров граф $G_{s,n}$, что всякая ориентированная эйлерова цепь в этом графе соответствует последовательности де Брёйна длины s^n над алфавитом $S=\{0, 1, 2, \ldots, s-1\}$. Граф $G_{s,n}$ строится следующим образом:

1) последовательность V образуется всеми s^{n-1} словами длины n—1 над алфавитом S;
2) E — множество всех s^n слов длины n над S;
3) дуга $b_1b_2 \ldots b_n$ имеет начальную вершину $b_1b_2 \ldots b_{n-1}$ и конечную вершину $b_2b_3 \ldots b_n$. Заметим, что каждая вершина имеет s заходящих и S исходящих дуг. Таким образом, для каждой вершины v справедливо равенство $d^-(v)=d^+(v)$.

Графы $G_{2,4}$ и $G_{2,3}$ представлены на рис. 5.12, *а*, *б*.

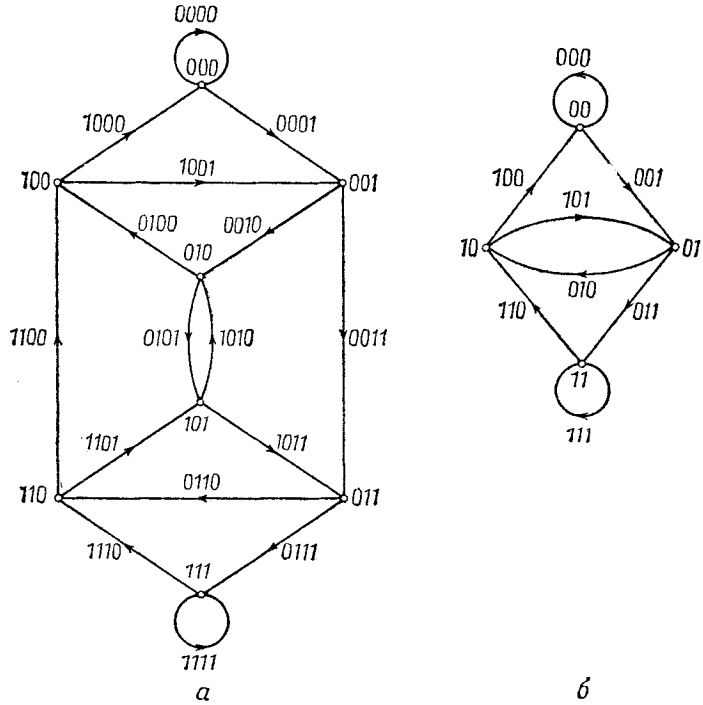


Рис. 5.12.
а—граф $G_{2,4}$, *б*—граф $G_{2,3}$.

Допустим, в графе $G_{s,n}$ имеется ориентированная эйлерова цепь C. Если мы конкатенируем первые буквы слов, представленных дугами $G_{s,n}$, в том порядке, в котором дуги появляются в цепи C, то получим такую последовательность длины s^n, что любая ее подпоследовательность из n последовательных букв будет соответствовать единственному слову, но никакие две различные подпоследовательности не будут соответствовать одному слову. Поэтому последовательность, построенная таким образом, будет последовательностью де Брёйна. Например, последовательность дуг 000, 001, 111, 110, 101, 010, 100 является ориентированной эйлеровой

цепью графа $G_{2,3}$. Конкатенация первых букв этих слов дает последовательность де Брёйна 00011101.

Таким образом, чтобы показать, что для любых $s \geqslant 2$ и n существует последовательность де Брёйна, необходимо доказать, что $G_{s,n}$ — ориентированный эйлеров граф. Это мы сделаем в следующей теореме:

Теорема 5.8. $G_{s,n}$ — ориентированный эйлеров граф.

Доказательство. $G_{s,n}$ — связный, так как для любых двух вершин $v_1 = b_1 b_2 \ldots b_{n-1}$ и $v_2 = c_1 c_2 \ldots c_{n-1}$ существует ориентированный путь из вершины v_1 в вершину v_2, состоящий из следующих дуг: $b_1 b_2 \ldots b_{n-1} c_1$, $b_2 b_3 \ldots b_{n-1} c_1 c_2$, $\ldots$, $b_{n-1} c_1 c_2 \ldots c_{n-1}$.

Поскольку, как мы видели раньше, полустепень захода каждой вершины графа $G_{s,n}$ равна ее полустепени исхода, то из теоремы 5.6 следует, что $G_{s,n}$ — ориентированный эйлеров граф.

Следствие 5.8.1. Для любых $s \geqslant 2$ и n существует последовательность де Брёйна.

5.5. Ориентированные остовы и ориентированные эйлеровы цепи

Пусть G — ориентированный эйлеров граф, не имеющий петель. В этом разделе мы свяжем число ориентированных эйлеровых цепей графа G с числом ориентированных остовов G.

Пусть $v_1, v_2, \ldots, v_n$ — вершины графа G. Рассмотрим ориентированную эйлерову цепь C графа G. Пусть e_{j_1} — дуга графа G, заходящая в вершину v_1. Для любого $p = 2, 3, \ldots, n$ будем считать, что e_{j_p} обозначает первую дугу C, входящую в вершину v_p после прохождения дуги e_{j_1}. Например, рассмотрим ориентированную эйлерову цепь $e_1, e_2, e_3, e_4, e_5, e_6$ графа, представленного на рис. 5.10. Если мы выберем $e_{j_1} = e_5$, то $e_{j_2} = e_1$, $e_{j_3} = e_6$ и $e_{j_4} = e_4$. Пусть H — подграф графа G на множестве дуг $\{e_{j_2}, e_{j_3}, \ldots, e_{j_n}\}$.

Лемма 5.1. Пусть C — ориентированная эйлерова цепь ориентированного эйлерова графа G. Подграф H, определяемый цепью C описанным способом, является ориентированным остовом с корнем v_1.

Доказательство. Очевидно, что в подграфе H $d^-(v_1) = 0$ и $d^-(v_p) = 1$ для всех $p = 2, 3, \ldots, n$. Допустим, что он содержит цикл C'. Тогда вершина v_1 не входит в цикл C', иначе $d^-(v_1) > 0$ или $d^-(v) > 1$ для некоторой другой вершины цикла C'. По этой же причине цикл C' является контуром. Поскольку $d^-(v) = 1$ для всякой вершины v в цикле C, никакая из дуг, не принадлежащих C, не входит ни в какую вершину C'. Это означает, что дуга e, являющаяся первой после прохождения e_{j_1} дугой цикла C, входящей в вершину цикла C', не принадлежит подграфу H, что противоречит его определению. Следовательно, подграф H не имеет циклов.

Теперь из п. 5 теоремы 5.4 следует, что подграф H — ориентированный остов графа G.

Пусть v_1 — корень ориентированного остова ориентированного эйлерова графа G без петель на n вершинах, а e_{j_1} — дуга, заходящая в вершину v_1 графа G. Пусть e_{j_p} ($p = 2, 3, \ldots, n$) — дуга, входя-

щая в вершину v_p подграфа H. Опишем метод построения ориентированной эйлеровой цепи G:

1) Начать с вершины v_1, проходя назад по любой дуге, входящей в вершину v_1, отличной от e_{j_1} (если такая дуга существует), или по e_{j_1}, если нет альтернативы;
2) По достижении вершины v_p оставить ее, проходя назад по дуге, которая еще не была пройдена и, если возможно, отличной от e_{j_p}. Остановиться, если не осталось непройденных дуг среди входящих в вершину v_p.

В этой процедуре каждый раз, когда достигается вершина $v_p=v_1$, будет существовать непройденная дуга, входящая в вершину v_p, поскольку полустепень захода всякой вершины графа G равна полустепени исхода. Следовательно, процедура закончится только в вершине v_1 после прохождения всех заходящих в нее дуг.

Предположим, что в графе G осталась непройденной дуга (u, v) после того, как процедура закончилась в вершине v_1. Поскольку полустепень захода в вершину u равна ее полустепени исхода, существует по крайней мере одна непройденная дуга, заходящая в вершину u. Если число таких непройденных дуг больше 1, то одна из них будет дугой y, входящей в вершину u подграфа H. Это следует из шага 2 процедуры. Непройденная дуга y приведет к другой непройденной дуге, также принадлежащей подграфу H. Наконец будет достигнута вершина v_1 и найдена непройденная дуга, заходящая в нее. Однако это невозможно, поскольку все дуги, заходящие в вершину v_1, были пройдены до того, как процедура завершилась в ней.

Таким образом, при выполнении описанной выше процедуры будут пройдены все дуги графа G и действительно построена ориентированная эйлерова цепь.

Поскольку в каждой вершине v_p возможны $(d^-(v_p)-1)!$ различных вариантов последовательности выбора заходящих дуг (e_{j_p} — последняя), то число различных ориентированных эйлеровых цепей, которые можно построить по данным ориентированному остову подграфа H и дуге e_{j_1}, определяется выражением $\prod_{p=1}^{n} (d^-(v_p)-1)!$.

Кроме того, различный выбор подграфа H приведет для некоторых $p=2, 3, \ldots, n$ к различным e_{j_p}, что в свою очередь даст в результате другое вхождение в вершину v_p после прохождения e_{j_1} в получившейся ориентированной эйлеровой цепи.

Наконец, поскольку процедура построения ориентированной эйлеровой цепи является обращением процедуры построения ориентированного остова, то всякую ориентированную эйлерову цепь можно построить по нескольким ориентированным остовам.

Таким образом, мы доказали следующую теорему, принадлежащую де Брёйну и ван Аардену — Эренфесту [5.3].

Теорема 5.9. Число ориентированных эйлеровых цепей в ориентированном эйлеровом графе G без петель равно $\tau_d(G) \prod_{p=1}^{n} (d^-(v_p)-1)!$, где $\tau_d(G)$ — число ориентированных остовов графа G с корнем v_1.

Поскольку число ориентированных эйлеровых цепей не зависит от выбора корня, приходим к следующему выводу.

Следствие 5.9.1. Число ориентированных остовов ориентированного эйлерова графа имеет постоянное значение, не зависящее от выбора корня.

Формула для нахождения числа $\tau_d(G)$ выводится в разд. 6.9.

5.6. Ориентированные гамильтоновы графы

Контур ориентированного графа G называется *ориентированным гамильтоновым контуром* G, если он содержит все его вершины.

Ориентированный путь графа G называется *гамильтоновым*, если в нем содержатся все вершины того же графа.

Ориентированным гамильтоновым графом называется граф, имеющий гамильтонов контур.

Гамильтоновым контуром, например, является последовательность дуг e_1, e_2, e_3, e_4, e_5, e_6 в графе, представленном на рис. 5.13, *а*.

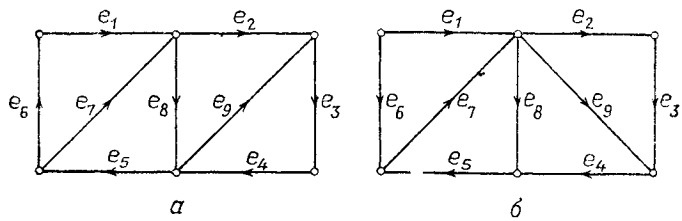


Рис. 5.13.

а — ориентированный гамильтонов граф, *б* — ориентированный граф с гамильтоновым ориентированным путем, но без гамильтоновых контуров

Последовательность дуг e_1, e_2, e_3, e_4, e_5 в графе на рис. 5.13, *б* образует ориентированный гамильтонов путь. Заметим, что в этом графе не содержится гамильтоновых контуров.

Характеризация ориентированных гамильтоновых графов в такой же степени трудна, как и в случае неориентированных гамильтоновых графов. Однако существует несколько достаточных условий, гарантирующих присутствие гамильтоновых контуров или ориентированных путей. Рассмотрим некоторые из этих условий.

Ориентированный граф называется полным, если полным является неориентированный граф, лежащий в его основе. Следующая теорема принадлежит Муну [5.4].

Теорема 5.10. Пусть u — произвольная вершина сильно связного полного ориентированного графа на $n \geqslant 3$ вершинах. Для любого k, $3 \leqslant k \leqslant n$, существует контур длины k, содержащий u.

Доказательство. Пусть $G=(V, E)$ — сильно связный полный ориентированный граф на $n\geqslant 3$ вершинах. Возьмем произвольную вершину u графа G. Пусть $S=\Gamma^+(u)$, а $T=\Gamma^-(u)$. Поскольку граф G — сильно связный, S и T не являются пустыми. По этой же причине в графе G имеется дуга, идущая из вершины $v\in S$ в вершину $w\in T$. Таким образом, вершина u лежит в контуре длины 3 (рис. 5.14).

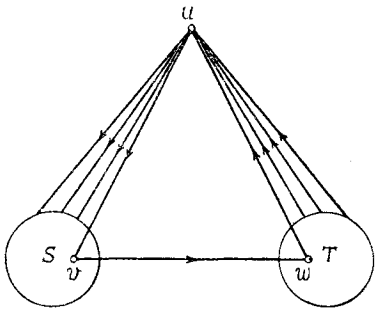


Рис. 5.14.

Докажем теорему индукцией по k. Допустим, что вершина u входит в контуры всех длин от 3 до p, где $p<n$. Покажем, что вершина u входит и в контур длины $p+1$.

Пусть C: $u=v_0, v_1, v_2, \ldots, v_p=u$ — контур длины p. Необходимо рассмотреть два случая.

Предположим, что некоторой вершине v, не входящей в контур C, инцидентны дуги (x, v) и (v, y), причем x и y входят в контур C. Тогда в нем найдутся две такие смежные вершины v_i и v_{i+1}, что в графе G имеются дуги (v_i, v) и (v, v_{i+1}). Следовательно, вершина u будет входить в контур $u=v_0, v_1, v_2, \ldots, v_i, v, v_{i+1}, \ldots, v_p=u$ длины $p+1$.

В противном случае обозначим через S множество всех вершин, не входящих в контур C и являющихся конечными в дугах, исходящих из вершин контура C, а через T — множество всех вершин, не входящих в контур C и являющихся начальными в дугах, заходящих в вершины того же контура. Снова, поскольку граф G сильно связный, S и T — непусты. Кроме того, существует дуга, идущая из вершины $v\in S$ в вершину $w\in T$. Таким образом, вершина u входит в контур $u=v_0, v, w, v_2, \ldots, v_p=u$ длины $p+1$ (рис. 5.15).

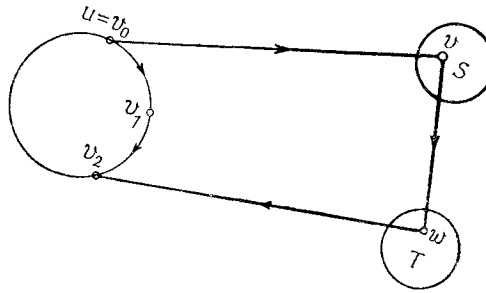


Рис. 5.15.

Следствие 5.10.1. Сильно связный полный ориентированный граф гамильтонов.

Приведем сейчас без доказательства очень сильную теорему, принадлежащую Гуйа-Ури. Доказательство ее исключительно сложно, его можно найти в работе [5.9].

Теорема 5.11. Пусть G — сильно связный граф на n вершинах без параллельных дуг и петель. Если для всякой вершины v справедливо неравенство $d^-(v)+d^+(v)\geqslant n$, то граф G имеет гамильтонов контур.

Следующее утверждение обобщает результат Дирака (следствие 3.4.1) для неориентированных графов, и его можно доказать с помощью теоремы 5.11 и упражнения 5.6.

Следствие 5.11.1. Пусть G — ориентированный граф на n вершинах без параллельных дуг и петель. Если $\min(\delta^-, \delta^+) \geqslant \frac{1}{2} n > 1$, то граф G содержит гамильтонов контур.

Непосредственное доказательство этого результата можно найти в работе [5.6].

Завершаем этот раздел утверждением о существовании ориентированного гамильтонова пути.

Теорема 5.12. Если ориентированный граф $G = (V, E)$ — полный, то он имеет ориентированный гамильтонов путь.

Доказательство. Рассмотрим ориентированный путь в графе G максимальной длины P: $a_1, a_2, \ldots, a_p$. Пусть b — вершина, не входящая в путь P. Тогда дуга $(b, a_1) \notin E$, поскольку иначе путь $b, a_1, a_2, \ldots, a_p$ был бы длиннее пути P. Поэтому $(a_1, b) \in E$. Теперь $(b, a_2) \notin E$, поскольку иначе путь $a_1, b, a_2, a_3, \ldots, a_p$ был бы длиннее пути P. Следовательно, $(a_2, b) \in E$.

Повторяя эти рассуждения, получим, что $(a_p, b) \in E$. Но это противоречит предположению, поскольку путь $a_1, a_2, \ldots, a_p, b$ длиннее пути P. Таким образом, нет вершины, не входящей в путь P, следовательно, P — ориентированный гамильтонов путь графа G.

5.7. Ациклические ориентированные графы

В этом разделе изучаются свойства важного класса ориентированных графов, а именно ациклических. Как мы знаем, ориентированный граф — ациклический, если он не содержит контуров. Очевидно, простейшим примером ациклического ориентированного графа является ориентированное дерево.

Основной результат, который мы получим в этом разделе, заключается в том, что вершины ациклического ориентированного графа G на n вершинах можно пометить таким образом целыми числами из множества $\{1, 2, \ldots, n\}$, что если в графе G имеется дуга (i, j), то $i < j$. Напомним, что дуга (i, j) направлена из вершины i в вершину j. Определенное нами упорядочение вершин называется *топологической сортировкой*. Топологически отсортированы, например, вершины ациклического ориентированного графа на рис. 5.16. Справедливость основного результата этого раздела определяется следующей теоремой:

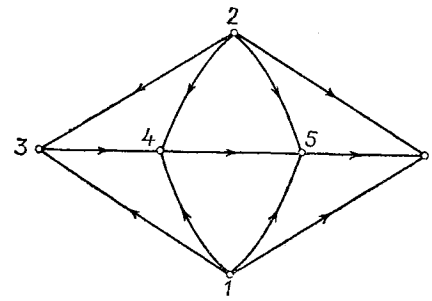


Рис. 5.16. Ациклический ориентированный граф.

Теорема 5.13. В ациклическом ориентированном графе имеются по крайней мере одна вершина с нулевой полустепенью захода и одна вершина с нулевой полустепенью исхода.

Доказательство. Пусть P: $v_1, v_2, \ldots, v_p$ — максимальный ориентированный путь графа G. Покажем, что полустепень захода v_1 и полустепень исхода v_p равны нулю.

Если полустепень захода v_1 не равна нулю, то существует такая вершина w, что в графе G имеется дуга (w, v_1). Возможны два случая.

Случай 1. Пусть $w \neq v_i$, $1 \leqslant i \leqslant p$. Тогда существует ориентированный путь P: $w, v_1, v_2, \ldots, v_p$, содержащий все дуги пути P. Это противоречит максимальности упомянутого пути.

Случай 2. Пусть для некоторого i $w = v_i$. Тогда в графе G имеется контур C: $v_1, v_2, \ldots, v_i, v_1$. Это противоречит условиям теоремы.

Таким образом, не существует такой вершины w, что (w, v_1) — дуга графа G. Другими словами, полустепень захода v_1 равна нулю.

Рассуждая аналогичным образом, можно показать, что равна нулю полустепень исхода v_p.

Для осуществления топологической сортировки вершин ациклического ориентированного графа G на n вершинах проделаем следующее.

Выберем произвольную вершину с нулевой полустепенью исхода. Это возможно, поскольку граф G — ациклический, а по теореме 5.13 в нем должна иметься хотя бы одна такая вершина. Пометим выбранную вершину целым n. Теперь удалим из графа G эту вершину и инцидентные ему дуги. Обозначим получившийся граф через G'. Поскольку граф G' также ациклический, можно выбрать в нем вершину с нулевой полустепенью исхода. Пометим эту вершину целым числом $n-1$. Повторим описанную процедуру до тех пор, пока не пометим все вершины. Легко видеть, что эта процедура порождает топологическую сортировку вершин графа G.

В соответствии с этой процедурой была сделана, например, разметка вершин графа на рис. 5.16.

5.8. Турниры

Полный ориентированный граф называется *турниром*. Этот термин получил свое название от соревнований по круговой системе, графическое представление которых имеет структуру полного ориентированного графа. В турнирах по круговой системе играют несколько команд, каждая со всеми остальными по одному разу. Игра по правилам не может закончиться вничью. В представлении турнира по круговой системе ориентированным графом командам соответствуют вершины и дуга (v_1, v_2) присутствует в графе, если соответствующая v_1 команда победила команду, представленную вершиной v_2. Очевидно, что в таком ориентированном графе нет параллельных дуг и петель и между каждыми двумя вершинами имеется точно одна дуга. Таким образом, граф является полным, а следовательно, и турниром. Пример турнира приведен на рис. 5.17. Участвующие в турнире команды можно ранжировать в соответствии с количеством очков. *Количество очков* команды соответствует

числу побежденных ею противников. Введем определение последовательности очков турнира.

Последовательностью очков турнира на n вершинах называется последовательность $(s_1, s_2, \ldots, s_n)$, в которой каждое s_i является полустепенью исхода вершины турнира. Вследующей теореме дается интересная характеризация турниров, исходя из последовательности очков.

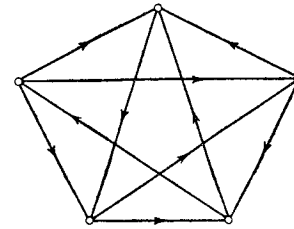


Рис. 5.17. Турнир.

Теорема 5.14. Последовательность неотрицательных целых чисел $s_1, s_2, \ldots, s_n$ является последовательностью очков турнира G тогда и

1) $\sum_{i=1}^{n} s_i = \frac{n(n-1)}{2}$, 2) $\sum_{i=1}^{k} s_i \quad \frac{k(k-1)}{2}$

только тогда, когда для всех $k < n$.

Доказательство.

Необходимость. Заметим, что сумма $\sum_{i=1}^{n} s_i$ равна числу дуг турнира G (теорема 5.1). Поскольку турнир — это полный граф, он имеет $n(n-1)/2$ дуг, где n — число вершин. Таким образом, $\sum_{i=1}^{k} s_i = \frac{n(n-1)}{2}$. Для доказательства условия 2 рассмотрим подтурнир из любых k вершин $v_1, v_2, \ldots, v_k$. Он содержит $k(k-1)/2$ дуг. Поэтому в целом турнире $\sum_{i=1}^{k} s_i \geqslant k(k-1)/2$, поскольку в нем могут присутствовать дуги, исходящие из вершин подтурнира и направленные в вершины, не принадлежащие подтурниру.

Достаточность доказывается в работе [5.5].

Предположим, что команды турнира по круговой системе можно упорядочить таким образом, что за каждой командой идет побежденная ею. Тогда для фиксации порядка можно будет присвоить командам целые числа 1, 2, . . ., n. Такое ранжирование всегда возможно, поскольку в турнире имеется ориентированный гамильтонов путь (теорема 5.12). Оно называется *ранжированием* гамильтоновым путем.

Заметим, что ранжирование гамильтоновым путем может отличаться от ранжирования по очкам. Более того, турнир может иметь не один ориентированный гамильтонов путь. В таком случае возможно более одного ранжирования гамильтоновым путем. Однако в транзитивном турнире существует точно один ориентированный гамильтонов путь. Это устанавливается в следующей легко доказываемой теореме:

Теорема 5.15. В транзитивном турнире имеется точно один ориентированный гамильтонов путь.

Другие процедуры ранжирования рассматриваются в работах [5.6—5.8].

5.9. Замечания, касающиеся литературы

Много результатов по ориентированным графам содержится в прекрасных монографиях [5.5, 5.9]. Монография [5.10] посвящена исключительно турнирам. Эта же тема изучается в работе [5.11].

Ориентированные деревья применяются при представлении в ЭВМ комбинаторных объектов. Широкое обсуждение этой темы проводится в монографиях [5.12—5.15].

Ориентированные деревья применяются также в других областях, например при топологическом исследовании электрических цепей. В этой области, в частности, представляет интерес определение числа ориентированных деревьев в ориентированном графе. Этот вопрос обсуждается в гл. 6, а применения ориентированных деревьев в теории цепей рассматриваются в гл. 13.

Случаи использования ориентированных деревьев обсуждаются в работе [5.16].

Упражнения

5.1. Пусть G — ориентированный граф без петель и параллельных дуг. Пусть $\max(\delta^-, \delta^+)=k$. Докажите, что а) граф G имеет ориентированный путь длины не менее k и б) если $k>0$, то граф G имеет контур длины не менее $k+1$.

5.2. Покажите, что дуги неориентированного графа $G=(V, E)$ можно ориентировать так, чтобы в получившемся ориентированном графе $|d^+(v)-d^-(v)|\leqslant 1$ для любой вершины $v\in V$.

5.3. Ориентированный разрез ориентированного графа — это разрез $\langle S, \bar{S}\rangle$, все дуги которого или исходят из S, или заходят в S. Покажите, что дуга ориентированного графа принадлежит либо ориентированному разрезу, либо контуру, но не принадлежит им одновременно. (Это частный случай результата, известного под названием «лемма о раскраске дуг», который доказывается в гл. 10, теорема 10.31.)

5.4. Покажите, что для ориентированного графа G, имеющего не менее одной дуги, следующие утверждения равносильны: а) граф G не имеет контуров и б) любая дуга графа G входит в ориентированный разрез.

5.5. Покажите, что для ориентированного графа G, имеющего по крайней мере одну дугу, следующие утверждения равносильны: а) граф G — сильно связный, б) любая дуга графа G лежит в контуре и в) граф G не имеет ориентированных разрезов.

5.6. Покажите, что ориентированный граф на n вершинах без параллельных дуг и петель сильно связен, если $\min\{\delta^-, \delta^+\}\geqslant (n-1)/2$.

5.7. Покажите, что сильно связный граф, содержащий цикл нечетной длины, содержит также контур нечетной длины.

5.8. Пусть $G=(V, E)$ — такой ориентированный граф, что а) $d^+(x)-d^-(x)=l=d^-(y)-d^+(y)$; б) $d^+(v)=d^-(v)$ для $v\in V-\{x, y\}$. Покажите, что в графе G имеется l реберно-непересекающихся ориентированных x—y-путей.

5.9. Покажите, что сильно связный граф имеет остовный ориентированный маршрут.

5.10. Что из себя представляет длиннейшая циклическая последовательность, образованная из трех символов x, y и z так, что никакая подпоследовательность из четырех символов не повторяется? Приведите пример такой последовательности.

5.11. Найдите такую циклическую последовательность из семи «0» и семи «1», что все 4-разрядные двоичные последовательности, кроме 0000 и 1111, являются ее подпоследовательностями.

5.12. Докажите, что гамильтонов контур $G_{s,n}$ соответствует ориентированной эйлеровой цепи $G_{s,n-1}$. Всегда ли $G_{s,n}$ имеет гамильтонов контур?

5.13. Покажите, что число ориентированных эйлеровых цепей в ориентированном графе, имеющем n вершин и $m > 2n$ дуг, четно.

5.14. Покажите, что ориентированный граф $G=(V, E)$ без параллельных дуг и петель имеет ориентированный гамильтонов путь, если $d^+(v)+d^-(v) \geqslant n-1$ для любой вершины $v \in V$.

5.15. Покажите, что любой турнир сильно связен или его можно преобразовать в сильно связный турнир сменой ориентации только одной дуги.

5.16. Подмножество $S \subseteq V$ называтся независимым множеством графа $G=(V, E)$, если никакие две вершины S не смежны. Докажите, что ориентированный граф G без петель имеет такое независимое подмножество S, что всякую вершину v графа G, не входящую в подмножество S, можно достичь из вершины в подмножестве S ориентированным путем длины не более 2 [5.17]. Этот результат имеет интересное следствие: турнир обладает вершиной, из которой все остальные достигаются ориентированным путем длиной не более 2.

5.17. Покажите, что очки s_i турнира на n-вершинах удовлетворяют условию

$$\sum_{i=1}^{n} s_i^2 = \sum_{i=1}^{n} (n-1-s_i)^2,$$

6. Матрицы графов

В этой главе вводятся матрицы инциденций, циклов, разрезов и смежности графа и устанавливаются некоторые свойства этих матриц, которые помогут раскрыть структуру графа. Матрицы инциденций, циклов и разрезов используются при исследовании электрических цепей; эти матрицы входят в качестве коэффициентов в уравнения Кирхгофа, описывающие цепь. Поэтому свойства этих матриц и другие связанные с ними результаты, формулируемые в настоящей главе, будут широко использоваться в ч. II книги. Изучаемые нами свойства матрицы смежности служат основой подхода сигнальных графов потоков, являющихся мощным инструментом при исследовании линейных систем. Теория сигнальных графов потоков развивается в разд. 6.11.

Мы рассматриваем главным образом матрицы, связанные с ориентированными графами. Однако наши рассуждения останутся справедливыми и для неориентированных графов, если считать, что операции сложения и умножения действуют в поле GF (2) целых чисел по mod 2.

6.1. Матрица инциденций

Рассмотрим граф G без петель на n вершинах и m ребрах. Матрица инциденций $A_c=[a_{ij}]$ графа G имеет n строк (по одной на каждую вершину) и m столбцов (по одному на каждую дугу). Элемент a_{ij} матрицы A_c определяется следующим образом:

Если граф G *ориентированный*, $a_{ij}=$
- 1, если i-я дуга инцидентна j-й вершине и исходит из нее;
- -1, если i-я дуга инцидентна j-й вершине и заходит в нее;
- 0, если i-я дуга не инцидентна j-й вершине.

Если граф G *неориентированный*, $a_{ij}=$
- 1, если j-е ребро инцидентно i-й вершине;
- 0 в противном случае.

Строки матрицы A_c называют *векторами инциденций* графа G. На рис. 6.1, *а* и *б* представлены два графа со своими матрицами инциденций.

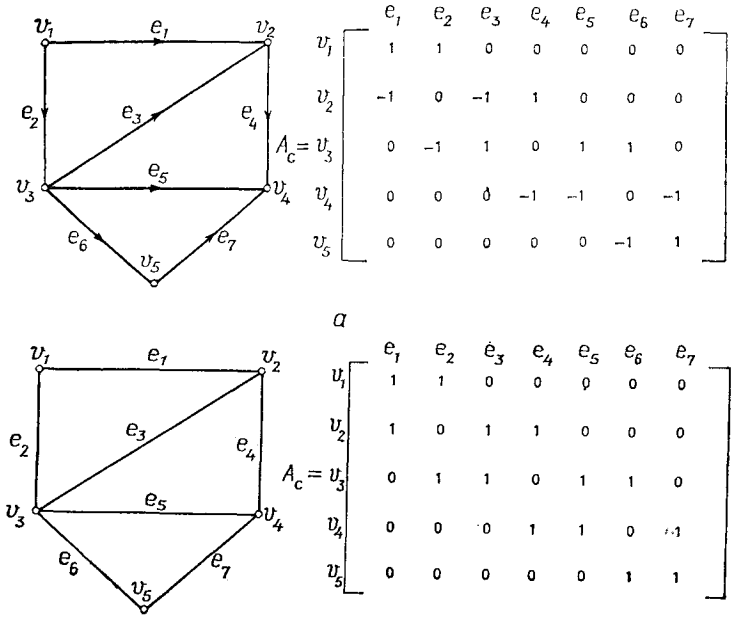


Рис. 6.1.

а — ориентированный граф G и его матрица инциденций; *б* — неориентированный граф G и его матрица инциденций.

Из определения очевидно, что всякий столбец матрицы A_c содержит точно два ненулевых элемента: $+1$ и -1. Поэтому любую строку этой матрицы можно определить по остальным $n-1$ строкам. Таким образом, произвольные $n-1$ строк матрицы A_c содержат всю информацию об этой матрице. Другими словами, строки матрицы A_c линейно-зависимы.

Подматрицу A матрицы A_c на $n-1$ строке называют *усеченной матрицей инциденций*. Если вершина соответствует строке матрицы A_c, отсутствующей в подматрице A, то говорят, что A — *матрица инциденций, усеченная по строке,— соответствует данной вершине.* Заметим, что

$$\operatorname{rank}(A) = \operatorname{rank}(A_c) \leqslant n - 1. \tag{6.1}$$

Покажем, что в случае связных графов ранг матрицы A_c равен $n-1$. Это утверждение основано на следующей теореме:

Теорема 6.1. Определитель любой усеченной матрицы инциденций дерева равен ± 1.

Доказательство: Доказательство проводим индукцией по числу n вершин дерева.

Любая усеченная матрица инциденций дерева на двух вершинах является матрицей размерности 1×1, единственный элемент которой равен ± 1. Следовательно, теорема верна для $n=2$. Заметим, что для $n=1$ она уже не выполняется.

Пусть теорема выполняется для $2\leqslant n\leqslant k$. Рассмотрим дерево T на $k+1$ вершине. Пусть A — усеченная матрица инциденций дерева T. По теореме 2.5 оно имеет по крайней мере 2 висячие вершины. Пусть i-я вершина дерева T является висячей, а матрица A усечена не по строке, соответствующей этой вершине. Если этой вершине инцидентна l-я дуга, то

$$a_{il}=\pm 1 \text{ и } a_{ij}=0 \ (j\neq l).$$

Разлагая определитель A по i-й строке, получаем

$$\det(A)=\pm(-1)^{i+l}\det(A'), \tag{6.2}$$

где A' получается из A удалением i-й строки и l-го столбца.

Пусть T' — граф, получающийся после удаления из дерева T i-й вершины и l-й дуги. Очевидно, что T' — дерево, поскольку i-я вершина и l-я дуга — висячие в дереве T. Более того, нетрудно убедиться в том, что A' — усеченная матрица инциденций дерева T'. Поскольку T' — дерево на $n-1$ вершине, из индуктивного предположения следует, что

$$\det A'=\pm 1. \tag{6.3}$$

В совокупности с выражением (6.2) это доказывает теорему для $n=k+1$.

Поскольку связный граф имеет хотя бы один остов, из доказанной теоремы следует, что в любой усеченной матрице инциденций A связного графа есть невырожденная подматрица порядка $n-1$. Таким образом, для связного графа

$$\operatorname{rank}(A)=n-1. \tag{6.4}$$

Так как $\operatorname{rank}(A_c)=\operatorname{rank}(A)$, получаем следующую теорему:

Теорема 6.2. Ранг матрицы инциденций связного графа G на n вершинах равен $n-1$, т. е. рангу графа.

Непосредственным следствием данной теоремы является следующее утверждение:

Следствие 6.2.1. Если граф на n вершинах имеет p компонент, то ранг его матрицы инциденций равен $n-p$, т. е. рангу графа G.

6.2. Матрица разрезов

Для определения матрицы разрезов ориентированного графа необходимо каждому разрезу графа присвоить ориентацию.

Рассмотрим ориентированный граф $G=(V, E)$. Если V_a — непустое подмножество множества V, то напомним (глава 2), что множество дуг, соединяющих вершины V_a и $\bar{V}_a$, является разрезом, обозначаемым $\langle V_a, \bar{V}_a\rangle$. Ориентацию $\langle V_a, \bar{V}_a\rangle$ можно принять как от вершины V_a к вершине $\bar{V}_a$, так и наоборот. Допустим, мы принимаем ориентацию от вершины V_a к вершине $\bar{V}_a$. Тогда говорят, что ориентация дуги из $\langle V_a, \bar{V}_a\rangle$ соответствует ориентации разреза $\langle V_a, \bar{V}_a\rangle$, если дуга ориентирована от вершины из V_a в вершину из $\bar{V}_a$.

Матрица разрезов $Q_c=[q_{ij}]$ графа G с m ребрами имеет m столбцов и столько строк, сколько в этом графе имеется разрезов. Элемент q_{ij} определяется следующим образом:

Если граф G *ориентированный*, $q_{ij}=$
- 1, если j-я дуга входит в i-й разрез и ее ориентация соответствует ориентации разреза;
- -1, если j-я дуга входит в i-й разрез и ее ориентация не соответствует ориентации разреза;
- 0, если j-я дуга не входит в i-й разрез.

Если граф G *неориентированный*, $q_{ij}=$
- 1, если j-е ребро входит в i-й разрез;
- 0 в противном случае.

Строки матрицы Q_c называют *векторами разрезов*.

На рис. 6.2 представлены три разреза ориентированного графа из рис. 6.1, *а*. Штриховыми линиями в каждом случае показана

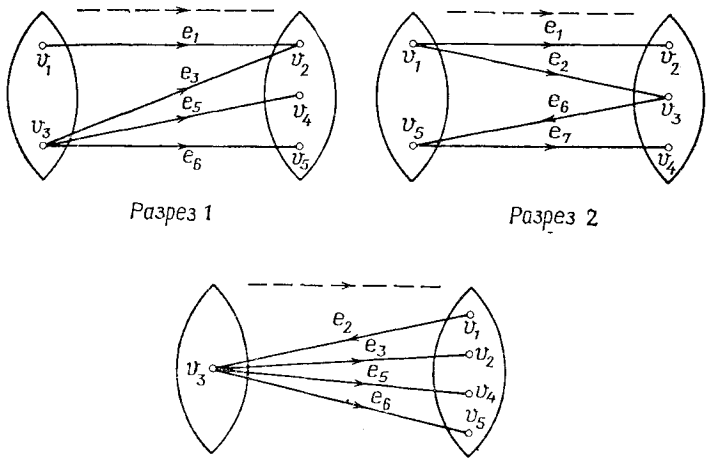


Рис. 6.2. Некоторые разрезы графа на рис. 6.1, *а*.

ориентация разреза. Соответствующая этим трем разрезам подматрица Q_c имеет вид

	e_1	e_2	e_3	e_4	e_5	e_6	e_7
Разрез 1	1	0	1	0	1	1	0
Разрез 2	1	1	0	0	0	-1	1
Разрез 3	0	-1	1	0	1	1	0

Соответствующую подматрицу в случае неориентированного графа можно получить, заменив в этой матрице «—1» на «+1».

Рассмотрим произвольную вершину v. Ненулевые элементы соответствующего вектора инциденций представляют инцидентные вершине v дуги. Эти дуги образуют разрез $\langle v, V-v\rangle$. Если принять ориентацию этого разреза от v к $V-v$, то из определений матриц разрезов и инциденций следует, что строка матрицы Q_c, соответствующая разрезу $\langle v, V-v\rangle$, совпадает со строкой матрицы A_c, соответствующей вершине v. Следовательно, A_c — подматрица матрицы Q_c.

Покажем, что ранги матриц Q_c и A_c одинаковы. Для этого докажем следующую теорему:

Теорема 6.3. Всякую строку матрицы разрезов Q_c можно выразить двумя способами в виде линейной комбинации строк матрицы A_c. В обоих случаях ненулевые коэффициенты в линейной комбинации равны +1 или —1.

Доказательство. Пусть $\langle V_a, \overline{V}_a\rangle$ — i-й разрез графа G на n вершинах и m дугах, а q_i — соответствующий вектор разреза. Пусть $V_a=\{v_1, v_2, \ldots, v_r\}$, а $\overline{V}_a=\{v_{r+1}, v_{r+2}, \ldots, v_n\}$. Пусть, далее, $a_i\,(1\leqslant i\leqslant n)$ — вектор инциденций, соответствующий вершине v_i.

Не нарушая общности, допустим, что ориентация $\langle V_a, \overline{V}_a\rangle$ от вершины V_a к вершине $\overline{V}_a$. Для доказательства теоремы установим следующее равенство:

$$q_i=a_1+a_2+\ldots+a_r=-(a_{r+1}+a_{r+2}+\ldots+a_n). \tag{6.5}$$

Пусть v_p и v_q — вершины, инцидентные k-й дуге, при этом она ориентирована от v_p к v_q. Поэтому

$$a_{pk}=1,\ a_{qk}=-1,\ a_{jk}=0,\ j\neq p,\ k. \tag{6.6}$$

Сейчас возможны четыре случая:

Случай 1. $v_p\in V_a$, а $v_q\in\overline{V}_a$, т. е. $p\leqslant r$, а $q\geqslant r+1$, поэтому $q_{ik}=1$.

Случай 2. $v_p\in\overline{V}_a$, а $v_q\in V_a$, т. е. $p\geqslant r+1$, а $q\leqslant r$, поэтому $q_{ik}=-1$.

Случай 3. $v_p, v_q\in V_a$, т. е. $p, q\leqslant r$, поэтому $q_{ik}=0$.

Случай 4. $v_p, v_q\in\overline{V}_a$, т. е. $p, q\geqslant r+1$, поэтому $q_{ik}=0$.

Используя выражение (6.6), легко показать, что в каждом из этих четырех случаев выполняется следующее равенство:

$$q_{ik}=(a_{1k}+a_{2k}+\ldots+a_{rk})=-(a_{r+1,\,k}+a_{r+2,\,k}+\ldots+a_{nk}). \tag{6.7}$$

Поскольку это равенство выполняется для всех $1\leqslant k\leqslant m$, справедливо и выражение (6.5), что доказывает теорему.

Для иллюстрации теоремы рассмотрим разрез 1 на рис. 6.2. Он разделяет вершины в $V_a=\{v_1, v_3\}$ и вершины в $\overline{V}_a$. Ориентация разреза от V_a к $\overline{V}_a$. Поэтому соответствующий вектор разреза можно выразить следующим образом: $[1\ 0\ 1\ 0\ 1\ 1\ 0] = a_1+a_3=-a_2-a_4-a_5$, где $a_1, a_2, \ldots, a_5$ — строки матрицы A (рис. 6.1, *а*).

Важным следствием из теоремы 6.3 является то, что $\operatorname{rank}(Q_c)\leqslant\operatorname{rank}(A_c)$. Однако, поскольку A_c — подматрица матрицы Q_c, то $\operatorname{rank}(Q_c)\geqslant\operatorname{rank}(A_c)$. Поэтому мы получаем, что $\operatorname{rank}(Q_c)=\operatorname{rank}(A_c)$.

Теперь из теоремы 6.2 и следствия 6.2.1 вытекает следующая теорема:

Теорема 6.4. Ранг матрицы разрезов Q_c связного графа G на n вершинах равен $n-1$, т. е. рангу G.

Следствие 6.4.1. Ранг матрицы разрезов Q_c графа G на n вершинах, имеющего p компонент, равен $n-p$, т. е. рангу этого графа.

Из этих выводов следует, что матрица инциденций A_c является важной подматрицей матрицы разрезов Q_c. Определим сейчас другую важную подматрицу матрицы Q_c.

Нам известно, что остов T связного графа G на n вершинах определяет множество из $n-1$ базисного разрезающего множества — по одному базисному разрезающему на ветвь дерева T. Подматрица Q_c, соответствующая этому $n-1$-базисному разрезающему множеству, называется *базисной матрицей разрезающих множеств* Q_f графа G по отношению к дереву T. Пусть $b_1, b_2, \ldots, b_{n-1}$ — ветви дерева T. Переставим столбцы и строки так, что

1) i-й столбец соответствует ветви b_i для $1 \leqslant i \leqslant n-1$;
2) i-я строка соответствует базисному разрезающему множеству, определяемому b_i. Если теперь ориентацию базисного разрезающего множества выбрать таким образом, чтобы ей соответствовала ориентация определяющей ветви, тогда матрицу Q_f можно представить в следующей удобной форме:

$$Q_f = [U \mid Q_{fc}], \tag{6.8}$$

где U — единичная матрица порядка $n-1$, а столбцы ее соответствуют ветвям дерева T.

Например, базисная матрица разрезающих множеств Q_f связного графа на рис. 6.1, *а* по отношению к остову $T=\{e_1, e_2, e_6, e_7\}$ имеет вид

$$Q_f = \begin{array}{c} \\ e_1 \\ e_2 \\ e_6 \\ e_7 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_6 & e_7 & e_3 & e_4 & e_5 \end{array} \\ \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right] \end{array}. \tag{6.9}$$

Из (6.8) следует, что ранг матрицы Q_f равен $n-1$, т. е. рангу матрицы Q_c. Таким образом, всякий вектор разреза можно выразить в виде линейной комбинации базисных векторов разрезающих множеств.

6.3. Цикломатическая матрица

Цикл можно обойти в одном из двух направлений: по часовой или против часовой стрелки. Направление, выбираемое для обхода цикла, определяет его ориентацию. Ориентацию цикла можно указать стрелкой, как на рис. 6.3.

Рассмотрим дугу e с концевыми вершинами v_i и v_j. Пусть дуга e ориентирована от v_i к v_j и входит в цикл C. Будем говорить, что ориентация дуги e соответствует ориентации цикла, если вершина v_i встречается при обходе цикла C в направлении, указанном его ориентацией, раньше вершины v_j. Например, ориентация дуги e_1 в цикле на рис. 6.3 соответствует ориентации цикла, а ориентация дуги e_4 — нет.

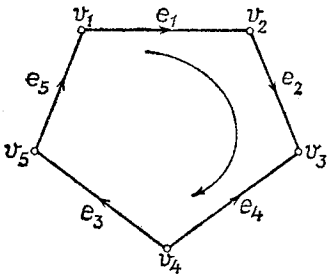


Рис. 6.3.

Цикломатическая матрица [1] $B_c=[b_{ij}]$ графа G с m ребрами имеет m столбцов и столько строк, сколько циклов имеется в графе G. Элемент b_{ij} определяется следующим образом:

Если граф G *ориентированный*, $b_{ij}=$
- 1, если j-я дуга входит в i-й цикл и ее ориентация соответствует ориентации цикла;
- -1, если j-я дуга входит в i-й цикл и ее ориентация не соответствует ориентации цикла;
- 0, если j-я дуга не входит в i-й цикл.

Если граф G *неориентированный*, $b_{ij}=$
- 1, если j-е ребро входит в i-й цикл;
- 0 в противном случае.

Строки матрицы B_c называются *циклическими векторами* графа G.

В качестве примера снова рассмотрим граф на рис. 6.1, *а*. На рис. 6.4 представлены три цикла этого графа и их ориентация. Подматрица B_c, соответствующая этим циклам, имеет вид

	e_1	e_2	e_3	e_4	e_5	e_6	e_7
Цикл 1	1	−1	0	1	−1	0	0
Цикл 2	−1	1	1	0	0	0	0
Цикл 3	1	−1	0	1	0	−1	−1

Соответствующая подматрица для неориентированного графа рис. 6.1, *б* получается заменой в этой матрице «—1» на «+1».

[1] Другое название — матрица циклов, для удобства мы применяем оба термина.— *Прим. ред.*

Сейчас определим подматрицу B_c, играющую важную роль.

Рассмотрим произвольный остов T связного графа G, имеющего n вершин и m ребер. Пусть c_1, c_2, . . . , c_{m-n+1} — хорды T. Как известно, эти $m-n+1$-хорды определяют множество из $m-n+1$-базисных циклов. Подматрица B_c, соответствующая этим базисным

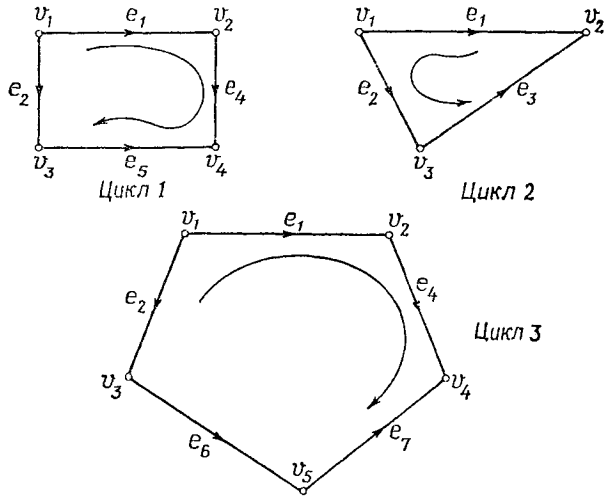


Рис. 6.4. Некоторые циклы графа на рис. 6.1, *а*.

циклам, называется *базисной цикломатической матрицей* B_f графа G по отношению к остову T.

Переставим столбцы и строки B_f так, что

1) i-й столбец соответствует хорде c_i, $1 \leqslant i \leqslant m-n+1$;
2) i-я строка соответствует базисному циклу, определяемому хордой c_i.

Выбирая ориентацию базисного цикла таким образом, чтобы ей соответствовала ориентация определяющей хорды, получим следующее представление матрицы B_f:

$$B_f = [U \mid B_{ft}], \tag{6.10}$$

где U — единичная матрица размерности $m-n+1$, столбцы которой соответствуют хордам остова T.

Например, базисная цикломатическая матрица графа на рис. 6.1, *а* по отношению к остову $T=\{e_1, e_2, e_8, e_7\}$ имеет вид

$$B_f = \begin{array}{c} \\ e_3 \\ e_4 \\ e_5 \end{array} \begin{array}{c} \begin{array}{ccc|cccc} e_3 & e_4 & e_5 & e_1 & e_2 & e_6 & e_7 \end{array} \\ \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 \end{array}\right] \end{array} \tag{6.11}$$

Из (6.10) следует, что ранг матрицы B_f равен $m-n+1$. Поскольку матрица B_f — подматрица B_c, получаем

$$\text{rank}\,(B_c) \geqslant m-n+1. \tag{6.12}$$

В следующем разделе мы покажем, что ранг подматрицы B_c в случае связного графа равен $m-n+1$.

Сейчас отметим, что используемый в предыдущем разделе подход к определению ранга матрицы Q_c не годится для случая подматрицы B_c. (Почему?) Означает ли это, что отмеченная нами «двойственность» (гл. 4) между циклами и разрезающими множествами просто случайна? Нет, мы увидим, что аргументы, изложенные в следующем разделе, еще более подтвердят эту «двойственность».

6.4. Соотношение ортогональности

В разд. 4.6 мы показали, что в случае неориентированного графа всякий циклический вектор ортогонален всякому вектору разреза. Теперь мы докажем, что этот результат справедлив также и для случая ориентированных графов. Наше доказательство основано на следующей теореме:

Теорема 6.5. Если разрез и цикл ориентированного графа имеют $2k$ общих дуг, то k из этих дуг имеют одинаковую относительную ориентацию в разрезе и цикле, а оставшиеся k дуг имеют противоположные ориентации в разрезе и цикле.

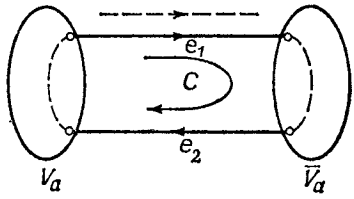


Рис. 6.5.

Доказательство: Рассмотрим разрез $\langle V_a, \overline{V}_a\rangle$ и цикл C ориентированного графа. Предположим, что мы обходим C, начиная с вершины, находящейся в V_a. Тогда для всякой дуги e_1, которая приводит нас из V_a в $\overline{V}_a$, существует дуга e_2, которая приводит нас из V_a в $\overline{V}_a$. Заметим, что если e_1 (e_2) имеет одинаковую относительную ориентацию в разрезе и в цикле, то e_2 (e_1) имеет в разрезе и цикле противоположные ориентации (рис. 6.5).

Докажем сейчас основной результат этого раздела.

Теорема 6.6. (Соотношение ортогональности.) Если столбцы цикломатической матрицы B_c и матрицы разрезов Q_c расположить в одном порядке, то $B_cQ_c^t=0$.

Доказательство. Рассмотрим цикл и разрез, имеющие $2k$ общих дуг. Произведение соответствующих циклического вектора и вектора разреза равно нулю, поскольку по теореме 6.5 оно равно сумме k «1» и k «—1». Поскольку каждый элемент матрицы $B_cQ_c^t$ равен произведению циклического вектора на вектор разреза, то теорема доказана.

Соотношение ортогональности — это очень глубокий результат, имеющий интересные применения в теориях графов и (как будет по-

казано в ч. II книги) электрических цепей. Сейчас мы используем это соотношение для определения ранга цикломатической матрицы B_c.

Рассмотрим связный граф G на n вершинах и m дугах. Пусть B_f и Q_f — базисные цикломатическая матрица и матрица разрезающих множеств по отношению к остову T. Если столбцы матрицы B_f и Q_f расположены в одинаковом порядке, мы можем представить их в виде $B_f=[B_{ft}\ U]$, $Q_f=[U\ Q_{fc}]$. Согласно соотношению ортогональности, $B_fQ_f^t=0$, т. е.

$$[B_{ft}U]\begin{bmatrix} U \\ Q_{fc}^t \end{bmatrix}=0 \text{ или } B_{ft}=-Q_{fc}^t. \tag{6.13}$$

Пусть $\beta=[\beta_1,\ \beta_2,\ \ldots,\ \beta_\rho|\beta_{\rho+1},\ \ldots,\ \beta_m]$, где ρ — ранг графа G — является циклическим вектором, элементы которого расположены в том же порядке, что и столбцы матриц B_f и Q_f. Тогда, согласно соотношению ортогональности,

$$\beta Q_f^t=[\beta_1,\ \beta_2,\ \ldots,\ \beta_\rho\,|\,\beta_{\rho+1},\ \ldots,\ \beta_m]\begin{bmatrix} U \\ Q_{fc}^t \end{bmatrix}=0.$$

Поэтому $[\beta_1,\ \beta_2,\ \ldots,\ \beta_\rho]=-[\beta_{\rho+1},\ \beta_{\rho+2},\ \ldots,\ \beta_m]Q_{fc}^t=[\beta_{\rho+1},\ \beta_{\rho+2},\ \ldots,\ \beta_m]B_{ft}$. Используя это равенство, можно записать вектор $[\beta_1,\ \beta_2,\ \ldots,\ \beta_m]$ в следующем виде:

$$\begin{aligned}[\beta_1,\ \beta_2,\ \ldots,\ \beta_m]&=[\beta_{\rho+1},\ \beta_{\rho+2},\ \ldots,\ \beta_m][B_{ft}U]=\\&=[\beta_{\rho+1},\ \beta_{\rho+2},\ \ldots,\ \beta_m]B_f.\end{aligned} \tag{6.14}$$

Таким образом, любой циклический вектор можно выразить в виде линейной комбинации базисных циклических векторов. Поэтому $\operatorname{rank}(B_c)\leqslant\operatorname{rank}(B_f)=m-n+1$. Объединяя это неравенство с выражением (6.12), получаем следующую теорему и следствие из нее:

Теорема 6.7. Ранг цикломатической матрицы B_c связного графа G, имеющего n вершин и m дуг, равен $m-n+1$, т. е. цикломатическому числу графа G.

Следствие 6.7.1. Ранг цикломатической матрицы B_c графа G на n вершинах и m дугах с p компонентами равен $m-n+p$, т. е. цикломатическому числу графа G.

Пусть $\alpha=[\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_\rho,\ \alpha_{\rho+1},\ \ldots,\ \alpha_m]$ — вектор разреза, элементы которого расположены в том же порядке, что и столбцы матриц B_f и Q_f, тогда можно, исходя из соотношения $\alpha B_f^t=0$, доказать, что

$$\alpha=[\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_\rho]Q_f \tag{6.15}$$

с помощью процедуры, аналогичной той, что использовалась для установления соотношения (6.14). Следовательно, всякий вектор разреза можно выразить в виде линейной комбинации базисных векторов разрезающих множеств. Так как $\operatorname{rank}(Q_f)=n-1$, то rank

$(Q_c)=\text{rank}\,(Q_f)=n-1$. Это является альтернативным доказательством теоремы 6.4.

Заметим, что кольцевая сумма двух подграфов соответствует сумме по mod 2 соответствующих векторов. Поэтому для случая неориентированных графов равенства (6.14) и (6.15) просто повторяют утверждения, уже доказанные нами по ходу рассуждений, приведших к теореме 6.4, а именно: цикл (разрез) можно выразить в виде суммы по mod 2 базисных циклов (базисных разрезающих множеств).

6.5. Подматрицы матриц разрезов, инциденций и циклов

В этом разделе мы охарактеризуем те подматрицы матриц Q_c, A_c и B_c, которые соответствуют циклам, разрезам, остовам, коостовам, и изучим свойства этих подматриц.

Теорема 6.8. 1. Между столбцами матрицы разрезов Q_c, соответствующими дугам цикла, существует линейная зависимость.
2. Между столбцами цикломатической матрицы B_c, соответствующими дугам разрезающего множества, существует линейная зависимость.

Доказательство. 1. Пусть $Q_c=[Q^{(1)}, Q^{(2)}, \ldots, Q^{(m)}]$ — разбиение Q_c на столбцы; $\beta=[\beta_1, \beta_2, \ldots, \beta_m]$ — циклический вектор. Тогда, согласно соотношению ортогональности, имеем $Q_c\beta^t=0$ или

$$\beta_1 Q^{(1)}+\beta_2 Q^{(2)}+\ldots+\beta_m Q^{(m)}=0. \qquad (6.16)$$

Допустим, не нарушая общности, что первые r элементов вектора β — ненулевые, остальные равны нулю, тогда из выражения (6.16) получим $\beta_1 Q^{(1)}+\beta_2 Q^{(2)}+\ldots+\beta_r Q^{(r)}=0$. Таким образом, между столбцами $Q^{(1)}, Q^{(2)}, \ldots, Q^{(r)}$, соответствующими дугам цикла, существует линейная зависимость 2. Доказательство в этом случае проводится по такой же схеме, как и в п. 1.

Следствие 6.8.1. Между столбцами матрицы инциденций, соответствующими дугам цикла, существует линейная зависимость.

Доказательство. Это утверждение следует из п. 1 теоремы 6.8, поскольку матрица инциденций является подматрицей матрицы Q_c.

Теорема 6.9. Квадратная подматрица порядка $n-1$ любой усеченной матрицы инциденций A связного графа G на n вершинах является невырожденной тогда и только тогда, когда дуги, соответствующие столбцам подматрицы, образуют остов графа G.

Доказательство. Необходимость. Рассмотрим $n-1$ столбцов невырожденной подматрицы A. Поскольку эти столбцы линейно-независимы, то, согласно следствию 6.8.1, цикл в соответствующем подграфе графа G не присутствует. А поскольку ациклический подграф содержит $n-1$ дуг, то из теоремы 2.2 следует, что он является остовом графа G.

Достаточность следует из теоремы 6.1.

Таким образом, между остовами связного графа и невырожденными подматрицами матрицы A имеется взаимно-однозначное соответствие. Этот результат является фактически основой для получения формулы, определяющей число остовов графа, которая обсуждается в разд. 6.7.

Теорема 6.10. Рассмотрим связный граф G на n вершинах и m дугах. Пусть Q — подматрица матрицы Q_c с $n-1$ строкой и рангом $n-1$. Квадратная подматрица Q порядка $n-1$ является невырожденной тогда и только тогда, когда соответствующие столбцам подматрицы дуги образуют остов графа G.

Доказательство. Необходимость. Пусть столбцы матрицы Q расположены таким образом, что $Q=[Q_{11}Q_{12}]$, где Q_{11} — невырожденная подматрица. Поскольку столбцы подматрицы Q_{11} линейно-независимы, то, согласно п. 7 теоремы 6.8, в соответствующем подграфе графа G не имеется цикла. Этот подграф без циклов имеет $n-1$ дугу и, следовательно, по теореме 2.2 является остовом.

Достаточность. Пусть столбцы Q расположены так, что $Q=[Q_{11}Q_{12}]$, где столбцы подматрицы Q_{11} соответствуют дугам остова T. Базисная матрица разрезающих множеств Q_f по отношению к T имеет вид $Q_f=[UQ_{fc}]$. Так как матрицу Q можно выразить в виде линейной комбинации строк матрицы Q_f, то представим матрицу Q в виде $Q=[Q_{11}Q_{12}]=DQ_f=D[UQ_{fc}]$. Следовательно, $Q_{11}=DU=D$. D — невырожденная матрица, поскольку матрицы Q и Q_f имеют максимальный ранг $n-1$. Поэтому и Q_{11} — невырожденная подматрица, т. е. достаточность доказана.

Рассмотрим теперь двойственную теорему.

Теорема 6.11. Рассмотрим связный граф G на n вершинах и m дугах. Пусть B — подматрица цикломатической матрицы B_c графа G с $m-n+1$ строкой и рангом $m-n+1$. Квадратная подматрица B порядка $m-n+1$ является невырожденной тогда и только тогда, когда ее столбцы соответствуют дугам коостова.

Доказательство. Необходимость. Пусть столбцы матрицы B расположены так, что $B=[B_{11}B_{12}]$, где B_{11} — невырожденная подматрица. Поскольку ее столбцы линейно-независимы, то, согласно п. 2 теоремы 6.8, соответствующий подграф не содержит разрезов графа G. Этот подграф имеет, кроме того, $m-n+1$ дугу. Таким образом, он является максимальным (почему?) подграфом графа G, не содержащим разрезов, и, следовательно, коостовом (упражнение 2.18).

Достаточность. Пусть столбцы матрицы B расположены таким образом, что $B=[B_{11}B_{12}]$, при этом столбцы подматрицы B_{11} соответствуют дугам коостова графа G. Базисная цикломатическая матрица B_f по отношению к этому коостову имеет вид $B_f=[UB_{f1}]$. Поскольку строки матрицы B можно выразить в виде линейной комбинации строк матрицы B_f, представим матрицу B в виде $B=[B_{11}B_{12}]=DB_f=D[UB_{f1}]$, поэтому $B_{11}=DU=D$.

Но D — невырожденная матрица, так как матрицы B и B_f имеют максимальный ранг $m-n+1$. Следовательно, B_{11} — невырожденная подматрица.

Для иллюстрации теорем 6.10 и 6.11 рассмотрим граф G на рис. 6.1, *а*, а также матрицы Q_f и B_f [выражения (6.9) и (6.11) соответственно]. Квадратными подматрицами 4-го порядка матрицы Q_f, соответствующей остову $\{e_1, e_3, e_5, e_7\}$, и 3-го порядка матрицы B_f, соответствующей коостову $\{e_2, e_4, e_6\}$, являются матрицы

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{и} \quad \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & -1 \end{bmatrix}$$

Легко видеть, что определители этих матриц не равны нулю, следовательно, матрицы — невырожденные.

Завершая раздел, рассмотрим интересное свойство обращения невырожденной подматрицы усеченной матрицы инциденций.

Теорема 6.12. Пусть A_{11} — невырожденная подматрица порядка $n-1$ усеченной матрицы инциденций связного графа G на n вершинах. Тогда все ненулевые элементы A_{11}^{-1} равны либо 1, либо -1.

Доказательство. Пусть A — матрица инциденций, усеченная по строке, соответствующей вершине v_r. Пусть $A=[A_{11}A_{12}]$, где A_{11} — невырожденная подматрица. По теореме 6.9 дуги, соответствующие столбцам подматрицы A_{11}, об-

разуют остов T графа G. Базисная матрица разрезающих множеств Q_f по отношению к остову T будет представляться в виде $Q_f=[UQ_{fc}]$.

По теореме 6.3 всякий вектор разреза можно выразить в виде линейной комбинации строк матрицы инциденций. Поэтому множество Q_f можно представить в виде $Q_f=[UQ_{fc}]=D[A_{11}A_{12}]$. Следовательно, $D=A_{11}^{-1}$.

Рассмотрим теперь i-ю строку q_i матрицы Q_f. Пусть ей соответствует разрез $\langle V_a, \bar{V}_a\rangle$, при этом

$V_a=\{v_1, v_2, \ldots, v_k\}$, а $\bar{V}_a=\{v_{k+1}, v_{k+2}, \ldots, v_n\}$.

Допустим, что разрез $\langle V_a, \bar{V}_a\rangle$ ориентирован от V_a к $\bar{V}_a$. Тогда из выражения (6.5) следует, что

$$q_i=a_1+a_2+\ldots+a_k, \tag{6.17}$$

$$q_i=-(a_{k+1}+a_{k+2}+\ldots+a_n), \tag{6.18}$$

где a_j — j-я строка матрицы инциденций A_c.

Заметим, что соответствующая v_r строка a_r отсутствует в матрице A. Поэтому если $v_r\in V_a$, то для представления q_i линейной комбинацией строк матрицы A мы должны записать q_i в виде (6.18). Если $v_r\in\bar{V}_a$, то мы должны записать q_i в виде (6.17). В обоих случаях все ненулевые коэффициенты равны 1 либо —1.

Таким образом, все ненулевые элементы в любой строке матрицы $D=A_{11}^{-1}$ равны 1 или —1.

В доказательстве этой теоремы, по существу, предлагается следующая простая процедура нахождения A_{11}^{-1}:

1. Построить дерево T, для которого A_{11} — матрица инциденций, усеченная по строке, соответствующей v_n.
2. Пометить дуги дерева $T(e_1, e_2, \ldots, e_{n-1})$ таким образом, чтобы i-му столбцу матрицы A_{11} соответствовала e_i. Аналогично пометить вершины дерева $T(v_1, v_2, \ldots, v_{n-1})$ так, чтобы i-й строке матрицы A_{11} соответствовала v_i.
3. Пусть i-й столбец матрицы A_{11}^{-1} соответствует v_i. Для каждого $i(1\leqslant i\leqslant n-1)$ удалим e_i из дерева T, получая компоненты T_1 и T_2 ставшего несвязным графа. Пусть V_a — множество вершин дерева T_1, $\bar{V}_a$ — множество вершин дерева T_2, а e_i ориентирована от вершины из множества V_a к вершине из множества $\bar{V}_a$. Если $v_n\in V_a$, то приравнять —1 все элементы i-й строки матрицы A_{11}^{-1}, соответствующие вершинам множества $\bar{V}_a$, а остальные элементы — нулю. Если $v_n\in\bar{V}_a$, то приравнять 1 все элементы i-й строки, соответствующие вершинам множества V_a, а остальные элементы — нулю. Рассмотрим в качестве иллюстрации матрицу

$$A_{11}=\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}\begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \end{array}.$$

Дерево T, для которого A_{11} является усеченной матрицей инциденций, представлено на рис. 6.6, *а*. Матрица усечена по строке, соответствующей вершине v_5.

Для нахождения первой строки матрицы A_{11}^{-1} удалим e_1 из дерева T, получая две компоненты T_1 и T_2, представленные на рис. 6.6, *б*. Таким образом, $V_a=\{v_1, v_2, v_5\}$, $\bar{V}_a=\{v_3, v_4\}$. Поскольку вершина,

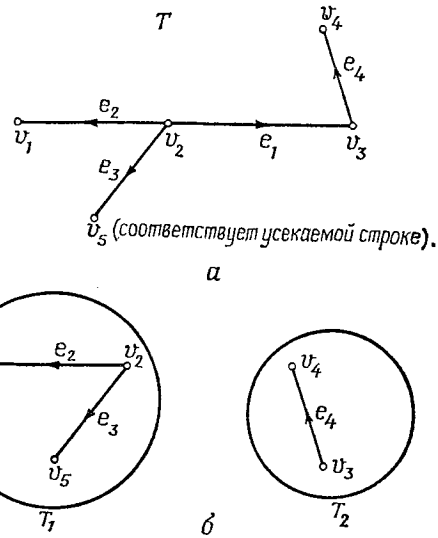


Рис. 6.6.
а—дерево T; *б*—поддеревья T_1 и T_2.

соответствующая усеченной строке, находится во множестве V_a, первая строка будет иметь значение —1 в элементах, соответствующих вершинам v_3 и v_4, находящимся во множестве $\bar{V}_a$, т. е. будет иметь вид [0 0 —1 —1].

Аналогичным образом получаем остальные строки матрицы A_{11}^{-1}. В результате получится следующая матрица:

$$A_{11}^{-1}=\begin{bmatrix} 0 & 0 & -1 & -1 \\ -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

6.6. Унимодулярные матрицы

Матрица называется *унимодулярной*, если определитель любой ее квадратной подматрицы равен 1, —1 или 0. В этом разделе мы покажем, что матрицы A_c, Q_f и B_f унимодулярны.

Теорема 6.13. Матрица инциденций A_c ориентированного графа унимодулярна.

Доказательство. Докажем теорему индукцией по порядку квадратной подматрицы A_c.

Очевидно, что определитель любой квадратной подматрицы A_c 1-го порядка равен 1, —1 или 0. В качестве индуктивного предположения примем, что определитель любой квадратной подматрицы порядка менее k равен 1, —1 или 0.

Рассмотрим произвольную невырожденную квадратную подматрицу A_c порядка k. Каждый ее столбец содержит не более двух ненулевых элементов (+1 и —1). Поскольку подматрица невырожденная, то не во всяком столбце имеются значения +1 и —1. По этой же причине подматрица не имеет столбцов, состоящих только из нулевых элементов. Таким образом, существует столбец, содержащий точно один ненулевой элемент. Разлагая определитель подматрицы по этому столбцу и учитывая индуктивное предположение, получаем, что искомый определитель равен ± 1.

Пусть Q_f — базисная матрица разрезов связного графа G на n вершинах по отношению к некоторому остову T, ветвями которого являются $b_1, b_2, \ldots, b_{n-1}$.

Пусть G' — граф, полученный из графа G отождествлением или замыканием концевых вершин одной из ветвей, например ветви b_1. Тогда $T - \{b_1\}$ — остов G'. Удалим теперь из матрицы Q строку, соответствующую ветви b_1 и получившуюся матрицу обозначим через Q'_f. Нетрудно показать, что Q'_f — базисная матрица разрезающих множеств G' по отношению к остову $T - \{b_1\}$. Таким образом, матрица, получившаяся в результате удаления из матрицы Q произвольной строки, является базисной матрицей разрезающих множеств некоторого связного графа. Обобщая этот результат, можно сделать вывод, что всякая матрица, образованная строками матрицы Q_f, является базисной матрицей разрезающих множеств некоторого связного графа.

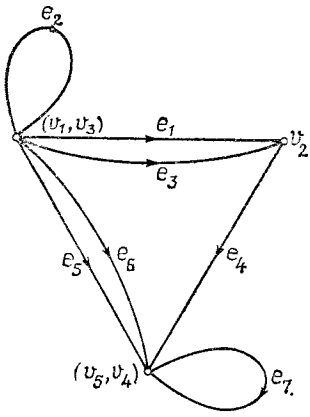


Рис. 6.7.

Рассмотрим, например, граф G на рис. 6.1, *а*. Базисная матрица разрезающих множеств представлена в (6.9). Подматрица, состоящая из двух строк матрицы Q_f, которые соответствуют ветвям e_1 и e_6, имеет вид

$$\begin{array}{c} \\ e_1 \\ e_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_6 & e_7 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}.$$

Легко проверить, что эта матрица — базисная матрица разрезающих множеств графа, представленного на рис. 6.7, по отноше-

нию к остову $\{e_1, e_6\}$. Этот граф получен из графа на рис. 6.1, *а* отождествлением концевых вершин e_2 и e_7.

Теорема 6.14. Любая базисная матрица разрезающих множеств Q_f связного графа G унимодулярна.

Доказательство. Пусть Q_f — базисная матрица разрезающих множеств графа G по отношению к остову T. Тогда $Q_f=[UQ_{fc}]$. Представим усеченную матрицу инциденций A графа в виде $A=[A_{11}A_{12}]$, где столбцы A_{11} соответствуют ветвям остова T. Из теоремы 6.9 мы знаем, что A_{11} — невырожденная подматрица. Запишем теперь матрицу Q_f в виде $Q_f=[UQ_{fc}]=A_{11}^{-1}[A_{11}A_{12}]$.

Если C — произвольная квадратная подматрица матрицы Q_f порядка $n-1$, где n — число вершин графа G, а D — соответствующая подматрица матрицы A, тогда $C=A_{11}^{-1}D$. Поскольку $\det D=\pm 1$ или 0, а $\det A_{11}^{-1}=\pm 1$, получаем

$$\det C=\pm 1 \text{ или } 0. \tag{6.19}$$

Рассмотрим произвольную квадратную подматрицу H матрицы Q_f порядка менее чем $n-1$. Из предшествующих теореме рассуждений следует, что H — подматрица базисной матрицы разрезающих множеств некоторого связного графа. Поэтому $\det H=\pm 1$ или 0, что доказывается так же, как и (6.19).

Таким образом, определитель всякой квадратной подматрицы матрицы Q равен ± 1 или 0 и, следовательно, матрица Q_f — унимодулярна.

Покажем сейчас, что матрица B_f также унимодулярна.

Теорема 6.15. Любая базисная цикломатическая матрица B_f связного графа G унимодулярна.

Доказательство. Пусть B_f и Q_f — базисные цикломатическая матрица и матрица разрезающих множеств графа G по отношению к остову T. Если $Q_f=[UQ_{fc}]$, то из (6.13) следует $B_f=[-Q_{fc}^tU]$. Так как матрица Q_{fc} унимодулярна, то унимодулярна также и матрица Q_{fc}^t. Элементарным упражнением теперь остается показать, что и матрица $[-Q_{fc}^tU]$ унимодулярна.

6.7. Число остовов

В этом разделе мы выведем формулу для определения числа остовов связного графа. Эта формула основана на теореме 6.9 утверждением теории матриц, известно под названием «теорема Бине — Коши».

Главным определителем матрицы называется определитель максимального порядка. Пусть P — матрица порядка $p\times q$, а Q — матрица порядка $q\times p$ $(p\leqslant q)$. Главные определители матриц P и Q имеют порядок p. Если главный определитель матрицы P состоит из столбцов $i_1, i_2, \ldots, i_p$ матрицы P, то соответствующий главный определитель матрицы Q — из строк матрицы $i_1, i_2, \ldots, i_p$ матрицы Q. Например, если

$$P=\begin{bmatrix}1 & -1 & 3 & 3\\ 2 & 2 & -1 & 2\end{bmatrix} \text{ и } Q=\begin{bmatrix}1 & 2\\ 2 & -1\\ -3 & 1\\ -1 & 2\end{bmatrix},$$

то для главного определителя

$$\begin{vmatrix}-1 & 3\\ 2 & -1\end{vmatrix}$$

матрицы P определитель

$$\begin{vmatrix} 2 & -1 \\ -3 & 1 \end{vmatrix}$$

является соответствующим главным определителем матрицы Q.

Теорема 6.16 (Бине—Коши). Если P — матрица порядка $p\times q$, а Q — матрица порядка $q\times p$ ($p\leqslant q$), то $\det(PQ)=\Sigma$. (Произведение соответствующих главных определителей P и Q).

Доказательство этой теоремы можно найти в работе [6.1].

Если в качестве иллюстрации к теореме Бине—Коши, как и ранее, взять матрицы P и Q, то получим

$$\det(PQ)=\begin{vmatrix}1&-1\\2&2\end{vmatrix}\begin{vmatrix}1&2\\2&-1\end{vmatrix}+\begin{vmatrix}1&3\\2&-1\end{vmatrix}\begin{vmatrix}1&2\\-3&1\end{vmatrix}+\begin{vmatrix}1&3\\2&2\end{vmatrix}\begin{vmatrix}1&2\\1&2\end{vmatrix}+$$

$$+\begin{vmatrix}-1&3\\2&-1\end{vmatrix}\begin{vmatrix}2&-1\\-3&1\end{vmatrix}+\begin{vmatrix}-1&3\\2&2\end{vmatrix}\begin{vmatrix}2&-1\\1&2\end{vmatrix}+\begin{vmatrix}3&3\\-1&2\end{vmatrix}\begin{vmatrix}-3&1\\1&2\end{vmatrix}=-167.$$

Теорема 6.17. Пусть G — связный неориентированный граф, A — усеченная матрица инциденций ориентированного графа, полученного присвоением ребрам графа G произвольной ориентации. Тогда $\tau(G)=\det(AA^t)$, где $\tau(G)$ — число остовов графа G.

Доказательство. По теореме Бине—Коши

$$\det(AA^t)=\sum\begin{pmatrix}\text{Произведение соответствующих}\\ \text{главных определителей } A \text{ и } A^t\end{pmatrix}. \qquad (6.20)$$

Заметим, что соответствующие главные определители A и A^t имеют одинаковое значение, равное 1, —1 или 0 (теорема 6.13). Поэтому каждое ненулевое слагаемое в правой части суммы (6.20) имеет значение 1. Кроме того, главный определитель A не равен нулю тогда и только тогда, когда дуги, соответствующие его столбцам, образуют остов.

Таким образом, между ненулевыми элементами правой части суммы (6.20) и остовами графа G существует взаимно-однозначное соответствие, что доказывает теорему.

Рассмотрим в качестве примера граф G, представленный на рис. 6.8, *а*. Ориентированный граф, полученный в результате присвоения произвольной ориентации ребрам графа G, представлен на рис. 6.8, *б*. Его матрица инциденций, усеченная по строке, которая соответствует вершине v_4 имеет вид

$$A=\begin{array}{c} \\ v_1\\ v_2\\ v_3\end{array}\begin{array}{c}\begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5\end{array}\\ \begin{bmatrix}1&0&0&1&0\\0&-1&1&-1&0\\0&0&-1&0&-1\end{bmatrix}\end{array}.$$

Следовательно,

$$AA^t=\begin{bmatrix}2&-1&0\\-1&3&-1\\0&-1&2\end{bmatrix} \text{ и } \det(AA)=8.$$

Граф G имеет восемь остовных деревьев: $\{e_1, e_4, e_5\}$, $\{e_1, e_3, e_4\}$, $\{e_1, e_3, e_5\}$, $\{e_1, e_2, e_3\}$, $\{e_1, e_2, e_5\}$, $\{e_3, e_4, e_5\}$, $\{e_2, e_3, e_4\}$, $\{e_2, e_4, e_5\}$.

Пусть $v_1, v_2, \ldots, v_n$ — вершины неориентированного графа G без петель. *Матрицей степеней* $K=[k_{ij}]$ графа G называется мат-

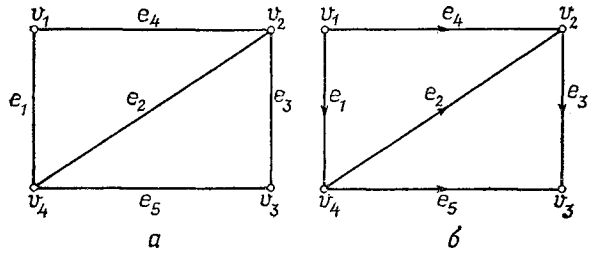


Рис. 6.8.

рица порядка $n\times n$, элементы которой определяются следующим образом:

$$k_{ij}=\begin{cases} -p, & \text{если } i\neq j \text{ и вершины } v_i \text{ и } v_j \\ & \text{связывают } p \text{ параллельных ребер;} \\ d(v_i), & \text{если } i=j. \end{cases}$$

Легко видеть, что $K=A_cA_c^t$ и не зависит от выбора ориентации ребер для получения матрицы инциденций A_c. Если A — матрица инциденций, усеченная по строке, соответствующей вершине v_r, то матрица AA^t получается удалением из матрицы K r-го столбца и r-й строки. Таким образом, используемую в теореме 6.17 матрицу AA^t можно найти просто в результате рассмотрения графа G.

Из определения матрицы степеней K ясно, что сумма всех элементов в ее любой строке равна нулю. Аналогично равна нулю сумма всех элементов в ее любом столбце. Квадратная матрица, обладающая такими свойствами, называется матрицей с равными алгебраическими дополнениями [6.2]. Таким образом, из теоремы 6.17 мы получаем следующий результат, принадлежащий Кирхгофу [6.3]:

Теорема 6.18. Все алгебраические дополнения матрицы степеней связного неориентированного графа имеют одинаковое значение, равное числу остовов графа G.

Теперь получим формулу определения числа различных деревьев, которые можно построить на n помеченных вершинах. Очевидно, это число равно числу остовных деревьев полного графа K_n на n помеченных вершинах.

Теорема 6.19 (Кэли). Существует n^{n-2} помеченных деревьев на $n\geqslant 2$ вершинах.

Доказательство. Матрица AA^t в случае полного графа K_n имеет вид

$$\begin{bmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ -1 & -1 & \dots & n-1 \end{bmatrix}.$$

По теореме 6.17 определитель этой матрицы дает число остовов графа K_n, являющееся числом помеченных деревьев на n вершинах.

Для вычисления $\det(AA^t)$ вычтем первый столбец матрицы AA^t из всех других столбцов той же матрицы. Тогда мы получим

$$\begin{bmatrix} n-1 & -n & -n & \dots & -n \\ -1 & n & 0 & \dots & 0 \\ -1 & 0 & n & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ -1 & 0 & 0 & \dots & n \end{bmatrix}.$$

Теперь прибавим первую строку получившейся матрицы к каждой оставшейся строке и получим

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & n & 0 & \dots & 0 \\ -1 & 0 & n & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ -1 & 0 & 0 & \dots & n \end{bmatrix}.$$

Определитель этой матрицы равен n^{n-2}. Это доказывает теорему, поскольку сложение строк и столбцов не меняет значения определителя.

Известно несколько доказательств теоремы Кэли [6.4—6.6].

6.8 Число остовных 2-деревьев

В этом разделе мы связываем алгебраические дополнения матрицы AA^t графа G с числом остовных 2-деревьев соответствующего типа. Для этой цели нам необходимы обозначения 2-деревьев, идентифицирующие вершины, которые находятся в разных компонентах. Будем обозначать через $T_{ijk\ \dots,\ rst\ \dots}$ остовные 2-деревья, в которых вершины v_i, v_j, v_k должны быть в одной компоненте, а вершины v_r, v_s, v_t — в другой. Число таких остовных 2-деревьев графа G будем обозначать через $\tau_{ijk\ \dots,\ rst\dots}$. Например, ребра e_1 и e_3 образуют в графе на рис. 6.8, *а* остовное 2-дерево типа $T_{14,23}$, а $\tau_{14,23}=1$.

Далее через A будем обозначать усеченную матрицу инциденций ориентированного графа, полученного присвоением произвольной

ориентации ребрам графа G, называя ее тем не менее усеченной матрицей инциденций G. Не нарушая общности, мы допускаем, что v_n соответствует усекаемой строке A, а i-я строка той же матрицы — вершине v_i. Символ Δ_{ij} будет обозначать алгебраическое дополнение элемента (i, j) матрицы AA^t.

Пусть A_{-i} — матрица, полученная удалением i-й строки из первоначальной матрицы A. Если G' — граф, полученный замыканием вершин v_i и v_n графа G, то, как легко убедиться, справедливы следующие утверждения:

1) A_{-i} — матрица инциденций графа G', усеченная по строке, соответствующей вершине v_n.
2) Множество ребер порождает остов графа G' тогда и только тогда, когда эти ребра образуют остовное 2-дерево $T_{i,n}$ в графе G.

Таким образом, между ненулевыми главными определителями A_{-i} и остовными 2-деревьями типа $T_{i,n}$ существует взаимно-однозначное соответствие.

Теорема 6.20. Для связного графа G $\Delta_{ii}=\tau_{i,n}$.

Доказательство.

Очевидно, что $\Delta_{ii}=\det(A_{-i}A^t_{-i})$. Это доказывает теорему, поскольку ненулевые главные определители A_{-i} соответствуют остовным 2-деревьям $T_{i,n}$ в графе G и наоборот.

Рассмотрим алгебраическое дополнение элемента (i, j) матрицы AA^t, имеющее вид

$$\Delta_{ij}=(-1)^{i+j}\det(A_{-i}A^t_{-j}). \tag{6.21}$$

По теореме Бине-Коши

$$\det(A_{-i}A^t_{-j})=\sum\begin{pmatrix}\text{Произведение соответствующих}\\ \text{главных определителей } A_{-i} \text{ и } A^t_{-j}\end{pmatrix}. \tag{6.22}$$

Всякий главный определитель A_{-i}, отличный от нуля, соответствует остовному 2-дереву типа $T_{i,n}$, а всякий ненулевой главный определитель A_j — остовному 2-дереву типа $T_{j,n}$. Поэтому ненулевые элементы суммы, стоящей в правой части выражения (6.22), соответствуют остовным 2-деревьям типа $T_{ij,n}$. Любое из этих ненулевых слагаемых равно определителю типа $\det(F_{-i}F^t_{-i})$, где F — усеченная матрица инциденций 2-дерева типа $T_{ij,n}$.

Теорема 6.21. Пусть F — матрица инциденций 2-дерева $T_{ij,n}$, усеченная по строке, которая соответствует вершине v_n. Если i-я ее строка соответствует вершине v_i, то $\det(F_{-i}F^t_{-j})=(-1)^{i+j}$.

Доказательство. Пусть T_1T_2 — компоненты 2-дерева $T_{ij,n}$. Допустим, что вершина v_n входит в T_2. Переставляя некоторые строки и соответствующие столбцы, можно записать матрицу FF^t в виде

$$S=\left[\begin{array}{c|c}C & 0\\ \hline 0 & D\end{array}\right],$$

где 1) C — матрица степеней компоненты T_1; 2) D получается из матрицы степеней компоненты T_2 удалением строки и столбца, соответствующих вершине v_n. Пусть строка k' матрицы S соответствует вершине v_k.

Перестановка некоторых строк и соответствующих столбцов матрицы не меняет значений ее алгебраических дополнений. Поэтому

$$(i', j') = \text{Алгебраическое дополнение элемента } (i', j') \text{ матрицы } S = \text{Алгебраическое дополнение элемента } (i, j) \text{ матрицы } FF^t = (-1)^{i+j} \det (F_{-i}F^t_{-j}). \tag{6.23}$$

По теореме 6.18 все алгебраические дополнения матрицы C имеют одинаковое значение, равное числу остовов компоненты T_1. Поэтому алгебраическое дополнение элемента (i', j') матрицы $C=1$. Более того, $\det D=1$. Поэтому

$$\text{Алгебраическое дополнение элемента } (i', j') \text{ матрицы } S = [\text{Алгебраическое дополнение элемента } (i', j') \text{ матрицы } C] \times [\det D] = 1. \tag{6.24}$$

Теперь из выражений (6.23), (6.24) получаем $\det (F_{-i}F^t_{-j} = (-1)^{i+j}$.

Доказательство этой теоремы приведено в работе [6.7]. Следующий результат получен Майедой:

Теорема 6.22. Для связного графа G справедливо равенство $\Delta_{ij} = \tau_{ij,\,n}$.

Доказательство. Поскольку каждый ненулевой член суммы в правой части выражения (6.22) равен определителю типа описанного в теореме 6.21, получаем $\det (A_{-i}A^t_{-j}) = (-1)^{i+j}\tau_{ij,\,n}$. Следовательно, $\Delta_{ij} = (-1)^{i+j}\det (A_{-i}A^t_{-j}) = \tau_{ij,\,n}$.

Для иллюстрации теорем 6.20 и 6.22 снова рассмотрим граф на рис. 6.8, *а*. Если A — матрица инциденций, усеченная по строке, соответствующей вершине v_4, то

$$A_{-2} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 \end{bmatrix} \text{ и } A_{-3} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & -1 & 0 \end{bmatrix}.$$

Поэтому

$$\tau_{2,\,4} = \det (A_{-2}A^t_{-2}) = \det \left(\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \right) = 4,$$

$$\tau_{23,\,4} = (-1)^{2+3} \det (A_{-2}A^t_{-3}) = -\det \left(\begin{bmatrix} 2 & -1 \\ 0 & -1 \end{bmatrix} \right) = 2.$$

Остовные 2-деревья типа $T_{2,4}$ образованы следующими множествами ребер: $\{e_4, e_5\}$, $\{e_3, e_4\}$, $\{e_1, e_3\}$, $\{e_1, e_5\}$, а остовные деревья типа $T_{23,4}$ — множествами $\{e_1, e_3\}$, $\{e_3, e_4\}$.

6.9 Число ориентированных остовов в ориентированном графе

В этом разделе мы обсуждаем предложенный в работе Татта [6.8] метод определения числа ориентированных остовных деревьев, имеющих определенную вершину в качестве корневой. Этот метод является, по существу, обобщением метода нахождения остовов данного ориентированного графа, предложенного в теореме 6.17, и описывается исходя из определяемой ниже матрицы полустепеней захода.

Матрица полустепеней захода $K=[k_{pq}]$ ориентированного графа $G=(V, E)$ без петель с $V=\{v_1, v_2, \ldots, v_n\}$ — это матрица порядка $n\times n$, определяемая следующим образом:

$$k_{pq}=\begin{cases} -w, & \text{если } p\neq q \text{ и существует } w \text{ параллельных дуг, идущих из } v_p \text{ в } v_q; \\ d^-(v_p), & \text{если } p=q. \end{cases}$$

Обозначим через K_{ij} матрицу, полученную удалением строки i и столбца j из первоначальной матрицы K. Метод Татта основан на следующей теореме:

Теорема 6.23. Ориентированный граф $G=(V, E)$ без петель с $V=\{v_1, v_2, \ldots, v_n\}$ является ориентированным деревом с корневой вершиной v_r тогда и только тогда, когда его матрица полустепеней захода обладает следующими свойствами:

1) $k_{pp}=\begin{cases}0, & \text{если } p=r,\\ 1, & \text{если } p\neq r,\end{cases}$ 2) $\det(K_{rr})=1$.

Доказательство.

Необходимость. Пусть данный ориентированный граф G является деревом с корневой вершиной v_r. Поскольку граф G — ациклический, то, как показано в разд. 5.7, можно пометить вершины графа G числами $1, 2, \ldots, n$ таким образом, что (i, j) является дугой графа G, только если $i<j$. Кроме того, корневая вершина при такой нумерации должна получить номер 1. Если в новой матрице полустепеней захода K' графа G i-е строка и столбец соответствуют вершине с номером i, тогда легко видеть, что матрица K' обладает следующими свойствами: $k'_{11}=0$; $k'_{pp}=1$ при $p\neq 1$; $k'_{pq}=0$, если $p>q$.

Поэтому $\det(K'_{11})=1$.

Матрицу K' можно получить перестановкой некоторых строк и соответствующих столбцов первоначальной матрицы K. Такая перестановка не меняет значения определителя любой подматрицы матрицы K. Следовательно, $\det(K_{rr})=\det(K'_{11})=1$.

Достаточность. Пусть матрица полустепеней захода K графа G удовлетворяет обоим условиям теоремы. Если граф G не является ориентированным деревом, то по условию 1 и п. 5 теоремы 5.4 он содержит цикл C. Корневая вершина v_r не может входить в цикл C, поскольку это означало бы $d^-(v_r)>0$ или $d^-(v)>1$ для некоторой другой вершины $v\in C$, что противоречит условию 1. Аналогичным образом можно показать, что 1) цикл C должен быть контуром; 2) никакая дуга, не принадлежащая циклу C, не инцидентна никакой вершине того же цикла.

Рассмотрим теперь подматрицу K_S матрицы K, состоящей из столбцов, которые соответствуют вершинам цикла C. Вследствие выполнения указанных свойств каждая строка матрицы K_S, соответствующая вершине цикла C, имеет точно одну ± 1. Все другие строки — нулевые. Таким образом, сумма столбцов матрицы K_S равна нулю. Другими словами, сумма соответствующих вершинам цикла C столбцов матрицы K равна нулю. Поскольку вершина v_r не входит в цикл C, это справедливо также и для матрицы K_{rr}, что противоречит условию 2. Это доказывает достаточность.

Применим метод Татта для нахождения числа τ_d ориентированных остовов ориентированного графа G с корневой вершиной v_r. Допустим, что граф G не имеет петель.

Через $K(g)$ для любого графа g будем обозначать его матрицу полустепеней захода, а через K' — матрицу, полученную из матрицы K

заменой r-го столбца на нулевой столбец. Набор подграфов G, в каждом из которых $d^-(v_r)=0$ и $d^-(v_p)=1$ при $p \neq r$, будем обозначать через S. Очевидно, что $|S| = \prod_{p=1}^{n} d^-(v_p)$. Кроме того, матрица полустепеней захода любого подграфа $g \in S$ удовлетворяет условию 1 теоремы 6.23.

Хорошо известно, что определитель квадратной матрицы является линейной функцией ее столбцов. Например, если $P=[p_1, p_2, \ldots, \ldots, p_i'+p_i'', \ldots, p_n]$ — квадратная матрица со столбцами $p_1, p_2, \ldots, p_i'+p_i'', \ldots, p_n$, то $\det P = \det [p_1, p_2, \ldots, p_i', \ldots, \ldots, p_n] + \det [p_1, p_2, \ldots, p_i'', \ldots, p_n]$.

Используя линейность функции определителя и свойство равенства нулю суммы элементов любого столбца матрицы $K'(G)$, запишем $\det K'(G)$ в виде суммы $|S|$ определителей, каждый из которых удовлетворяет условию 1 теоремы 6.23. Легко видеть, что между этими определителями и матрицами полустепеней захода подграфов из набора S существует взаимно-однозначное соответствие. Таким образом, $\det K'(G) = \sum_{g \in S} \det K'(g)$. Поэтому $\det K'_{rr}(G) = \sum_{g \in S} \det K'_{rr}(g)$. Однако поскольку $\det K'_{rr}(G) = \det K_{rr}(G)$, а $\det K'_{rr}(g) = \det K_{rr}(g)$ для всех $g \in S$, то $\det K_{rr}(G) = \sum_{g \in S} \det K_{rr}(g)$. Из теоремы 6.23 следует, что каждый определитель в правой части суммы — ненулевой и равен 1 тогда и только тогда, когда соответствующий подграф из набора S является ориентированным деревом. Таким образом, мы доказали следующую теорему:

Теорема 6.24. Пусть K — матрица полустепеней захода ориентированного графа G без петель. Пусть i-я строка этой матрицы соответствует вершине v_i графа G. Тогда число τ_d ориентированных остовов графа G, вершина v_r которого является корневой, определяется следующим образом: $\tau_d = \det K_{rr}$, где K_{rr} — матрица, полученная из первоначальной матрицы K удалением r-х строки и столбца.

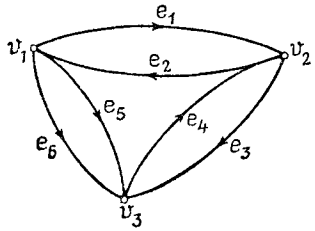


Рис. 6.9.

Проиллюстрируем эту теорему, а также рассуждения, которые приводят к ее доказательству.

Рассмотрим ориентированный граф G, приведенный на рис. 6.9. Определим число ориентированных остовов с корневой вершиной v_1.

Матрица полустепеней захода K графа G имеет вид

$$K=\begin{bmatrix} 1 & -1 & -2 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix} \text{ и } K'=\begin{bmatrix} 0 & -1 & -2 \\ 0 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}.$$

Запишем $\det K'$ в виде

$$\det K' = \begin{vmatrix} 0 & -1 & -2 \\ 0 & 2 & -1 \\ 0 & -1 & 3 \end{vmatrix} = \begin{vmatrix} 0 & -1 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 3 \end{vmatrix} + \begin{vmatrix} 0 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & -1 & 3 \end{vmatrix}$$

$$= \begin{vmatrix} 0 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} + \begin{vmatrix} 0 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} + \begin{vmatrix} 0 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{vmatrix}$$

$$+ \begin{vmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{vmatrix} + \begin{vmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{vmatrix} + \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{vmatrix}.$$

Шесть определителей в правой части этого равенства соответствуют подграфам, порожденным следующими множествами дуг: $\{e_1, e_5\}$, $\{e_1, e_6\}$, $\{e_1, e_3\}$, $\{e_4, e_5\}$, $\{e_4, e_6\}$, $\{e_4, e_3\}$. Удаляя из этих определителей первые строки и столбцы, получим

$$\det K'_{11} = \begin{vmatrix} 2 & -1 \\ -1 & 3 \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} + \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} + \begin{vmatrix} 1 & -1 \\ 0 & 1 \end{vmatrix}$$

$$+ \begin{vmatrix} 1 & 0 \\ -1 & 1 \end{vmatrix} + \begin{vmatrix} 1 & 0 \\ -1 & 1 \end{vmatrix} + \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} = 5$$

Граф G имеет 5 ориентированных остовов с корневой вершиной v_1, порождающихся следующими множествами дуг: $\{e_1, e_5\}$, $\{e_1, e_6\}$, $\{e_1, e_3\}$, $\{e_4, e_5\}$, $\{e_4, e_6\}$.

6.10. Матрица смежности

Пусть $G=(V, E)$ — ориентированный граф без параллельных дуг, в котором $V=\{v_1, v_2, \ldots, v_n\}$. *Матрицей смежности* $M=[m_{ij}]$ графа G называется матрица порядка $n \times n$, элементы которой m_{ij} определяются следующим образом:

$$m_{ij} = \begin{cases} 1, & \text{если } (v_i, v_j) \in E; \\ 0 & \text{в противном случае.} \end{cases}$$

Например, граф, изображенный на рис. 6.10, имеет следующую матрицу смежности:

$$M = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{array}.$$

В случае неориентированного графа $m_{ij}=1$ тогда и только тогда, когда существует ребро, соединяющее вершины v_i и v_j. Перейдем к изучению результатов, связанных с матрицей смежности.

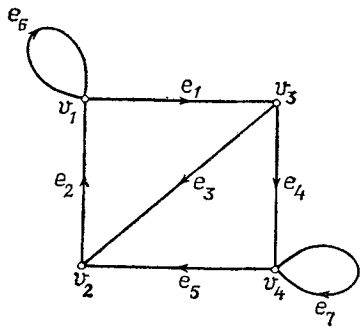


Рис. 6.10.

Теорема 6.25. (i, j)-й элемент m_{ij}^r матрицы M^r равен числу ориентированных маршрутов длины r из вершины v_i в вершину v_j.

Доказательство. Доказательство проводим индукцией по r. Очевидно, что результат справедлив при $r=1$.

В качестве индуктивного предположения допустим, что теорема верна для M^{r-1}. Тогда

$$m_{ij}^{(r)}=\sum_{k=1}^{n}\left(m_{ik}^{(r-1)}\cdot m_{kj}\right)=\sum_{k=1}^{n}\left(\begin{array}{l}\text{Число ориентированных маршрутов длины}\\ r-1 \text{ из вершины } v_i \text{ в вершину } v_k\end{array}\right)\times m_{kj}.$$

Для любого k соответствующий член суммы определяет число ориентированных маршрутов длины r из вершины v_i в вершину v_j, последней дугой которых является (v_k, v_j). Однако число ориентированных маршрутов длины r из v_i в $v_j=\sum_{k=1}^{n}$ (число ориентированных маршрутов длины r из v_i в v_j, последняя дуга которых идет из v_k в v_j), что доказывает теорему для M^r.

Рассмотрим, например, третью степень матрицы M графа, изображенного на рис. 6.10:

$$M^3=\begin{bmatrix}2 & 2 & 1 & 2\\ 1 & 1 & 1 & 1\\ 2 & 1 & 1 & 1\\ 2 & 1 & 1 & 1\end{bmatrix}.$$

Элемент (1,4) этой матрицы определяет число ориентированных маршрутов длины три из вершины v_1 в вершину v_4. Этими маршрутами являются (e_6, e_1, e_4) и (e_1, e_4, e_7).

Рассмотрим теперь важный результат, полученный в работе [6.9], который служит основой подхода сигнальных графов к решению линейных алгебраических уравнений. Для этого нам необходимо ввести несколько новых определений.

1-фактором ориентированного графа G называется остовный подграф графа G, в котором полустепень захода и полустепень исхода каждой вершины равны 1. Очевидно, что такой подграф представляет собой совокупность вершинно-непересекающихся контуров, включая петли графа G. Например, два 1-фактора графа на рис. 6.10 приведены на рис. 6.11, *а* и *б*.

Рассмотрим подстановку $j_1, j_2, \ldots, j_n$ целых чисел 1, 2, . . . , n. Подстановка называется четной, если для возвращения ее к перво-

начальной подстановке требуется четное число перестановок (обмена местами) ее элементов. Аналогично определяется нечетная подстановка. Рассмотрим, например, подстановку 1 3 4 2. К виду

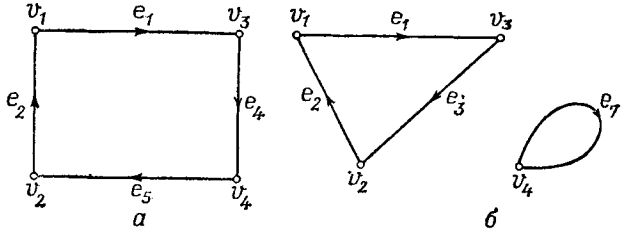


Рис. 6.11. Два 1-фактора графа рис. 6.10.

1 2 3 4 приводит следующая последовательность перестановок: 1) поменять местами 3 и 2; 2) поменять местами 3 и 4. Следовательно, эта подстановка является четной.

Теорема 6.26. Пусть H_i ($i=1, 2, \ldots, p$) — 1-факторы ориентированного графа G на n вершинах. Пусть L_i — число контуров H_i, а M — матрица смежности графа G. Тогда $\det M = (-1)^n \sum_{i=1}^{p} (-1)^{L_i}$.

Доказательство. По определению

$$\det M = \sum_{(j)} \varepsilon_{j_1, j_2, \ldots, j_n} m_{1j_1} \cdot m_{2j_2} \ldots m_{nj_n}, \tag{6.25}$$

где

1. $(j)=(j_1, j_2, \ldots, j_n)$ — подстановка целых чисел $1, 2, \ldots, n$.
2. $\varepsilon_{j_1, j_2, \ldots, j_n}=1$, если (j) — четная подстановка, в противном случае $\varepsilon_{j_1, j_2, \ldots, j_n}=-1$.
3. Сумма $\sum_{(j)}$ берется по всем подстановкам целых чисел $1, 2, \ldots, n$.

Ненулевой элемент $m_{1j_1} m_{2j_2} \ldots m_{nj_n}$ соответствует множеству дуг (v_1, v_{j_1}), (v_2, v_{j_2}), $\ldots$, (v_n, v_{j_n}). Каждая вершина v_k появляется в этом множестве дважды: один раз как первый элемент, в другой — как второй элемент в некоторых упорядоченных парах. Это означает, что полустепени захода и исхода любой вершины подграфа, порожденного дугами (v_1, v_{j_1}), (v_2, v_{j_2}), $\ldots$, (v_n, v_{j_n}), равны 1. Таким образом, всякий ненулевой элемент суммы выражения (6.25) соответствует 1-фактору графа G. Обратно: всякий 1-фактор соответствует ненулевому элементу $m_{1j_1} m_{2j_2} \ldots m_{nj_n}$.

Например, 1-фактор, приведенный на рис. 6.11, *б*, соответствует элементу $m_{13} m_{32} m_{21} m_{44}$.

Теперь необходимо зафиксировать знак $\varepsilon_{j_1, j_2, \ldots, j_n}$. Пусть C — контур в 1-факторе, соответствующем $j_1, j_2, \ldots, j_n$. Пусть контур состоит из ω дуг (v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), $\ldots$, $v_{i_\omega}, (v_{i_1})$.

Начальные вершины этих дуг образуют последовательность $i_1, i_2, \ldots, i_\omega$, а их конечные вершины — $i_2, i_3, \ldots, i_1$.

Легко видеть, что для приведения последовательности $i_2, i_3, \ldots, i_1$ к последовательности $i_1, i_2, \ldots, i_\omega$ достаточно совершить $\omega-1$ перестановку.

Пусть в 1-факторе, соответствующем $j_1, j_2, \ldots, j_n$, имеется L контуров, а их длины равны $\omega_1, \omega_2, \ldots, \omega_L$. Тогда для приведения последовательности j_1,

$j_2, \ldots, j_n$ к виду $1, 2, \ldots, n$ необходимо совершить $(\omega_1-1)+(\omega_2-1)+\ldots+(\omega_L-1)=W$ перестановок. Поэтому $\varepsilon_{j_1, j_2, \ldots, j_n}=(-1)^W=(-1)^{n+L}$.

Подводя итоги, заключаем:

1) Каждый ненулевой элемент $m_{1j_1}\ldots m_{nj_n}$ соответствует 1-фактору H_i графа G (заметим, что значение этого элемента равно 1);

2) $\varepsilon_{j_1, j_2, \ldots, j_n}=(-1)^{n+L_i}$, где L_i — число контуров в 1-факторах H_i.

Таким образом, выражение (6.25) сводится к следующему $\det M=(-1)^n\sum_{i=1}^{p}(-1)^{L_i}$.

Рассмотрим, например, 1-факторы, приведенные на рис. 6.11. Им соответствуют следующие значения L_i: $L_1=1$, $L_2=2$. Следовательно, определитель матрицы смежности графа на рис. 6.10 равен $(-1)^1+(-1)^2=0$. В этом легко убедиться путем непосредственного разложения определителя.

Пусть каждой дуге (i, j) ориентированного графа G присвоен вес w_{ij}. Определим *матрицу смежности* $M=[m_{ij}]$ графа G следующим образом:

$$m_{ij}=\begin{cases} w_{ij}, & \text{если существует дуга } (v_i, v_j); \\ 0 & \text{в противном случае.} \end{cases}$$

Определим весовое произведение $w(H)$ подграфа H графа G как произведение весов всех дуг подграфа H. Если последний не имеет дуг, т. е. подграф H является пустым, то пусть $w(H)=1$. Теперь легко получить следующее обобщение теоремы 6.26:

Теорема 6.27. Определитель матрицы смежности M взвешенного графа G на n вершинах определяется следующим выражением:

$$\det M=(-1)^n\sum_{H}(-1)^{L_H}w(H),$$

где H—1-фактор графа G, а L_H — число контуров в подграфе H.

6.11. Графы Коутса и Мэзона

В этом разделе мы рассматриваем теоретико-графовый подход к решению линейных алгебраических уравнений. Обсуждаются два тесно связанных метода, принадлежащих Коутсу [6.10] и Мэзону [6.11, 6.12].

6.11.1. Метод Коутса

Рассмотрим линейную систему, описываемую системой уравнений

$$AX=Bx_{n+1}. \tag{6.26}$$

где $A=[a_{ij}]$ — невырожденная матрица порядка $n\times n$, X — вектор-столбец неизвестных переменных $x_1, x_2, \ldots, x_n$. B — вектор-столбец элементов $b_1, b_2, \ldots, b_n$, а x_{n+1} — входная переменная.

Решение для x_k можно записать в виде

$$x_k/x_{n+1} = \sum_{i=1}^{n} b_i \Delta_{ik}/\det A, \tag{6.27}$$

где Δ_{ik} — алгебраическое дополнение элемента (i, k) матрицы A.

Пусть A' — матрица, полученная добавлением $-B$ к правой части матрицы A и затем к нижней части получившейся матрицы нулевой строки. Свяжем взвешенный ориентированный граф $G_c(A')$ с матрицей A'.

Пусть $G_c(A')$ имеет $n+1$ вершину $x_1, x_2, \ldots, x_{n+1}$. Если $a_{ji}=0$, то пусть в $G_c(A')$ будет дуга (x_i, x_j), имеющая вес a_{ji}. Очевидно, что A' — транспонированная матрица смежности $G_c(A')$. Граф $G_c(A')$ называется *графом потоков Коутса* или просто *графом Коутса*, связанным с матрицей A'. Будем также говорить, что граф Коутса связан с системой уравнений (6.26).

Рассмотрим в качестве примера систему уравнений

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & -1 & -2 \\ 2 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} x_4. \tag{6.28}$$

В этом случае матрица A' имеет вид

$$A' \begin{bmatrix} 2 & 1 & 1 & -1 \\ 1 & -1 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Граф Коутса $G_c(A')$, связанный с системой уравнений (6.28), представлен на рис. 6.12, *а*.

Заметим, что каждую вершину $x_i (1 \leqslant i \leqslant n)$ графа $G_c(A')$ можно рассматривать как представляющую уравнение в системе (6.26). Например, i-е уравнение в (6.26) можно получить, приравнивая к нулю сумму произведений весов дуг, заходящих в вершину x_i, и переменных, соответствующих вершинам, из которых эти дуги исходят. Кроме того, удаляя из графа $G_c(A')$ вершину x_{n+1}, можно получить граф Коутса $G_c(A)$, связанный с матрицей A. В случае системы уравнений (6.28) граф Коутса $G_c(A)$ представлен на рис. 6.12, *б*.

Поскольку A — транспонированная матрица смежности $G_c(A)$, а сама матрица и ее транспонированная матрица имеют равные определители, из теоремы 6.27 следует, что

$$\det A = (-1)^n \sum_{H} (-1)^{L_H} w(H), \tag{6.29}$$

где H — 1-фактор графа $G_c(A)$, $w(H)$ — весовое произведение подграфа H, а L_H — число контуров в нем. Таким образом знаменатель выражения (6.27) можно выразить через весовые произведения 1-факторов графа $G_c(A)$. Для вывода подобного выражения для числителя формулы (6.27) необходимо выражение для Δ_{ik}.

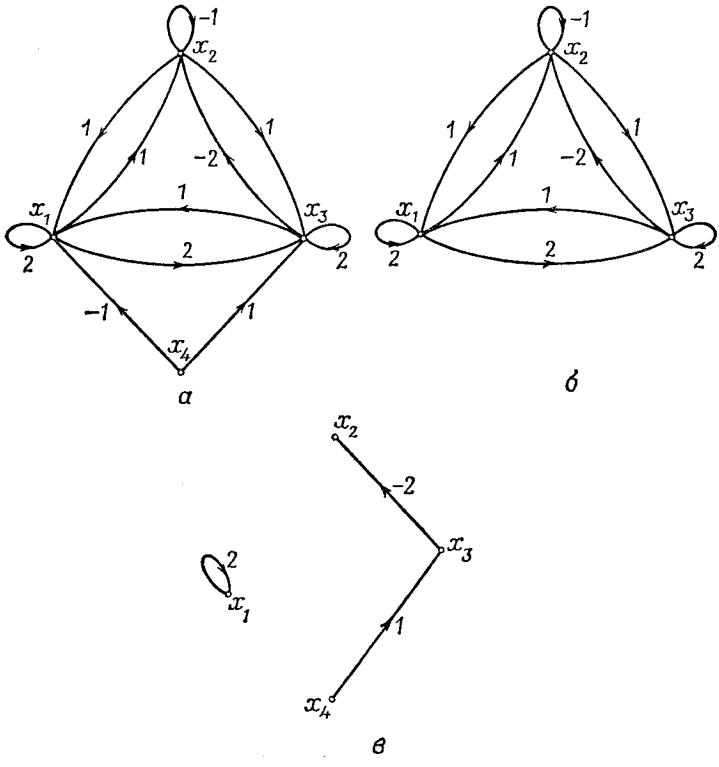


Рис. 6.12.

а — граф Коутса $G_c(A')$; *б* — граф $G_c(A)$; *в* — 1-факториальное соединение $H_{4,2}$ графа $G_c(A')$.

1-факториальным соединением H_{ij} вершины x_i с вершиной x_j в графе $G_c(A)$ является остовный подграф графа $G_c(A)$, который содержит а) ориентированный путь P из вершины x_i в вершину x_j; б) множество вершинно-непересекающихся контуров, содержащих все вершины графа $G_c(A)$, за исключением контуров, входящих в путь P; 1-факториальное соединение, например вершины x_4 с вершиной x_2 графа $G_c(A')$ (рис. 6.12, *а*), представлено на рис. 6.12, *в*.

Теорема 6.28. Пусть $G_c(A)$ — граф Коутса, связанный с матрицей A порядка $n \times n$. Тогда

1) $\Delta_{ii} = (-1)^{n-1} \sum_H (-1)^{L_H} w(H)$;

2) $\Delta_{ij}=(-1)^{n-1}\sum_{H_{ij}}(-1)^{L'_H}w(H_{ij})$, $i\neq j$, где H—1=фактор графа, полученного из $G_c(A)$ удалением вершины x_i, H_{ij}—1-факториальное соединение в $G_c(A)$ вершины x_i с вершиной x_j; L_H и L'_H — число контуров в H и H_{ij} соответственно.

Доказательство. 1. Следует из теоремы 6.27.

2. Заметим, что Δ_{ij} — определитель матрицы, полученной из матрицы A заменой j-го столбца на столбец, состоящий из нулевых элементов, за исключением i-го, который равен 1. Обозначим эту матрицу через A_α. Тогда граф Коутса $G_c(A_\alpha)$ получается из исходного графа $G_c(A)$ удалением всех дуг, исходящих из вершины x_j (включая и петли, инцидентные вершине x_j), и введением дуги (x_j, x_i) с весом 1. Теперь из теоремы 6.27 получаем

$$\Delta_{ij}=\det A_\alpha=(-1)^n\sum_{H_\alpha}(-1)^{L_\alpha}w(H_\alpha), \tag{6.30}$$

где H_α—1-фактор графа $G_c(A)$, а L_α — число контуров в H_α.

В каждом 1-факторе H_α обязательно должна содержаться введенная дуга с весом 1. Удаляя эту дугу из 1-фактора H_α, мы получаем 1-факториальное соединение H_{ij} графа $G_c(A)$. При этом $w(H_\alpha)=w(H_{ij})$. Легко убедиться также в том, что между H_α в графе $G_c(A_\alpha)$ и H_{ij} в графе $G_c(A)$ существует такое взаимно-однозначное соответствие, что $w(H_\alpha)=w(H_{ij})$. Поскольку число контуров в 1-факториальном соединении H_{ij} на единицу меньше, чем в H_α, $L'_H=L_\alpha-1$. Поэтому из выражения (6.30) получаем $\Delta_{ij}=(-1)^{n-1}\sum_{H_{ij}}(-1)^{L'_H}w(H_{ij})$.

Рассмотрим теперь числитель выражения (6.27) $\sum_{i=1}^{n}b_i\Delta_{ik}$. Эта сумма равна определителю матрицы, полученной из матрицы A заменой j-го столбца на B. Легко связать его с $\det A'_{n+1,\,k}$ (где $A'_{n+1,\,k}$ — матрица, полученная из матрицы A' удалением $(n+1)$-й строки и k-го столбца):

$$\sum_{i=1}^{n}b_i\Delta_{ik}=(-1)^{(n+1)-k-1}(-1)\det(A'_{n+1,\,k}). \tag{6.31}$$

Из части 2 теоремы 6.28 получаем

$$(-1)^{n+1+k}\det(A'_{n+1,\,k})=(-1)^n\sum_{H_{n+1,\,k}}(-1)^{L'_H}w(H_{n+1,\,k}), \tag{6.32}$$

где L'_H — число контуров в 1-факториальном соединении $H_{n+1,k}$ в графе $G_c(A')$. Объединяя выражения (6.31) и (6.32), получим

$$\sum_{i=1}^{n}b_i\Delta_{ik}=(-1)^n\sum_{H_{n+1,\,k}}(-1)^{L'_H}w(H_{n+1,\,k}). \tag{6.33}$$

Из выражений (6.29) и (6.33) получаем следующую теорему:

Теорема 6.29. Если матрица коэффициентов A невырожденная, то решение уравнения (6.26) определяется следующим образом:

$$\frac{x_k}{x_{n+1}}=\frac{\sum_{H_{n+1,\,k}}(-1)^{L'_H}w(H_{n+1,\,k})}{\sum_{H}(-1)^{L_H}w(H)}. \tag{6.34}$$

1. $H_{n+1,\,k}$ — 1-факториальное соединение вершины x_{n+1} с вершиной x_k в графе $G_c(A')$.
2. H — 1-фактор графа $G_c(A)$.
3. L'_H и L_H — число циклов в $H_{n+1,\,k}$ и H соответственно.

Уравнение (6.34) называется формулой *коэффициента усиления Коутса*. Проиллюстрируем применение этой формулы, решая уравнение (6.28) относительно x_2/x_4.

1-факторы графа $G_c(A)$ (рис. 6.12, *б*), связанные с матрицей A из уравнения (6.28), приведены ниже со своими весовыми произведениями. Различные контуры в 1-факторе отличаются порядком вершин в скобках

1-фактор	*Весовое произведение*
$(x_1)(x_2)(x_3)$	-4
$(x_1)(x_2, x_3)$	-4
$(x_2)(x_1, x_3)$	-2
$(x_3)(x_1, x_2)$	2
(x_1, x_2, x_3)	1
(x_1, x_3, x_2)	-4

Подсчитаем числитель выражения (6.34):

$$\sum_H (-1)^{L_H} w(H) = (-1)^3(-4) + (-1)^2(-4) + (-1)^2(-2) + \\ + (-1)^2(2) + (-1)(1) + (-1)(-4) = 4 - 4 - 2 + 2 - 1 + 4 = 3.$$

Приведем 1-факториальные соединения x_4 с x_2 в $G_c(A')$ (рис. 6.12, *а*). Сейчас в скобки заключаются также вершины, лежащие в ориентированном пути.

1-факториальное соединение	*Весовое произведение*
$(x_4, x_1, x_2)(x_3)$	-2
(x_4, x_1, x_3, x_2)	4
$(x_4, x_3, x_2)(x_1)$	-4
(x_4, x_3, x_1, x_2)	1

Подсчитаем числитель выражения (6.34):

$$\sum_{H_{n+1,\,k}} (-1)^{L'_H} w(H_{n+1,\,k}) = (-1)(-2) + 4 + (-1)(-4) + 1 = \\ = 2 + 4 + 4 + 1 = 11.$$

Таким образом, $x_2/x_4 = 11/3$.

6.11.2. Метод Мэзона

Для иллюстрации метода Мэзона перепишем уравнение (6.26) в виде

$$x_j = (a_{jj}+1)x_j + \sum_{\substack{k=1 \\ k\neq j}}^{n} a_{jk}x_k - b_j x_{n+1}, \quad j = 1, 2, \ldots, n;$$

$$x_{n+1} = x_{n+1}.$$

Можно представить эти уравнения в матричном виде следующим образом: $(A'+U_{n+1})X' = X'$, где A' — определенная ранее матрица

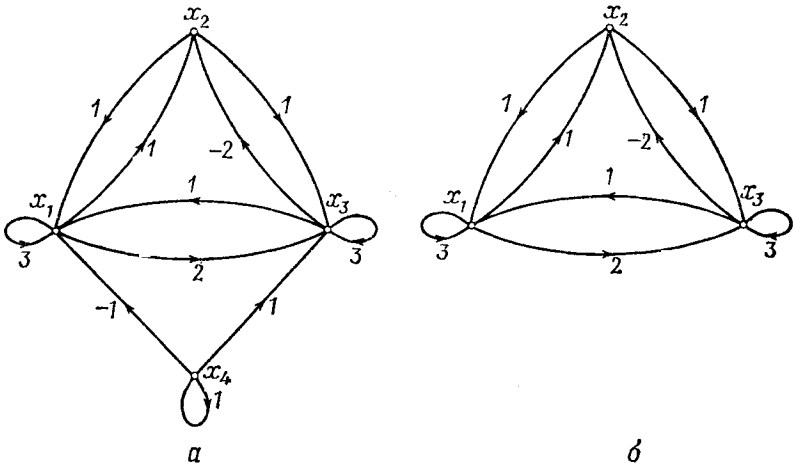


Рис. 6.13.
а — граф Мэзона $G_m(A')$, *б* — граф $G_m(A)$.

порядка $(n+1)\times(n+1)$, U_{n+1} — единичная матрица порядка $n+1$, а X' — вектор-столбец переменных $x_1, x_2, \ldots, x_{n+1}$.

Граф Коутса $G_c(A'+U_{n+1})$ называется *сигнальным графом потоков Мэзона* или просто *графом Мэзона*, связанным с матрицей A', и обозначается через $G_m(A')$. Например, на рис. 6.13 приведены граф Мэзона $G_m(A')$ и граф $G_m(A)$, связанный с системой уравнений (6.28).

Каждую вершину в графе $G_m(A')$ можно рассматривать как переменную. Если между вершинами x_i и x_j есть дуга, то можно считать, что переменная x_i делает вклад $(a_{ji}x_i)$ в переменную x_j. Другими словами, x_j равна сумме произведений весов дуг, заходящих в вершину x_j и переменных, соответствующих вершинам, из которых исходят эти дуги. Таким образом, граф Мэзона — удобное графическое представление потока переменных в системе.

Заметим, что для получения графа Коутса из данного графа Мэзона из веса каждой петли мы просто вычитаем единицу, а на

каждую вершину графа Мэзона без петли вводим петлю с весом —1. Иначе говоря, граф Коутса G_c можно получить из графа Мэзона G_m простым введением на каждую вершину петли с весом —1. Будем обозначать множество таких петель веса —1, добавляемых для построения графа G_c, через S. Отметим, что построенный таким образом граф G_c будет иметь в каждой вершине, самое большее, две и в то же время не менее одной петли.

Рассмотрим связанный с матрицей A граф Мэзона $G_m(A)$; пусть граф G_c будет соответствующим графом Коутса. Пусть H — 1-фактор графа G_c, имеющий j петель из множества S. Пусть L_{Q+j} — общее число контуров H. При удалении из H j добавленных петель множества S получим первоначальный подграф Q графа G_m, являющийся набором из L_Q вершинно-непересекающихся контуров. Кроме того, при $j < n$

$$w(H) = (-1)^j w(Q). \tag{6.35}$$

Заметим, что если $j=n$, то Q — пустой подграф G_m, имеющий по определению вес $w(Q)=1$. Поэтому в этом случае

$$w(H) = (-1)^n. \tag{6.36}$$

Легко видеть также, что всякому подграфу Q графа G_m, являющемуся набором из L_Q вершинно-непересекающихся контуров, в графе G_c соответствует единственный 1-фактор, который получается введением на вершины, не входящие в Q, петель (из множества S) весом —1.

По теореме 6.27 мы имеем

$$\det A = (-1)^n \sum_H (-1)^{L_Q+j} w(H) = (-1)^n \left[1 + \sum_Q (-1)^{L_Q} w(Q)\right]. \tag{6.37}$$

Последний шаг в этом выводе следует из выражений (6.35) и (6.36). Выражение (6.37) можно представить в виде

$$\det A = (-1)^n \left[1 - \sum_j Q_{j1} + \sum_j Q_{j2} - \sum_j Q_{j3} + \dots\right], \tag{6.38}$$

где Q_{jk} — весовое произведение j-го подграфа G_m, являющегося набором из k вершинно-непересекающихся контуров.

Таким образом, мы выразили знаменатель выражения (6.27) в весовых произведениях соответствующих подграфов графа Мэзона.

Назовем $(-1)^n \det(A)$ определителем графа $G_m(A)$ и обозначим его через Δ. Тогда, используя выражение (6.33) и рассуждая точно так же, как при выводе выражения (6.38), выразим числитель в виде

$$\sum_{i=1}^{n} b_i \Delta_{ik} = (-1)^n \sum_j w(P^j_{n+1,\,k}) \Delta_j,$$

где $P^j_{n+1,\,k}$ — j-й ориентированный путь из вершины x_{n+1} к вершине x_k в графе $G_m(A')$, а Δ_j — определитель подграфа того же графа,

вершинно-непересекающегося с j-м ориентированным путем $P^j_{n+1,\,k}$. Таким образом, приходим к следующему утверждению:

Теорема 6.30. Если матрица коэффициентов A в выражении (6.26) — невырожденная, то

$$\frac{x_k}{x_{n+1}} = \frac{\sum_j w(P^j_{n+1,\,k})\,\Delta_j}{\Delta}, \quad k = 1, 2, \ldots, n, \tag{6.39}$$

где $P^j_{n+1,\,k}$ — j-й ориентированный путь в графе $G_m(A')$ от вершины x_{n+1} к вершине x_k, Δ_j — определитель подграфа $G_m(A')$, вершинно-непересекающегося с j-м ориентированным путем $P^j_{n+1,\,k}$, а Δ — определитель графа $G_m(A)$.

Уравнение (6.39) называется *формулой коэффициента усиления Мэзона*. В литературе по теории систем $P^j_{n+1,\,k}$ — прямой путь от вершины x_{n+1} к вершине x_k. Контуры $G_m(A')$ называются *петлями обратной связи*. Проиллюстрируем применение формулы (6.39) к решению уравнения (6.28) относительно x_2/x_4.

Приведем различные наборы вершинно-непересекающихся контуров $G_m(A)$ (рис. 6.13, *б*) и их весовые произведения.

Контур	*Весовое произведение*
(x_1)	3
(x_3)	3
(x_1, x_2)	1
(x_2, x_3)	−2
(x_1, x_3)	2
(x_1, x_2, x_3)	1
(x_1, x_3, x_2)	−4
$(x_1)(x_3)$	9
$(x_1)(x_2, x_3)$	−6
$(x_3)(x_1, x_2)$	3

Вычислим знаменатель выражения (6.39): $\Delta = 1 - (3+3+1-2+2+1-4) + (9-6+3) = 3$.

Приведем различные ориентированные пути $G_m(A')$ (рис. 6.13, *а*) из вершины x_4 в вершину x_2 вместе с их весовыми произведениями:

j	$P^j_{4,2}$	*Весовое произведение*
1	(x_4, x_1, x_2)	−1
2	(x_4, x_1, x_3, x_2)	4
3	(x_4, x_3, x_2)	−2
4	(x_4, x_3, x_1, x_2)	1

Контурами, вершинно-непересекающимися с прямыми путями $P^1_{4,2}$ и $P^3_{4,2}$, являются петли (x_3) и (x_1) соответственно. Поэтому $\Delta_1 = 1 - 3 =$

$=-2$, $\Delta_3=1-3=-2$. Контуров, вершинно-пересекающихся с прямыми путями $P^2_{4,2}$ и $P^4_{4,2}$, нет, т. е. $\Delta_2=1$, $\Delta_4=1$.

Таким образом, числитель выражения (6.39) имеет вид $P^1_{4,2}\Delta_1+P^2_{4,2}\Delta_2+P^3_{4,2}\Delta_3+P^4_{4,2}\Delta_4=2+4+4+1=11$. Поэтому $x_2/x_4=11/3$.

6.12. Замечания, касающиеся литературы

Результаты, полученные в этой главе, образуют основу для большинства результатов части II книги. Рассмотренные в главе темы обсуждаются также в работах [6.2, 6.13—6.15]. Кроме того, в работе [6.2] приводятся исторические подробности, касающиеся представленных здесь результатов.

Монография [6.16] посвящена исключительно задачам перечисления графов, в частности относящимся к непомеченным графам.

Задаче подсчета числа остовов уделено значительное внимание и в работах по теории электрических цепей. Получены рекуррентные отношения для определения числа остовов специальных классов графов [6.17—6.21]. В работе [6.22] содержатся формулы для определения числа остовов с заданными свойствами.

В работе [6.2] широко обсуждаются некоторые аспекты, касающиеся подхода сигнальных графов потоков. Применения сигнальных графов потоков рассмотрены в работе [6.23]. Применение сигнальных графов потоков при исследовании электрических цепей описано в работе [6.24].

Упражнения

6.1. Пусть $Q_f=[q_{ij}]$ — базисная матрица разрезающих множеств, а $B_f=[b_{ij}]$ — базисная цикломатическая матрица связного ориентированного графа. Докажите, что а) если произведения $(q_{ij}q_{kj})$ и $(q_{ir}q_{kr})$ не равны нулю, то они равны между собой, б) выполните п. а для матрицы B_f.

6.2. Пусть B — матрица, образованная любыми $\mu(G)$ независимыми строками цикломатической матрицы связного ориентированного графа G, имеющего m дуг. Покажите, что если F — произвольная матрица порядка $\rho(G)\times m$ и ранга $\rho(G)$, имеющая элементы, равные 1, —1 и 0, и удовлетворяющая условию $BF^t=0$, то каждая строка матрицы F — это строка матрицы разрезов графа G.

6.3. Пусть Q — матрица, образованная $\rho(G)$ независимыми строками матрицы разрезов связного ориентированного графа G, имеющего m дуг. Покажите, что если F — произвольная матрица порядка $\mu(G)\times m$ и ранга $\mu(G)$, имеющая элементы, равные 1, —1 и 0, и удовлетворяющая условию $QF^t=0$, то каждая строка матрицы F соответствует циклу или объединению реберно-непересекающихся циклов G.

6.4. Пусть $Q(x, y)$ — подматрица матрицы разрезов Q_c связного неразделимого ориентированного графа G; каждая содержит только те строки Q_c, которые соответствуют разрезам, разделяющим вершины x и y. Покажите, что $Q(x, y)$ содержит базисную матрицу разрезающих множеств графа G. (Более общий результат приведен в работе [6.25].)

6.5. Пусть B — подматрица цикломатической матрицы B_c планарного графа G, содержащая только те строки B_c, которые соответствуют ячейкам графа G. Покажите, что подматрица B унимодулярна. (Определение планарного графа и ячеек дано в гл. 7.)

6.6. Пусть B — матрица, образованная любыми $\mu(G)$ независимыми строками цикломатической матрицы связного ориентированного графа G. Покажите, что все главные определители невырожденных подматриц матрицы B имеют одинаковое значение.

6.7. Докажите теорему 4.11.

6.8. Докажите, что для связного ориентированного графа $\tau(G)=\det(B_f B_f^t)=$ $=\det(Q_f Q_f^t$, где B_f — базисная цикломатическая матрица, а Q_f — базисная матрица разрезающих множеств графа G.

6.9. Пусть Q — матрица, образованная любыми $\rho(G)$ независимыми строками матрицы разрезов связного ориентированного графа G, а B — матрица, образованная любыми $\mu(G)$ независимыми строками цикломатической матрицы G. Покажите, что матрица $\begin{bmatrix} Q \\ B \end{bmatrix}$ — невырожденная.

6.10. Докажите, что подпространства над полем $GF(2)$ циклов W_C и разрезов W_S связного неориентированного графа G являются ортогональными дополнениями векторного пространства W_G тогда и только тогда, когда число остовов графа G нечетно.

6.11. Пусть для любого ребра e графа G произведение $G\cdot e$ обозначает граф, полученный из графа G стягиванием ребра e. Покажите, что если граф G связный, то $\tau(G)=\tau(G-e)+\tau(G\cdot e)$.

6.12. Используя рекурсивную формулу из упражнения 6.11, найдите число остовов колеса на n вершинах. (Определение колеса см. в разд. 8.4.)

6.13. Покажите, что если e — ребро графа K_n, то $\tau(K_n-e)=(n-2)n^{n-3}$.

6.14. а) Пусть H — граф на n вершинах, в котором между всякой парой смежных вершин имеется k параллельных дуг. Пусть G — простой граф, лежащий в основе графа H. Покажите, что $\tau(H)=k^{n+1}\tau(G)$.

б) Пусть H — граф, полученный из связного графа G на n вершинах заменой каждого ребра этого графа G на путь длин k. Покажите, что $\tau(H)=k^{m-n+1}\tau(G)$, где m — число ребер в графе G.

6.15. Покажите, что теорема 6.17 является частным случаем теоремы 6.24.

6.16. Граф G с весами на дугах называется *взвешенным графом*. Напомним, что весовое произведение подграфа графа G равно произведению весов его дуг и что если подграф имеет единственную изолированную вершину, то его весовое произведение по определению равно 1.

Пусть A — матрица инциденций связного ориентированного графа G, усеченная по строке, соответствующей вершине v_n, а i-я строка матрицы A соответствует вершине v_i. Пусть W — диагональная матрица, представляющая веса графа G. Примем, что столбцы A и W, соответствующие дугам, расположены в одинаковом порядке. Покажите, что а) $\det(AWA^t)$ — сумма весовых произведений всех остовов графа G и б) Δ_{ij} — сумма весовых произведений всех остовных 2-деревьев $T_{ij,n}$ графа G, где Δ_{ij} — алгебраическое дополнение элемента (i, j) матрицы AWA^t.

6.17. Матрица полустепеней исхода $K=[k_{ij}]$ ориентированного графа G без петель определяется следующим образом:

$$k_{ij}=\begin{cases} -p, & \text{если } i\neq j \text{ и существует } p \text{ дуг, идущих} \\ & \text{из вершины } v_i \text{ в вершину } v_j; \\ d^+(v_i), & \text{если } i=j. \end{cases}$$

Определим входящее дерево как ориентированный граф, имеющий вершину v_r, в которую существует ориентированный путь из любой другой вершины, а лежащий в основе неориентированный граф является деревом. Покажите, что число входящих остовов графа G определяется аналогично теореме 6.24.

6.18. Матрица полустепеней исхода $K=[k_{ij}]$ взвешенного ориентированного графа без петель определяется следующим образом:

а) $k_{ij}=-$(Сумма весов дуг, направленных от вершины v_i к v_j), если $i\neq j$.

б) $k_{ii}=$(Сумма весов дуг, исходящих из вершины v_i). Докажите, что $\det(K_{ii})$ равен сумме весовых произведений всех входящих остовных деревьев графа G с корневой вершиной v_i. *Примечание.* Под корнем здесь понимается такая вер-

шина v_i, что из каждой другой вершины существует ориентированный путь в нее.

6.19. Решите систему уравнений:

$$\begin{bmatrix} 3 & -2 & 1 \\ -1 & 2 & 0 \\ 3 & -2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ -2 \end{bmatrix} x_4. \qquad (6.40)$$

6.20. Пусть G_m — граф Мэзона, связанный с системой линейных уравнений. Покажите, что а) можно удалить петлю, инцидентную вершине x_k с весом $a_{kk} \neq 1$, просто умножив веса всех дуг, инцидентных вершине x_k, на коэффициент $1/(1-a_{kk})$ и б) можно удалить вершину x_p без петли, добавив (a_{pi}, a_{kp}) к весу для всех $i \neq p$ и $k \neq p$ дуги (x_i, x_j).

6.21. Используя метод, предложенный в упражнении 6.20, приведите граф Мэзона, связанный с выражением (6.40), к графу, имеющему лишь вершины x_4 и x_2, и решите систему относительно x_2/x_4.

7. Планарность и двойственность

В этой главе мы обсудим два важных понятия теории графов: планарность и двойственность. Сначала мы рассмотрим планарные графы и установим их некоторые свойства. Будут обсуждаться также характеризации планарных графов, данные Куратовским, Вагнером, Харари и Таттом, Мак-Лейном. Затем мы обсудим определение Уитни двойственных графов, которое дано, исходя из циклов и разрезающих множеств, и свяжем это понятие с, казалось бы, не имеющим к нему отношения понятием планарности.

Двойственность очень интересна для специалистов в области электрических цепей. Этот интерес обусловлен тем, что напряжение и ток в электрической цепи являются двойственными переменными. Их двойственность следует из законов Кирхгофа. Закон Кирхгофа для напряжений дается, исходя из циклов, а закон Кирхгофа токов — исходя из разрезающих множеств. Двойственность циклов и разрезающих множеств обсуждалась в гл. 2 и 4. Она станет очевидной в гл. 10, в которой мы покажем, что множества циклов и разрезающих множеств имеют одинаковую алгебраическую структуру.

7.1. Планарные графы

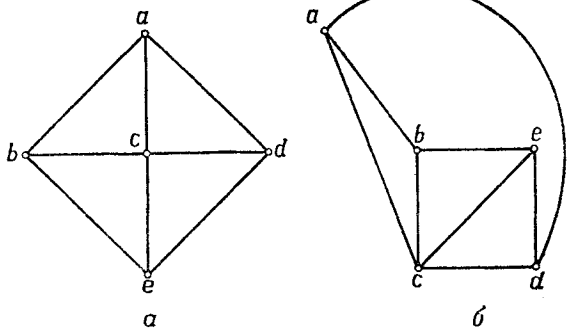
Говорят, что граф G *укладывается* на поверхности S, если его можно нарисовать на этой поверхности таким образом, что его ребра пересекаются только в своих концевых вершинах. Граф называется *планарным*, если его можно уложить на плоскости. Описанное изображение планарного графа G называется *планарной укладкой* графа G.

Две планарные укладки графа представлены на рис. 7.1. В одной из них (рис. 7.1, *а*) все ребра изображены в виде отрезков прямых линий, тогда как в другой (рис. 7.1, *б*) одно из ребер изображено в виде кривой линии. Заметим, что ребро, соединяющее вершины a и d на рис. 7.1, *б*, нельзя изобразить в виде прямой линии, если все остальные ребра изображены так, как показано.

Очевидно, что если граф имеет петли или параллельные ребра, то ни в какой из его планарных укладок нельзя изобразить все ребра в виде отрезков прямых линий. Здесь, естественно, возникает вопрос: для каждого ли планарного графа G существует укладка, в которой все ребра могут быть изображены в виде отрезков прямых? Как

устанавливается в следующей теореме, ответ на этот вопрос — положительный.

Теорема 7.1. Для каждого простого планарного графа существует планарная укладка, в которой все ребра графа можно изобразить в виде отрезков прямых линий.



а *б*

Рис. 7.1. Две планарные укладки графа.

Этот результат получен независимо авторами работ [7.1—7.3].

Если граф не укладывается на плоскости, то его можно уложить на некоторой другой поверхности. Покажем, что укладываемость графа на плоскости и на сфере эквивалентны, т. е. если граф укладывается на плоскости, то он укладывается на сфере, и наоборот. В доказательстве этого утверждения используется понятие так называемой стереографической проекции сферы на плоскость, описываемое ниже.

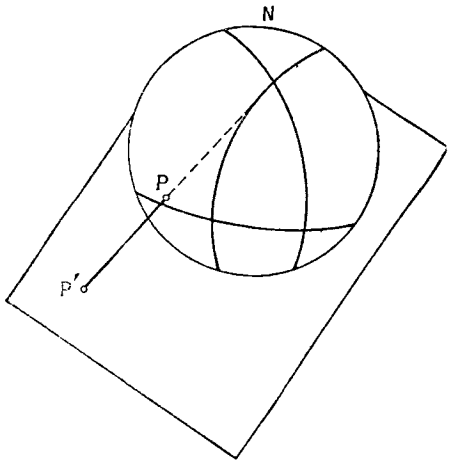


Рис. 7.2. Стереографическая проекция.

Допустим, что мы положили сферу на плоскость (рис. 7.2). Назовем точку соприкосновения южным полюсом, а диаметрально противоположную точку на сфере — северным полюсом N. Пусть P — произвольная точка на сфере. Тогда точка P', в которой прямая, соединяющая точки N и P пересекает плоскость, называется *стереографической проекцией* точки P на плоскости. Очевидно, что между точками на сфере и их стереографическими проекциями на плоскости существует взаимно-однозначное соответствие.

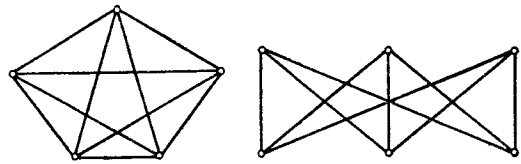
Теорема 7.2. Граф G укладывается на плоскости тогда и только тогда, когда он укладывается на сфере.

Доказательство. Пусть G' — укладка графа G на сфере. Положим сферу на плоскость так, чтобы северный полюс не был ни вершиной, ни точкой на ребре укладки G'.

Тогда образ G' в стереографической проекции — это укладка графа G на плоскости, поскольку ребра укладки G' пересекаются только в своих концевых вершинах, а между точками на сфере, и их образами в стереографической проекции существует взаимно-однозначное соответствие.

Обратное доказывается аналогично.

Два основных непланарных графа, называемые графами Куратовского, представлены на рис. 7.3. Один из них K_5 — полный граф на пяти вершинах, а другой — $K_{3,3}$. Называем эти графы основными



а *б*

Рис. 7.3. Основные непланарные графы.
$а$ — K_5; $б$ — $K_{3,3}$.

непланарными графами потому, что они играют основополагающую роль в характеризации планарности, данной Куратовским (разд.7.3). Непланарность этих графов устанавливается в следующем разделе.

До того как мы закончим этот раздел, хотелось бы указать, что в работе [7.4] доказано, что разделимый граф планарен тогда и только тогда, когда планарны его блоки. Поэтому, касаясь вопросов, связанных с укладкой на плоскости, будет достаточно, если мы рассмотрим только неразделимые графы.

7.2. Формула Эйлера

Укладка планарного графа на плоскости делит ее на *области*. Область называется конечной, если ее площадь конечна, в противном случае — бесконечной.

Например, в планарном графе, представленном на рис. 7.4, заштрихованная область f_5 бесконечна; f_1, f_2, f_3 и f_4 — конечные области.

Очевидно, что ребра на границе области содержат точно один цикл, который называется замыкающим область. Например, ребра e_1, e_8, e_9, e_{10} и e_{13} образуют область f_1 в графе на рис. 7.4 и содержат цикл $\{e_1, e_8, e_9, e_{10}\}$.

Заметим, что всякая область в любой сферической укладке планарного графа конечна.

Пусть планарный граф уложен на сфере, которая поставлена на плоскость так, что северный полюс находится внутри какой-либо выбранной области, например f. Тогда образ f в стереографической проекции будет бесконечной областью. Следовательно, планарный граф всегда можно уложить на плоскости таким образом, чтобы любая выбранная область стала бесконечной областью.

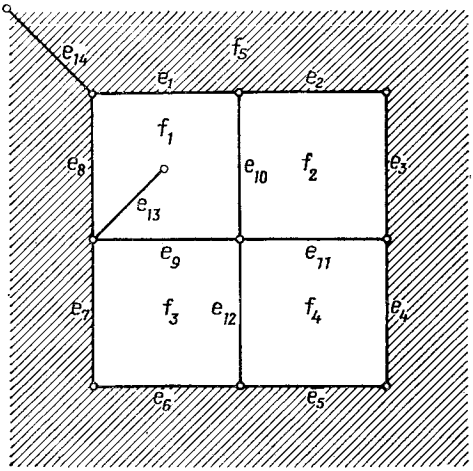


Рис. 7.4.

Пусть $f_1, f_2, \ldots, f_r$ — области планарного графа, f_r — бесконечная область. Обозначим через C_i $(1 \leqslant i \leqslant r)$ цикл на границе области f_i.

Циклы $C_1, C_2, \ldots, C_{r-1}$, соответствующие конечным областям, называются *ячейками*.

Легко убедиться в том, что сумма по mod 2 любых $k \geqslant 2$ ячеек, например $C_1, C_2, \ldots, C_k$, есть цикл или объединение реберно-непересекающихся циклов, замыкающих области $f_1, f_2, \ldots, k_k$. Так как каждая ячейка замыкает только одну область, никакую из них нельзя получить в результате суммы по mod 2 некоторых других ячеек. Таким образом, ячейки $C_1, C_2 \ldots, C_{r-1}$ — линейно-независимы.

Пусть некоторый элемент C подпространства циклов G замыкает конечные области $f_1, f_2, \ldots, f_k$. Тогда можно доказать, что $C = C_1 \oplus C_2 \oplus \ldots, + C_k$. Например, в графе на рис. 7.4 множество $C = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$ замыкает области f_1, f_2, f_3, f_4. Поэтому $C = C_1 \oplus C_2 \oplus C_3 \oplus C_4$.

Таким образом, всякий элемент подпространства циклов можно выразить в виде суммы по mod 2 нескольких или всех ячеек графа G. Поскольку все ячейки взаимно-независимы, приходим к следующей теореме:

Теорема 7.3. Ячейки планарного графа G образуют базис подпространства циклов графа G.

Из этой теоремы непосредственно вытекает следующее следствие:

Следствие 7.3.1 (формула Эйлера). Если связный планарный граф G имеет m ребер, n вершин и r областей, то $n - m + r = 2$.

Доказательство. Доказательство следует из теоремы 7.3, если заметить, что цикломатическое число μ графа G равно $r - 1$.

В общем случае трудно проверить, является граф планарным или нет. Используем формулу Эйлера для получения некоторых свойств планарных графов.

Эти свойства могут быть полезны, как мы вскоре увидим, для обнаружения в определенных случаях непланарности графов.

Следствие 7.3.2. Если связный простой планарный граф G имеет m ребер и $n \geqslant 3$ вершин, то $m \leqslant 3n-6$.

Доказательство. Пусть $F=\{f_1, f_2, \ldots, f_r\}$ — множество областей графа G.

Пусть *степень* $d(f_i)$ *области* f_i означает число ребер на границе f_i, мосты [1] считаются дважды. (Например, в графе на рис. 7.4 степень области f_1 равна 6.) Заметив аналогию между определениями степени вершины и степени области, получим из теоремы 1.1 $\sum\limits_{f_i \in F} d(f_i)=2m$.

Так как граф G не содержит ни параллельных ребер, ни петель и $n \geqslant 3$, то $d(f_i) \geqslant 3$ для всех i. Следовательно, $\sum\limits_{f_i \in F} d(f_i) \geqslant 3r$. Таким образом, $2m \geqslant 3r$, т.е. $r \leqslant {}^2/_3 m$. Используя это неравенство в формуле Эйлера, получим $n-m+{}^2/_3 m \geqslant 2$, или $m \leqslant 3n-6$.

Следствие 7.3.3. Граф K_5 непланарен.

Доказательство. Для K_5 $n=5$, $m=10$. Если бы он был планарным, то по следствию 7.3.2 $m=10 \leqslant 3n-6=9$; получаем противоречие. Таким образом, граф K_5 должен быть непланарным.

Следствие 7.3.4. Граф $K_{3,3}$ непланарен.

Доказательство. Для $K_{3,3}$ $m=9$, $n=6$. Если бы он был планарным, то по формуле Эйлера он имел бы $r=9-6+2=5$ областей.

В графе $K_{3,3}$ нет циклов длины менее 4. Следовательно, степень каждой области по крайней мере равна 4. Таким образом, $2m=\sum\limits_{i=1}^{r} d(f_i) \geqslant 4r$, или $r \leqslant {}^2/_4 m$, т. е. $r \leqslant 4$; получим противоречие. Следовательно, граф $K_{3,3}$ непланарен.

Следствие 7.3.5. В простом планарном графе G имеется по крайней мере одна вершина степени не более 5.

Доказательство. Пусть G имеет m ребер и n вершин. Если каждая вершина имеет степень, превышающую 5, то по теореме 1.1 $2m \geqslant 6n$, или $m \geqslant 3n$. Но, согласно следствию 7.3.2, $m \leqslant 3n-6$. Эти неравенства противоречат друг другу. Следовательно, утверждение следствия верно.

7.3. Теорема Куратовского и другие характеризации планарности

В этом разделе представлены характеризации планарных графов, данные Куратовским, Вагнером, Харари и Таттом, Маклейном.

Для обсуждения характеризации Куратовского необходимо ввести понятие «гомеоморфизм графов».

Два ребра, инцидентные вершине степени 2, называются *последовательными ребрами*.

Пусть $e_1=(u, v)$ и $e_2=(v, w)$ — последовательные ребра, инцидентные вершине v. Удаление вершины v и замена ребер e_1 и e_2 ребром (u, w) называется *слиянием последовательности* (рис. 7.5, *а*).

Добавление новой вершины v на ребро (u, w) посредством введения ребер (u, v) и (v, w) называется *включением последовательности* (рис. 7.5, *б*).

---

[1] Определение моста см. в упражнении 1.28.

Будем говорить, что два графа *гомеоморфны*, если они изоморфны или могут стать изоморфными в результате повторных включений или слияний последовательностей.

Очевидно, что если граф G планарен, то любой граф, гомеоморфный ему, также планарен, т. е. на планарность графа не влияет включение или слияние последовательностей.

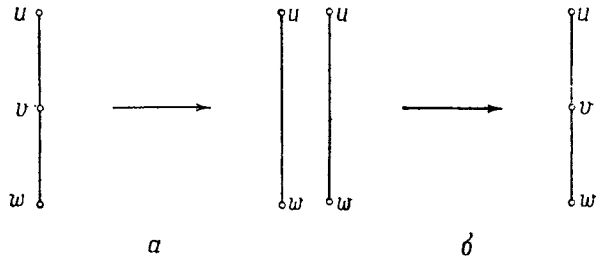


Рис. 7.5.

а — слияние последовательности; *б* — включение последовательности.

В предыдущем разделе мы доказали, что графы K_5 и $K_{3,3}$ непланарны. Следовательно, планарный граф не содержит подграфа, гомеоморфного графу K_5 или $K_{3,3}$. Замечательно то, что Куратовскому [7.5] удалось доказать справедливость обратного. В следующей теореме формулируется эта знаменитая характеризация планарности. Доказательство можно найти в работе [7.6].

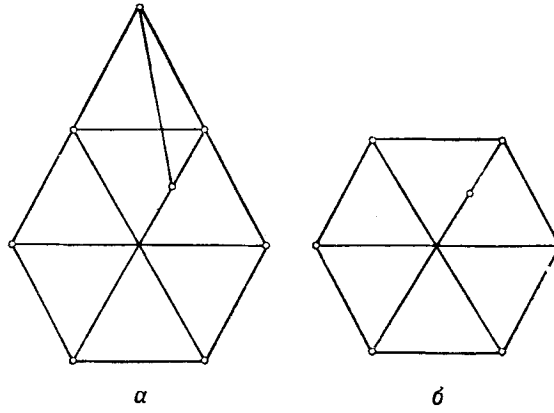


Рис. 7.6.

а — граф G; *б* — подграф графа G, гомеоморфный $K_{3,3}$.

Теорема 7.4 (Куратовского) [1]. Граф планарен тогда и только тогда, когда он не содержит подграфа, гомеоморфного графу K_5 или $K_{3,3}$.

[1] Этот результат был получен ранее (но не опубликован) Л. С. Понтрягиным, в связи с чем теорему, в которой характеризуются планарные графы, называют также теоремой Понтрягина — Куратовского.— *Прим. ред.*

Конструктивное доказательство этой теоремы дал Татт [7.7].

Для иллюстрации теоремы Куратовского рассмотрим граф G, изображенный на рис. 7.6, *а*. Он содержит подграф (рис. 7.6, *б*), который гомеоморфен графу $K_{3,3}$. Следовательно, граф G непланарен.

Представим теперь другую характеризацию планарности, полученную независимо авторами работ [7.8, 7.9].

Теорема 7.5. Граф планарен тогда и только тогда, когда он не содержит подграфа, стягиваемого к графу K_5 или $K_{3,3}$.

Рассмотрим граф (называемый *графом Петерсена*), представленный на рис. 7.7. Этот граф не содержит подграфа, изоморфного графу K_5 или $K_{3,3}$, однако, как известно, он непланарный. Поэтому, если мы хотим использовать критерий Куратовского для установления непланарности графа Петерсена, нам необходимо указать подграф, гомеоморфный графу K_5 или $K_{3,3}$. Однако его непланарность следует очевидным образом из только что приведенной характеризации, поскольку граф приводится к графу K_5 после стягивания ребер e_1, e_2, e_3, e_4 и e_5.

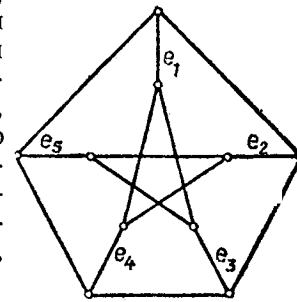


Рис. 7.7. Граф Петерсена.

В следующем утверждении формулируется характеризация планарных графов Маклейна.

Теорема 7.6. Граф G планарен тогда и только тогда, когда в нем существует такое множество базисных циклов, что никакое ребро не присутствует более чем в двух из этих циклов.

Мы знаем, что ячейки планарного графа образуют базис подпространства циклов графа и что никакое ребро графа не присутствует более чем в двух ячейках. Это доказывает необходимость в теореме 7.6. Доказательство достаточности можно найти в работе [7.10].

Другая важная характеризация планарных графов, использующая двойственные графы, обсуждается в разд. 7.5.

7.4. Двойственные графы

Граф G_3 является двойственным к графу G_1, если существует такое взаимно-однозначное соответствие их ребер, что множество ребер графа G_2 — циклический вектор тогда и только тогда, когда соответствующее множество ребер графа G_1 — вектор разрезающего множества G_1. Двойственность была впервые определена Уитни [7.11], хотя его определение было дано в другой форме.

Очевидно, чтобы доказать двойственность графа G_2 к G_1, достаточно показать, что векторы, образующие базис подпространства

циклов графа G_2, соответствуют векторам, образующим базис подпространства разрезов графа G_1.

Рассмотрим, например, графы G_1 и G_2, представленные на рис. 7.8. Ребро e_i графа G_1 соответствует ребру e_i^* графа G_2. Легко видеть, что циклы $\{e_1, e_2, e_3\}$, $\{e_3, e_4, e_5, e_6\}$ и $\{e_6, e_7, e_8\}$ образуют базис подпространства циклов графа G_1, а соответствующие множества ребер

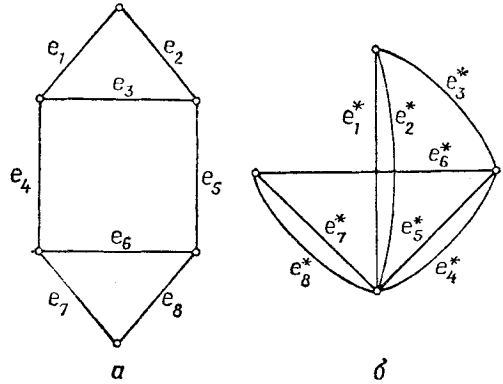


Рис. 7.8. Двойственные графы.
а—граф G_1; *б*—граф G_2

$\{e_1^*, e_2^*, e_3^*\}$, $\{e_3^*, e_4^*, e_5^*, e_6^*\}$, $\{e_6^*, e_7^*, e_8^*\}$ образуют базис подпространства разрезов графа G_2. Таким образом, граф G_2 двойственный к графу G_1.

Изучим теперь некоторые свойства двойственных графов.

Теорема 7.7. Пусть граф G_2 двойственный к графу G_1. Тогда цикл графа G_2 соответствует разрезающему множеству графа G_1 и наоборот.

Доказательство. Пусть C^* — цикл графа G_2, а C — соответствующее множество ребер графа G_1. Предположим, что цикл C не является разрезающим множеством. Тогда из определения двойственности следует, что цикл C должен быть объединением непересекающихся разрезающихся множеств $C_1, C_2, \ldots, C_k$, $k \geqslant 2$.

Пусть $C_1^*, C_2^*, \ldots, C_k^*$ — множества ребер графа G_2, соответствующие разрезающим множествам $C_1, C_2, \ldots, C_k$. Из определения двойственности следует, что $C_1^*, C_2^*, \ldots, C_k^*$ являются циклами или объединениями непересекающихся циклов.

Поскольку C^* — объединение $C_1^*, C_2^*, \ldots, C_k^*$, $k \geqslant 2$, то очевидно, что множество C^* должно содержать более одного цикла. Однако это невозможно, так как C^* — цикл, а никакое собственное подмножество цикла не является циклом. Следовательно, $k=1$, или, другими словами, C — разрезающее множество графа G_1.

Аналогичным образом можно показать, что каждому разрезающему множеству графа G_1 соответствует цикл графа G_2.

Теорема 7.8. Если граф G_2 — двойственный к графу G_1, то последний — двойственный к графу G_2.

Доказательство. Нам необходимо показать, что всякий циклический вектор графа G_1 соответствует вектору разрезающего множества графа G_2 и наоборот.

Пусть C — циклический вектор графа G_1, C^* — соответствующее ему множество ребер графа G_2.

По теореме 4.7 вектор C имеет четное число общих ребер со всяким вектором разрезающего множества графа G_1. Так как граф G_2 — двойственный к графу G_1, то множество C^* имеет четное число общих ребер со всяким циклическим вектором графа G_2. Поэтому по теореме 4.8 C^* — вектор разрезающего множества графа G_2.

Аналогичным образом можно доказать, что всякий вектор разрезающего множества графа G_2 соответствует циклическому вектору графа G_1, а следовательно, и теорему.

Принимая во внимание эту теорему, будем называть графы G_1 и G_2 просто двойственными, если один из них двойственный к другому. Следующий результат вытекает непосредственно из теоремы 7.8 и определения двойственности.

Теорема 7.9. Если G_1 и G_2 — двойственные графы, то ранг одного равен цикломатическому числу другого, т. е. $\rho(G_1)=\mu(G_2)$ и $\rho(G_2)=\mu(G_1)$.

Предположим, граф G имеет двойственный граф. Тогда возникает вопрос: каждый ли его подграф имеет двойственный граф? Для ответа на него нам необходим следующий результат.

Теорема 7.10. Рассмотрим двойственные графы G_1 и G_2. Пусть $e=(v_1, v_2)$ — ребро графа G_1, а $e^*=(v_1^*, v_2^*)$ — соответствующее ребро графа G_2. Пусть G_1' — граф, полученный из графа G_1 удалением ребра e, а G_2' — граф, полученный стягиванием в графе G_2 ребра e^*. Тогда G_1' и G_2' — двойственны, а взаимно-однозначное соответствие между их ребрами остается тем же, что и в графах G_1 и G_2.

Доказательство. Пусть C и C^* — соответствующие множества ребер графов G_1 и G_2. Предположим, что C — цикл графа G_1'. Поскольку множество C не содержит ребра e, то оно является циклом и в графе G_1. Следовательно, C^* — разрезающее множество графа в G_2, например $\langle V_a^*, V_b^*\rangle$. Так как C^* не содержит e^*, вершины v_1^* и v_2^* находятся одновременно либо в V_a^*, либо в V_b^*. Следовательно, C^* — разрезающее множество также и в графе G_2'. Таким образом, каждый цикл графа G_1' соответствует разрезающему множеству графа G_2'.

Пусть C^* — разрезающее множество графа G_2'. Поскольку C^* не содержит e^*, оно является разрезающим множеством в графе G_2. Следовательно, C — цикл графа G_1. Так как C не содержит e, то оно является циклом и в графе G_1'. Таким образом, всякое разрезающее множество графа G_2' соответствует циклу графа G_1'.

На основе этой теоремы можно сказать, используя язык электрических цепей, что «размыкание» ребра в графе G означает «замыкание» соответствующего ребра в графе, двойственном к графу G.

Полезным будет следующее следствие из теоремы 7.10:

Следствие 7.10.1. Если граф G имеет двойственный граф, то всякий его реберно-порожденный подграф также имеет двойственный граф.

Доказательство. Этот результат следует из теоремы 7.10, если мы заметим, что всякий реберно-порожденный подграф H графа G можно получить из графа G удалением ребер, не принадлежащих подграфу H.

Для иллюстрации этого следствия рассмотрим двойственные графы на рис. 7.8. Граф G_1', представленный на рис. 7.9, *а*, получен из графа G_1 удалением ребер e_3 и e_6. Граф G_2' на рис. 7.9, *б* получен стягиванием ребер e_3^* и e_6^* графа G_2. Легко видеть, что G_1' и G_2' — двойственные графы.

Замечая, что последовательные ребра графа G соответствуют параллельным ребрам графа, двойственного к графу G, получаем еще одно следствие теоремы 7.10.

Следствие 7.10.2. Если граф G имеет двойственный граф, тогда любой граф, гомеоморфный графу G, также имеет двойственный граф.

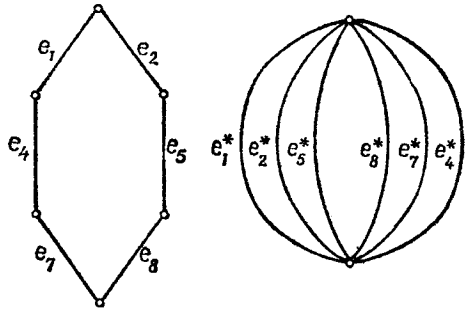
Продолжим наше продвижение по пути к характеризации двойственных графов.

Пусть G — граф на n вершинах. Не нарушая общности, допустим, что он связный. Пусть K^* — подграф графа G, а G' — граф, полученный стягиванием ребер подграфа K^*. Заметим, что G' — также связный граф.

Если K^* имеет n^* вершин и p компонент связности, то G' будет иметь $n-(n^*-p)$ вершин. Следовательно, ранг графа G' определяется следующим образом:

$$\rho(G')=n-(n^*-p)-1=\rho(G)-\rho(K^*). \tag{7.1}$$

Теорема 7.11. Пусть G_1 и G_2 — графы с взаимно-однозначным соответствием между ребрами. Пусть H — произвольный подграф графа G_1, H^* — соответствующий подграф графа G_2. Пусть K^* — дополнение H^* в графе G_2. В этом случае G_1 и G_2 двойственны тогда и только тогда, когда



а *б*

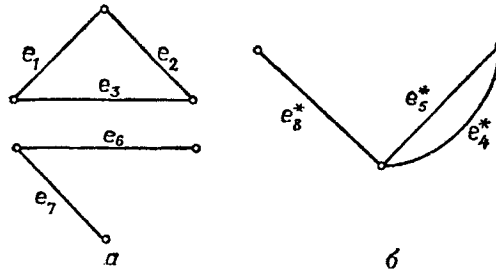
Рис. 7.9. Другой пример двойственных графов
а—граф G_1'; *б*—граф G_2'.

$$\mu(H)=\rho(G_2)-\rho(K^*). \tag{7.2}$$

Доказательство. Необходимость. Пусть G_1 и G_2 — двойственные графы. Граф G_2' получен из графа G_2 стягиванием ребер K^*. Тогда по теореме 7.10 графы H и G_2' являются двойственными. Поэтому по теореме 7.9 $\mu(H)=\rho(G_2')$. Но по формуле (7.1) $\rho(G_2')=\rho(G_2)-\rho(K^*)$. Следовательно, $\mu(H)=\rho(G_2)-\rho(K^*)$.

Достаточность. Допустим, что соотношение (7.2) выполняется для любого подграфа H графа G_1. Покажем, что всякий цикл графа G_1 соответствует разрезающему множеству графа G_2 и наоборот. Пусть H — цикл графа G_1. Тогда $\mu(H)=1$. Поэтому по формуле (7.2) $\rho(K^*)=\rho(G_2)-1$.

Так как H — минимальный подграф графа G с цикломатическим числом, равным 1, а K^* — дополнение подграфа H^* графа G_2, то очевидно, что K^* — максимальный подграф графа G_2, ранг которого равен $\rho(G_2)-1$.



а *б*

Рис. 7.10. Иллюстрация определения двойственности Уитни.
а—граф H, $\mu(H)=1$; *б*—граф K^*, $\rho(K^*)=2$.

мальный подграф графа G_2 с рангом, равным $\rho(G_2)-1$. Из определения разрезающего множества следует, что H^* — разрезающее множество графа G_2.

Аналогичным образом можно показать, что разрезающее множество G_2 соответствует циклу графа G_1.

Таким образом, G_1 и G_2 — двойственные графы.

Первоначальное определение двойственности, данное в работе [7.11], формулировалось так, как в теореме 7.11.

Для иллюстрации этого определения рассмотрим двойственные графы G_1 и G_2 на рис. 7.8. Подграф H графа G_1 и дополнение K^* соответствующего подграфа в графе G_2 приведены на рис. 7.10. Легко видеть, что $\mu(H)=\rho(G_2)-\rho(K^*)$.

7.5. Планарность и двойственность

Теперь мы охарактеризуем класс графов, имеющих двойственные к себе графы. Попутно свяжем два, казалось бы, несвязанных понятия: планарность и двойственность.

Сначала мы докажем, что каждый планарный граф имеет двойственный. Доказательство основано на процедуре построения двойственного графа к данному планарному графу.

Рассмотрим планарный граф. Пусть G будет его планарной укладкой, тогда $f_1, f_2, \ldots, f_r$ — области укладки G. Построим граф G^*, определяемый следующим образом:

1. G^* имеет r вершин $v_1^*, v_2^*, \ldots, v_r^*$; вершина v_i^* $(1\leqslant i\leqslant r)$ соответствует области f_i.
2. G^* имеет столько же ребер, сколько и G.
3. Если ребро e графа G является общим для областей f_i и f_j (не обязательно различных), то соответствующее ребро e^* графа G^* соединяет вершины v_i^* и v_j^*. (Заметим, что каждое ребро e графа G является общим, самое большее, для двух областей; возможно, что ребро лежит точно в одной области.)

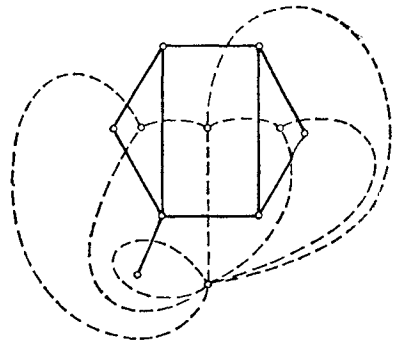


Рис. 7.11. Построение двойственного графа.

Простой способ построения G^* заключается, во-первых, в размещении вершин $v_1^*, v_2^*, \ldots, v_r^*$ в каждой области графа G. Затем для каждого ребра e, общего для областей f_i и f_j, проводим линию, соединяющую вершины v_i^* и v_j^*, так чтобы она пересекала ребро e Эта линия представляет ребро e^*.

Процедура построения G^* иллюстрируется на рис. 7.11. Сплошные линии представляют собой ребра данного планарного графа G, а штриховые — ребра графа G^*.

Докажем, что граф G^* двойственный к графу G.

Пусть $C_1, C_2, \ldots, C_{r-1}$ — ячейки графа G, $C_1^*, C_2^*, \ldots, C_{r-1}^*$ — соответствующие множества ребер G^*. Из процедуры, используемой

для построения графа G^*, следует, что ребра в C_i^* инцидентны вершине v_i^* и образуют разрез, удаление которого отделит вершину v_i^* от остальных вершин графа G^*.

Согласно теореме 7.3, $C_1, C_2, \ldots, C_{r-1}$ образуют базис подпространства циклов графа G, а векторы инциденций $C_1^*, C_2^*, \ldots, C_{r-1}^*$ образуют базис подпространства разрезов G^*. Поскольку между множествами C_i и C_i^* имеется взаимно-однозначное соответствие, графы G и G^* двойственны. Таким образом, справедлива следующая теорема:

Теорема 7.12. Всякий планарный граф имеет двойственный к себе граф.

Сразу же возникает вопрос: имеет ли непланарный граф двойственный к себе граф? Ответ на него отрицательный, и он основан на следующих двух леммах.

Лемма 7.1. Граф $K_{3,3}$ не имеет двойственного графа.

Доказательство. Сначала заметим, что

1. Граф $K_{3,3}$ не имеет разрезающих множеств из двух ребер.
2. Граф $K_{3,3}$ имеет циклы длиной только четыре или шесть.
3. Граф $K_{3,3}$ имеет девять ребер.

Допустим, что граф $K_{3,3}$ имеет двойственный граф G. Тогда из этих замечаний вытекают соответственно следующие замечания для G:

1. Граф G не имеет циклов из двух ребер, т. е. он не имеет параллельных ребер.
2. Граф G не имеет разрезающих множеств с менее чем четырьмя ребрами. Таким образом, каждая вершина графа G имеет степень, не меньшую 4.
3. Граф G имеет девять ребер.

Из первых двух замечаний следует, что граф G имеет по крайней мере 5 вершин, каждая со степенью не менее 4. Таким образом, он должен иметь не менее $^1/_2 \times 5 \times 4 = 10$ ребер. Однако это противоречит замечанию 3. Следовательно, граф $K_{3,3}$ не имеет двойственного графа.

Лемма 7.2. Граф K_5 не имеет двойственного графа.

Доказательство. Заметим сначала, что

1. Граф K_5 не имеет циклов длиной один или два.
2. Граф K_5 имеет разрезающие множества только из четырех или шести ребер.
3. Граф K_5 имеет десять ребер.

Допустим, граф K_5 имеет двойственный граф G. Тогда, согласно замечанию 2, граф G имеет циклы длиной только четыре или шесть. Другими словами, все циклы графа G имеют четную длину. Таким образом, G — двудольный граф. Поскольку двудольный граф с шестью или менее вершинами не может иметь больше девяти ребер, необходимо, чтобы граф G имел по крайней мере семь вершин. Однако, согласно замечанию 1, степень каждой вершины графа G не менее 3. Следовательно, граф G должен иметь по крайней мере $^1/_2 \times 7 \times 3 > 10$ ребер. Однако это противоречит замечанию 3. Следовательно, граф K_5 не имеет двойственного графа.

Основной результат этого раздела заключается в следующем:

Теорема 7.13. Граф имеет двойственный граф тогда и только тогда, когда он планарный.

Доказательство.

Достаточность устанавливается теоремой 7.12.

Необходимость можно доказать, показав, что непланарный граф G не имеет двойственного. По теореме Куратовского граф G содержит подграф H, гомеоморфный графу $K_{3,3}$ или K_5. Если бы граф G имел двойственный граф, тогда, согласно следствию 7.10.1, и подграф H имел бы двойственный граф. Однако тогда по следствию 7.10.2 граф $K_{3,3}$ или K_5 должен был иметь двойственный. Однако это противоречит тому, что ни один из них не имеет двойственного графа. Следовательно, граф G не имеет двойственного графа.

Эта теорема характеризует планарные графы, исходя из существования

двойственных графов; впервые она была доказана Уитни. Используемое здесь доказательство предложил Парсонс [7.12]. Первоначальное доказательство Уитни, в котором не используется теорема Куратовского, можно найти в работе [7.13].

Очевидно, процедура, рассмотренная ранее в этом разделе, может для различных (хотя и изоморфных) планарных укладок графа привести к неизоморфным двойственным графам (упражнение 7.6). В следующей теореме раскрывается связь между двойственными графами к данному графу.

Теорема 7.14. Все двойственные графы графа G — 2-изоморфны; любой граф, 2-изоморфный двойственному графу G, также двойственный к графу G.

Доказательство этой теоремы непосредственно вытекает из определения двойственного графа и теоремы 1.7.

7.6. Замечания, касающиеся литературы

Для дальнейшего изучения рекомендуются статьи [7.4, 7.11, 7.14] и книги [7.6, 7.13, 7.15, 7.16]. Алгоритм проверки планарности графа можно найти в работе [7.17].

Представляет интерес следующие две характеристики непланарного графа:

1. Минимальное число планарных подграфов, объединение которых дает граф G, называется *толщиной* графа G.
2. Минимальное число скрещиваний (или пересечений) ребер в изображении на плоскости графа называется *числом скрещиваний* графа G.

Некоторые результаты, касающиеся толщины и числа скрещиваний непланарных графов, можно найти в работах [7.6, 7.18].

Упражнения

7.1. Покажите, что если G — связный планарный граф с m ребрами, n вершинами и обхватом [1] $k \geqslant 3$, то $m \leqslant k(n-2)/(k-2)$. Используя этот результат, докажите, что граф Петерсена непланарный.

7.2. Докажите, что простой планарный граф с $n \geqslant 4$ вершинами имеет по крайней мере четыре вершины степени 5 или меньше.

7.3. Докажите или опровергните: любой связный простой непланарный граф стягивается к K_5 или $K_{3,3}$.

7.4. Пусть G — простой граф, имеющий по крайней мере одиннадцать вершин. Покажите, что G и его дополнение $\bar{G}$ не могут быть одновременно планарными. (На самом деле аналогичный результат можно доказать, заменив одиннадцать на девять [7.19].) Приведите пример такого графа G на восьми вершинах, что графы G и $\bar{G}$ одновременно планарны.

7.5. Используя теорему Куратовского, докажите, что граф Петерсена непланарный.

7.6. Найдите два неизоморфных графа, двойственных к графу на рис. 7.12.

7.7. Докажите, что планарный граф без петель неразделим тогда и только тогда, когда неразделим двойственный к нему граф.

7.8. Покажите, что двойственный граф к неразделимому планарному графу эйлеров тогда и только тогда, когда граф двудольный.

---

[1] Определение обхвата см. в упражнении 1.18.

7.9. Планарный граф является *самодвойственным*, если он изоморфен двойственному к нему графу. Покажите, что если граф G на n вершинах и m ребрах самодвойственный, то $m=2n-2$.

7.10. Графом *с одной терминальной парой* называется граф, две вершины которого помечены как терминальные. *Планарным графом с одной терминальной парой* называется граф с одной терминальной парой, являющийся планарным и остающийся им после введения ребра, которое соединяет терминальные вершины.

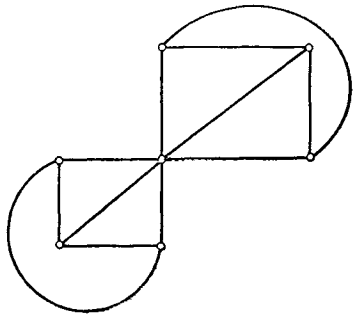


Рис. 7.12.

Параллельно-последовательным графом является граф с одной терминальной парой, определяемый рекурсивно:

а) отдельное ребро (с концевыми вершинами) есть параллельно-последовательный граф. Если G' и G'' — параллельно-последовательные графы, то

б) последовательная комбинация графов G' и G'' есть параллельно-последовательный граф. Под последовательной комбинацией графов G' и G'' мы понимаем соединение одной из терминальных вершин графа G' с одной из терминальных вершин графа G'' (рис. 7.13, *а*).

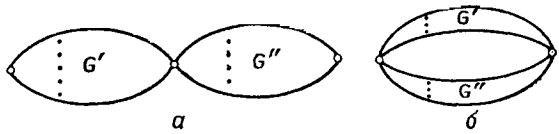


Рис. 7.13.

а — последовательная комбинация графов G' и G''; *б* — параллельная комбинация графов G' и G''.

в) Параллельная комбинация графов G' и G'' есть параллельно-последовательный граф. Под параллельной комбинацией этих графов мы понимаем соединение двух терминальных вершин графа G' с двумя терминальными вершинами графа G'' (рис. 7.13, *б*). Покажите, что граф, двойственный к графу G, является параллельно-последовательным тогда и только тогда, когда граф G — параллельно-последовательный.

8. Связность и паросочетания

В гл. 1 мы определили, что граф является связным, если между двумя произвольными вершинами существует путь. Предположим, что граф G связный. Тогда нас может интересовать, «как хорошо» он связан. Другими словами, нам бы хотелось знать минимальное число вершин или ребер, удаление которых превращало бы граф G в несвязный. Это приводит нас к понятиям «вершинная связность» и реберная связность графа». В настоящей главе мы предлагаем некоторые результаты, касающиеся вершинной и реберной связностей графа. Обсуждаем также классический результат теории графов — теорему Менгера, которая соотносит связности с числом вершинно-непересекающихся и реберно-непересекающихся путей.

Паросочетание графа — это множество ребер, никакие два из которых не имеют общих вершин. В заключительной части этой главы мы развиваем теорию паросочетаний, начиная ее изучение с теоремы Холла о свадьбах.

Связность и паросочетания — это широко изученные темы теории графов. Этим областям принадлежит много глубоких результатов теории графов.

8.1. Связность, или вершинная связность

Связностью $\varkappa(G)$ графа G называется минимальное число вершин, удаление которых из G приводит к несвязному или тривиальному графу [1]. Величина $\varkappa(G)$ называется также *вершинной связностью*, чтобы отличить ее от реберной связности, которая будет введена в следующем разделе.

Например, связность графа на рис. 8.1 равна 2, поскольку удаление вершин v_1 и v_2 из него приводит к несвязному графу, а удаление произвольной вершины не достигает этого. Очевидно, что связность несвязного графа равна 0.

Рассмотрим граф на n вершинах. Ясно, что $\varkappa(G)=n-1$, если граф G полный. Если же он не полный, то имеет хотя бы две несмежные вершины v_1 и v_2. Удаление из графа G оставшихся $n-2$ вершин

---

[1] Напоминаем, что тривиальным называется граф, имеющий только одну вершину (разд. 1.7).

приведет к графу, в котором вершины v_1 и v_2 не соединены путем. Таким образом, если граф G не полный, то $\varkappa(G) \leqslant n-2$.

Разделяющее множество графа G — это множество вершин, удаление которых из графа G приводит к несвязному или тривиальному графу.

Граф называется *k-связным*, если $\varkappa(G) \geqslant k$. Таким образом, k-связный граф не содержит разделяющих множеств S мощности $|S| \leqslant k-1$.

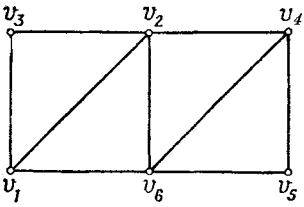


Рис. 8.1.

Если граф связный, его связность больше или равна 1. Следовательно, связные графы 1-связны.

Если связный граф не содержит точек сочленения, то его связность больше 1. Поэтому связные графы без точек сочленения 2-связны.

В следующей теореме мы представляем простую верхнюю границу связности графа.

Теорема 8.1. $\varkappa(G) \leqslant \delta(G)$ для простого связного графа G, где $\delta(G)$ — минимальная степень вершины в графе G.

Доказательство. Рассмотрим произвольную вершину v простого связного графа $G=(V, E)$. Пусть $\Gamma(v)$ — множество смежных с v вершин. Ясно, что $\Gamma(v)$ — разделяющее множество, поскольку удаление его вершин приведет к тривиальному или несвязному графу, в котором вершина v не соединена ни с одной из оставшихся вершин. Поэтому $\varkappa(G) \leqslant |\Gamma(v)|$ для всех $v \in V$. Поскольку G простой граф, $|\Gamma(v)| = d(v)$ и, следовательно, $\varkappa(G) \leqslant \min\limits_{v \in V} \{d(v)\} = \delta(G)$.

Если граф G имеет m ребер и n вершин $v_1, v_2, \ldots, v_n$, то из теоремы 1.1 следует, что $d(v_1)+d(v_2)+\ldots+d(v_n)=2m$. Поэтому $n \cdot \delta(G) \leqslant 2m$ и

$$\delta(G) \leqslant \left\lfloor \frac{2m}{n} \right\rfloor, \tag{8.1}$$

где $\lfloor x \rfloor$ — целая часть x (наибольшее целое, меньшее или равное x).

Объединяя выражение (8.1) с результатом, установленным теоремой 8.1, получаем следующее утверждение:

Теорема 8.2. Для простого связного графа G, имеющего m ребер и n вершин, $\varkappa(G) \leqslant \lfloor 2m/n \rfloor$.

Пусть $f(k, n)$ — наименьшее число ребер, которые должен иметь k-связный граф на n вершинах. Конечно, мы полагаем $k<n$. Из теоремы 8.2 следует, что

$$f(k, n) \geqslant \left\lceil \frac{kn}{n} \right\rceil, \tag{8.2}$$

где $\lceil x \rceil$ — ближайшее целое к x (наименьшее целое, большее или равное x).

Харари [8.1] доказал, что равенство в выражении (8.2) достигается с помощью специальной процедуры построения k-связного графа $H_{k, n}$, который содержит точно $\lceil kn/2 \rceil$ ребер. Процедура заключается в следующем:

Случай 1. k четно.

Пусть $k=2r$. Тогда $H_{2r,\,n}$ строится на вершинах $v_0, v_1, \ldots, v_{n-1}$ и две вершины v_i и v_j смежны, если $i-r \leqslant j \leqslant i+r$, где сложение ведется по mod n. Граф $H_{6,\,8}$ представлен на рис. 8.2, *а*.

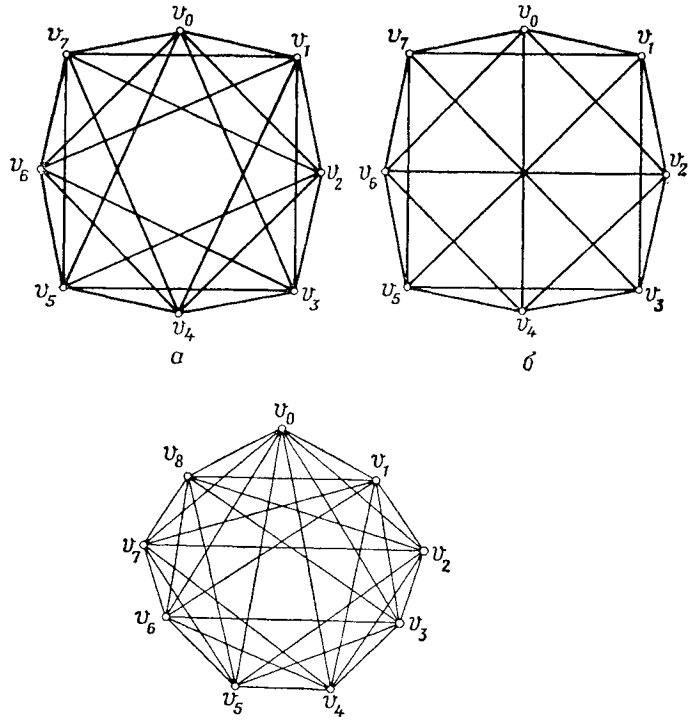


Рис. 8.2.

а — $H_{6,\,8}$; *б* — $H_{5,\,8}$; *в* — $H_{7,\,9}$.

Случай 2. k нечетно, n четно.

Пусть $k=2r+1$. Граф $H_{2r+1,\,n}$ получается из графа $H_{2r,\,n}$ введением ребер, соединяющих вершины v_i и v_j для $i=1, 2, \ldots, n/2$ и $j=i+n/2 \pmod n$. Граф $H_{5,\,8}$ показан на рис. 8.2, *б*.

Случай 3. k нечетно, n нечетно.

Пусть $k=2r+1$. Граф $H_{2r+1,\,n}$ получается из графа $H_{2r,\,n}$ введением ребер, соединяющих вершины v_i и v_j для $i=0, 1, 2, \ldots, (n-2)/2$ и $j=i+(n+1)/2 \pmod n$. Граф $H_{7,\,9}$ представлен на рис. 8.2, *в*.

Легко проверить, что граф $H_{k,\,n}$, построенный указанным образом, содержит точно $\lceil kn/2 \rceil$ ребер. Докажем сейчас, что граф $H_{k,\,n}$ k-связный.

Теорема 8.3. Граф $H_{k,n}$ k-связный.

Доказательство.

Случай 1. $k=2r$.

Поскольку граф $H_{2r,n}$ симметричен, достаточно показать, что вершины v_0 и v_α, $\alpha=1, 2, \ldots, n-1$ можно соединить путем после удаления менее чем $2r$ вершин. Допустим, что вершины v_0 и v_α нельзя связать путем после удаления $2r-1$ вершин $v_{i_1}, v_{i_2}, \ldots, v_{i_{2r-1}}$. Один из двух интервалов $[0, \alpha]$, $[\alpha, n]$ содержит, самое большее, $r-1$ этих индексов. Предположим, что это интервал $[0, \alpha]$. Тогда две последовательные вершины из последовательности, полученной после удаления вершин $v_{i_1}, v_{i_2}, \ldots, v_{i_{2r-1}}$ из последовательности вершин $v_0, v_1, v_2, \ldots, v_\alpha$, соединены ребром (поскольку разность между их индексами меньше или равна r). Следовательно, существует путь из вершины v_0 в вершину v_α, что противоречит допущению.

Случай 2. $k=2r+1$, n четно.

Допустим, что вершины v_0 и v_α нельзя связать путем после удаления $2r$ вершин $v_{i_1}, v_{i_2}, \ldots, v_{i_{2r}}$. Если один из двух интервалов $[0, \alpha]$, $[\alpha, n]$ не содержит последовательных индексов из этого множества, то путь из вершины v_0 в вершину v_α можно построить так, как показано в случае 1. Поэтому предположим, что вершины v_0 и v_α нельзя соединить путем после удаления вершин $v_i, v_{i+1}, \ldots, v_{i+r-1}$, где $1 \leqslant i \leqslant \alpha - r$ и $v_{\alpha+j}, v_{\alpha+j+1}, \ldots, v_{\alpha+j+r-1}$, где $1 \leqslant j \leqslant n-r-\alpha$.

Пусть

$$\beta = \left\lfloor \frac{\alpha+i+j+r+n-1}{2} \right\rfloor, \quad \beta' = \beta + \frac{n}{2} \pmod{n}.$$

Тогда $\beta \in [\alpha+j+r, n+i-1]$, $\beta' \in [i+r, \alpha+j-1]$. Существует путь из вершины v_0 в вершину v_β и из вершины $v_{\beta'}$ в вершину v_α. Следовательно, существует путь из вершины v_0 в вершину v_α, поскольку вершины v_β и $v_{\beta'}$ смежны в $H_{2r+1,n}$.

Случай 3. $k=2r+1$, n нечетно.

Доказательство проводится аналогично случаю 2.

В следующей теореме мы представляем достаточные условия того, что граф k-связен. Этот результат получен Бонди [8.2].

Теорема 8.4. Пусть G — простой граф размерности n. Пусть вершины графа G упорядочены так, что $d(v_1) \leqslant d(v_2) \leqslant \ldots \leqslant d(v_n)$. Тогда граф G k-связен, если $d(v_r) \geqslant r+k-1$ для $1 \leqslant r \leqslant n-1-d(v_{n-k+1})$.

Доказательство. Допустим, простой граф G удовлетворяет условиям теоремы. Если он не k-связный, то существует такое разделяющее множество S, что $|S|=s<k$.

Рассмотрим граф $G-S$, который должен быть несвязным. Пусть H — компонента графа $G-S$ минимального порядка h. Тогда степень каждой вершины в графе H не больше $h-1$. Следовательно, в графе G степень каждой вершины графа H не превышает $h+s-1$. Таким образом,

$$d(v_h) \leqslant h+s-1 < h+k-1. \tag{8.3}$$

Поэтому по условиям теоремы

$$h > n-1-d(v_{n-k-1}). \tag{8.4}$$

Так как граф $G-S$ имеет $n-s$ вершин, а H — компонента графа $G-S$ минимального порядка, то $h \leqslant n-s-h$, или $h+s \leqslant n-h$.

Следовательно,

$$d(v) \leqslant h+s-1 \leqslant n-h-1, \quad v \in V(H), \tag{8.5}$$

где $V(H)$ — множество вершин графа H. Так как каждая вершина $u \in V(G)-V(H)-S$ смежна не больше чем с $n-h-1$ вершинами

$$d(u) \leqslant n-h-1, \quad u \in V(G)-V(H)-S. \tag{8.6}$$

Из выражений (8.5) и (8.6) следует, что все вершины степени больше $n-h-1$ находятся в S. Следовательно, существует не более s вершин степени, превышающей $n-h-1$. Поэтому

$$d(v_{n-s}) \leqslant n-h-1. \tag{8.7}$$

Используя выражения (8.4) и (8.7), получим $d(v_{n-s})<d(v_{n-k+1})$. Следовательно, $n-s<n-k+1$ или $s \geqslant k$, что противоречит допущению.

Например, степени вершин графа на рис. 8.3 удовлетворяют условиям теоремы 8.4 для $k=3$. Следовательно, он 3-связный.

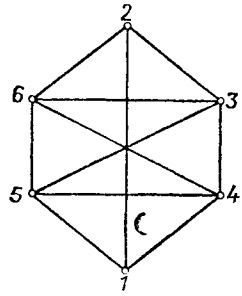


Рис. 8.3. 3-связный граф, удовлетворяющий условиям теоремы 8.4 для случая $k=3$.

8.2. Реберная связность

Реберная связность $\varkappa'(G)$ графа G — это минимальное число ребер, удаление которых из графа приводит к несвязному или тривиальному графу. Другими словами, $\varkappa'(G)$ — число ребер в разрезе с минимальным числом ребер. Например, для графа на рис. 8.4 реберная связность равна 2, поскольку удаление двух ребер e_1 и e_2 делает граф несвязным, а удаление произвольного одного ребра не приводит к несвязному графу.

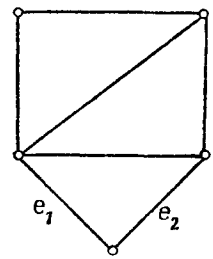


Рис. 8.4. 2-реберно-связный граф.

Граф G называется *k-реберно-связным*, если $\varkappa'(G) \geqslant k$. Таким образом, чтобы сделать несвязным k-реберно-связный граф, необходимо удалить хотя бы k ребер.

Поскольку ребра, инцидентные любой вершине v графа G, образуют разрез, то $\varkappa'(G) \leqslant \delta(G)$.

В следующей теореме мы связываем величины $\varkappa(G)$, $\varkappa'(G)$ и $\delta(G)$.

Теорема 8.5. Для простого графа $\varkappa(G) \leqslant \varkappa'(G) \leqslant \delta(G)$.

Доказательство. Второе неравенство мы уже доказали. Первое неравенство можно доказать следующим образом:

Если G — несвязный граф, то $\varkappa(G)=\varkappa'(G)=0$. В этом случае условие $\varkappa(G) \leqslant \varkappa'(G)$ выполняется.

Если G — связный граф и $\varkappa'(G)=1$, то граф G имеет мост e.

Если мы удалим в этом случае одну из вершин, с которыми инцидентен мост e, то получится несвязный или тривиальный граф. Следовательно, и в этом случае условие $\varkappa(G) \leqslant \varkappa'(G)$ выполняется.

Предположим, $\varkappa'(G) \geqslant 2$. Тогда в графе G имеется $\varkappa'(G)$ ребер, удаление которых делает его несвязным. Удаление любых $\varkappa'(G)-1$ из этих ребер приводит к графу с мостом $e=(v_1, v_2)$. Для каждого из этих $\varkappa'(G)-1$ ребер выберем концевую вершину, отличную от вершин v_1 и v_2. Удаление этих вершин приведет к удалению из графа G $\varkappa'(G)-1$ ребер и, возможно, еще некоторых. Предположим, что получившийся граф несвязный. Тогда $\varkappa(G) \leqslant \varkappa'(G)-1$. В противном случае граф будет иметь мост e, и поэтому удаление вершины v_1 или v_2 приведет к несвязному или тривиальному графу. В этом случае $\varkappa(G) \leqslant \varkappa'(G)$.

Таким образом, во всех случаях $\varkappa(G) \leqslant \varkappa'(G)$.

Сейчас мы представим достаточные условия того, что $\varkappa'(G)$ равно $\delta(G)$. Этот результат получен Чартрэндом [8.3].

Теорема 8.6. Пусть G — простой граф на n вершинах. Если $\delta(G) \geqslant \lfloor n/2 \rfloor$, то $\varkappa'(G) = \delta(G)$.

Доказательство. Можно показать, что G — связный граф (упражнение 1.14). Поэтому $\varkappa'(G) > 0$. Так как $\varkappa'(G) \leqslant \delta(G)$, теорема будет доказана, если мы покажем, что $\varkappa'(G) \geqslant \delta(G)$.

Допустим, что $\varkappa'(G) < \delta(G)$. Тогда существует такой разрез $S = \langle V_1, \overline{V}_1 \rangle$, что $\varkappa'(G) = |S| < \delta(G)$. Пусть ребра из разреза S инцидентны q вершинам множества V_1 и p вершинам множества $\overline{V}_1$.

Допустим, $|V_1| = q$. Тогда каждая вершина множества V_1 является концевой вершиной по крайней мере одного ребра разреза S. Если обозначить через G_1 порожденный подграф графа G на множестве вершин множества V_1, то G_1 имеет по крайней мере $m_1 = {}^1/_2[q\delta(G) - \varkappa'(G)]$ ребер. Поскольку $\varkappa'(G) < \delta(G)$, то $m_1 > {}^1/_2(q\delta(G) - \delta(G)) = {}^1/_2\delta(G)(q-1) > {}^1/_2 q(q-1)$, так как $\delta(G) > \varkappa'(G) \geqslant q$. Это противоречит тому, что в простом графе не может быть больше чем $q(q-1)/2$ ребер, соединяющих q вершин. Поэтому $|V_1| > q$. Аналогично доказываем, что $|\overline{V}_1| > p$.

Если $|V_1| > q$ и $|\overline{V}_1| > p$, то во множествах V_1 и $\overline{V}_1$ имеются вершины, смежные только с вершинами множеств V_1 и $\overline{V}_1$ соответственно. Следовательно, каждое из множеств V_1 и $\overline{V}_1$ содержит по крайней мере по $\delta(G)+1$ вершин. Таким образом, граф G содержит не менее $2\delta(G)+2$ вершин. Но $2\delta(G)+2 \geqslant 2\lfloor n/2 \rfloor + 2 > n$, что противоречит условию. Следовательно, не существует разреза S с $|S| < \delta(G)$.

8.3. Графы с заданными степенями

Напомним, что последовательность $(d_1, d_2, \ldots, d_n)$ неотрицательных целых чисел называется графической, если существует граф с такими вершинами $v_1, v_2, \ldots, v_n$, что вершина v_i имеет степень d_i.

В этом разделе сначала опишем алгоритм построения простого графа (если он существует), имеющего заданную последовательность степеней. Затем мы используем этот алгоритм для установления теоремы Эдмондса о существовании k-реберно-связных простых графов, имеющих заданные последовательности степеней.

Рассмотрим графическую последовательность $(d_1, d_2, \ldots, d_n)$, упорядоченную по убыванию: $d_1 \geqslant d_2 \geqslant \ldots \geqslant d_n$. Пусть d_i — степень вершины v_i. «Изъять d_k» означает соединить соответствующую вершину v_k с вершинами $v_1, v_2, \ldots, v_{d_k}$, если $d_k < k$, или $v_1, v_2, \ldots, v_{k-1}, v_{k+1}, \ldots, v_{d_k+1}$, если $d_k \geqslant k$. Последовательность $(d_1-1, \ldots, d_{d_k}-1, d_{d_k+1}, \ldots, d_{k-1}, 0, d_{k+1}, \ldots, d_n)$, если $d_k < k$, или $(d_1-1, \ldots, d_{k-1}-1, 0, d_{k+1}-1, \ldots, d_{d_k+1}-1, d_{d_k+2}, \ldots, d_n)$,

если $d_k \geqslant k$, называется *остаточной последовательностью* после изъятия d_k или просто остаточной последовательностью.

Хакими [8.4] и Гавел [8.5] предложили алгоритм построения простого графа (если он существует), имеющего заданную последовательность степеней. Этот алгоритм основан на результате, являющемся частным случаем (при $k=1$) следующей теоремы, доказанной Вонгом и Клейтманом [8.6].

Теорема 8.7. Если последовательность $(d_1, d_2, \ldots, d_1)$, $d_1 \geqslant d_2 \geqslant \ldots \geqslant d_n$, является последовательностью степеней простого графа, то этим свойством обладает и остаточная последовательность после изъятия d_k.

Доказательство. Для доказательства нам необходимо показать, что существует такой граф с последовательностью степеней $(d_1, d_2, \ldots, d_n)$, в котором вершина v_k смежна с первыми d_k вершинами, отличными от самой вершины v_k. Допустим противное. Выберем среди графов с последовательностью степеней $(d_1, d_2, \ldots, d_n)$ простой граф G, в котором вершина v_k смежна с максимальным числом вершин из первых d_k вершин, отличных от v_k. Пусть v_m— такая вершина, не смежная с вершиной v_k в графе G, что $m \leqslant d_k$, если $d_k < k$ или $m \leqslant d_k + 1$, если $k \leqslant d_k$. Другими словами, вершина v_m находится среди первых d_k вершин, не считая саму вершину v_k. Поэтому последняя смежна в графе G с некоторой вершиной v_q, не входящей в число первых d_k вершин. Тогда $d_m > d_k$ (если бы было равенство, m и q можно было поменять местами) и, следовательно, вершина v_m смежна с некоторой такой вершиной v_t, $t \neq q$, $t \neq m$, что вершины v_t и v_q не смежны. Теперь, если заменить ребра (v_m, v_t) и (v_k, v_q) на (v_m, v_k) и (v_t, v_q) соответственно, мы получим граф G', в котором число первых d_k вершин, отличных от вершин v_k и смежных с ней, на 1 больше, что противоречит определению графа G.

Эта теорема предлагает следующий алгоритм, являющийся обобщением алгоритма Хакими реализации последовательности $D = (d_1, d_2, \ldots, d_n)$, $d_1 \geqslant d_2 \geqslant \ldots \geqslant d_n$, простым графом.

Выберем произвольное $d_k \neq 0$. «Изымем» d_k, соединяя вершину v_k с первыми d_k вершинами, не считая саму v_k. Определим остаточную последовательность. Переупорядочим вершины так, чтобы остаточные степени в получившейся последовательности были невозрастающими. Повторяем этот процесс до тех пор, пока не возникнет одна из следующих ситуаций:

1. Все остаточные степени равны нулю. В этом случае получившийся граф имеет последовательность степеней D.
2. Одна из остаточных степеней отрицательна. Это означает, что последовательность D не графическая.

Для иллюстрации описанного алгоритма рассмотрим последовательность

$$\begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix}$$
$$D = (4 \quad 3 \quad \textcircled{3} \quad 2 \quad 2).$$

После изъятия d_3 (обведенной кружком), получим последовательность

$$\begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix}$$
$$D' = (3 \quad 2 \quad 0 \quad 1 \quad 2)$$

которая после переупорядочения остаточных степеней принимает вид

$$\begin{matrix} & v_1 & v_2 & v_5 & v_4 & v_3 \end{matrix}$$
$$D_1 = (3 \quad 2 \quad \textcircled{2} \quad 1 \quad 0).$$

Затем, изымая степень, соответствующую вершине v_5, получим

$$\begin{matrix} & v_1 & v_2 & v_5 & v_4 & v_3 \end{matrix}$$
$$D_1' = (2 \quad 1 \quad 0 \quad 1 \quad 0).$$

Переупорядочивая остаточные степени в D_1', получим

$$\begin{matrix} & v_1 & v_2 & v_4 & v_5 & v_3 \end{matrix}$$
$$D_2 = (\textcircled{2} \quad 1 \quad 1 \quad 0 \quad 0).$$

Теперь, изымая степень, соответствующую вершине v_1, получим

$$\begin{matrix} & v_1 & v_2 & v_4 & v_5 & v_3 \end{matrix}$$
$$D_2' = (0 \quad 0 \quad 0 \quad 0 \quad 0).$$

Здесь алгоритм заканчивает работу. Поскольку все остаточные степени равны нулю, последовательность (4, 3, 3, 2, 2) графическая. Требуемый граф (рис. 8.5) получается в результате выполнения шагов, соответствующих порядку изъятия степеней:

1. Соединяем вершину v_3 с v_1, v_2 и v_4.
2. Соединяем вершину v_5 с v_1 и v_2.
3. Соединяем вершину v_1 с v_2 и v_4.

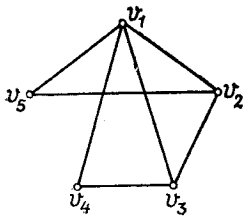


Рис. 8.5. Граф с последовательностью степеней (4, 3, 3, 2, 2).

Эрдёш и Галлаи [8.7] определили необходимые и достаточные условия (неалгоритмического типа) того, что последовательность графическая. Их результат описывается также в работе [8.8].

Предположим, что в описанном алгоритме мы изымаем на каждом шаге наименьшую ненулевую остаточную степень. Тогда с помощью индукции легко показать, что получающийся граф связен, если

$$d_i \geqslant 1 \text{ для всех } i, \tag{8.8}$$

$$\sum_{i=1}^{n} d_i \geqslant 2(n-1). \tag{8.9}$$

Заметим, что выполнение неравенств (8.8) и (8.9) необходимо, чтобы граф был связным. В следующей теореме мы доказываем более

сильный результат. Он получен Эдмондсом [8.9]. Данное здесь доказательство предложили Вонг и Клейтман [8.10].

Теорема 8.8 (Эдмондс). Необходимым и достаточным условием того, что графическая последовательность $(d_1, d_2, \ldots, d_n)$ есть последовательность степеней простого k-реберно-связного графа при $k \geqslant 2$, является выполнение для всех степеней $d_i \geqslant k$.

Доказательство. Необходимость очевидна.

Для доказательства достаточности покажем, что алгоритм, который мы только что описали, приводит, в случае когда все степени в данной графической последовательности больше или равны k, к k-реберно-связному графу. Заметим, что на каждом шаге алгоритма мы должны изымать наименьшую ненулевую остаточную степень.

Доказываем с помощью индукции. Допустим, что алгоритм действует для всех последовательностей, в которых каждая степень $d_i \geqslant p$, $p \leqslant k-1$.

Теперь необходимо показать, что в графе, построенном с помощью алгоритма, любое разрезающее множество $\langle A, \bar{A} \rangle$ содержит по меньшей мере k ребер. Доказательство этого тривиально, если $|A|=1$ или $|\bar{A}|=1$. Допустим, что $|A| \geqslant 2$ и $|\bar{A}| \geqslant 2$.

На некотором шаге алгоритма, например r (шаг, на котором r-я вершина полностью связана), следующие случаи полностью охватывают возможные варианты:

Случай 1. Все ненулевые остаточные степени не меньше k.

Случай 2. Все ненулевые остаточные степени не меньше $k-1$, и существует хотя бы одно ребро, построенное на шагах $1, 2, \ldots, r$ и лежащее в множестве $\langle A, \bar{A} \rangle$.

Случай 3. Все ненулевые остаточные степени не меньше $k-2$, и существует хотя бы два ребра, построенных на шагах $1, 2, \ldots, r$ и лежащих во множестве $\langle A, \bar{A} \rangle$.

Во всех трех случаях по индуктивному предположению разрезающее множество $\langle A, \bar{A} \rangle$ содержит не менее k ребер.

Докажем полноту описанных случаев. Пусть v_i — вершина, соединяемая с другими на шаге i. Не нарушая общности, будем считать, что вершина v_1 находится во множестве A. Покажем, что если случаи 1 и 2 не возникают, то на некотором шаге алгоритма возникнет случай 3.

На шаге 1 вершина v_1 полностью связна. Тогда *а*) наименьшие ненулевые остаточные степени равны $k-1$, поскольку случай 1 не возникал; б) степени ни одной из вершин A не уменьшались на 1 при соединении вершины v_1, поскольку случай 2 не возникал. Следовательно, связываемая на шаге 2 вершина v_2 должна быть в A. (Все вершины A должны иметь степень, не меньшую k.) Если после соединения вершины v_2 не возникает случай 1 и никакое из ребер не связывает A с $\bar{A}$, тогда все еще, как и прежде, мы имеем случай *а* или *б*. Следовательно, следующая связываемая вершина будет снова в A.

Так как остаточные степени на каждом шаге процедуры соединения вершин уменьшаются, то рано или поздно будет связана такая лежащая в A вершина v_r, что между A и $\bar{A}$ появится ребро. Если не возникает случай 2, то одна из ненулевых остаточных степеней вершин, лежащих в A и еще не полностью связанных, станет равной $k-2$. Это означает, что вершина v_r должна соединяться с каждой вершиной $\bar{A}$, поскольку все вершины $\bar{A}$ имеют остаточные степени, равные k, и вследствие того, что мы соединяем вершину v_r с вершинами, имеющими наибольшие остаточные степени. Так как $|\bar{A}| \geqslant 2$, на этом шаге возникает случай 3.

Вонг и Клейтман [8.6] определили также необходимые и достаточные условия, что графическая последовательность является последовательностью степеней простого k-вершинно-связного графа.

8.4. Теорема Менгера

В этом разделе мы представляем классический результат теории графов — теорему Менгера [8.11]. Она поможет соотнести связность графа с числом вершинно-непересекающихся путей между двумя различными вершинами графа.

Теорема 8.9 (Менгер). Минимальное число вершин, удаление которых из графа разделяет две несмежные вершины s и t, равно максимальному числу вершинно-непересекающихся s—t-путей графа.

Доказательство этой теоремы дается в разд. 15.7.

Теорема 8.10. Чтобы простой граф $G=(V, E)$, $|V|\geqslant k+1$, был k-связным, необходимо и достаточно, чтобы между любыми вершинами s и t в графе G проходило k вершинно-непересекающихся s—t-путей.

Доказательство. Очевидно, что теорема верна при $k=1$. Следовательно, необходимо рассмотреть вариант $k\geqslant 2$.

Необходимость. Если s и t не смежны, то необходимость следует из теоремы 8.9.

Предположим, что s и t смежны и что в графе G существует, самое большее, $k-1$ вершинно-непересекающихся s—t-путей. Пусть $e=(s, t)$. Рассмотрим граф $G'=G-e$. Поскольку в графе G было не больше $k-1$ вершинно-непересекающихся s—t-путей, в графе G' их не может быть больше $k-2$. Таким образом, в графе G существует множество $A\subseteq V-\{s, t\}$ вершин, $|A|\leqslant k-2$, удаление которых разделяет s и t. Тогда $|V-A|=|V|-|A|\geqslant k+1-(k-2)=3$ и, следовательно, существует вершина $u\in V-A$, отличная от s и t.

Теперь покажем, что в графе G' существует s—u-путь, не содержащий вершин из A. Ясно, что это верно, если s и u смежны. Если s и u не смежны, тогда в графе G имеется k вершинно-непересекающихся s—u-путей и, следовательно, в графе G' имеется $k-1$ вершинно-непересекающихся s—u-путей. Поскольку $|A|\leqslant k-2$, хотя бы один из этих $k-1$ путей не содержит всех вершин A.

Аналогичным образом можно показать, что в графе G' имеется u—t-путь, который не содержит любую вершину A. Таким образом, в G' существует s—t-путь, не содержащий любую вершину A. Это же противоречит тому, что A — разделяющее вершины s и t множество в графе G'. Следовательно, необходимость доказана.

Достаточность. Поскольку между двумя различными вершинами графа G имеется k вершинно-непересекающихся путей, G — связный граф. Более того, не более чем один из этих путей может иметь длину 1, так как в графе G нет параллельных ребер. Объединение оставшихся $k-1$ путей должно содержать по крайней мере $k-1$ различных вершин, отличных от s и t. Поэтому $|V|\geqslant (k-1)+2>k$.

Допустим, что в графе G имеется такое разделяющее множество A, что $|A|<k$. Рассмотрим тогда подграф G' графа G на множестве вершин $V-A$. Он имеет по меньшей мере две компоненты. Если мы выберем две вершины s и t из различных компонент подграфа G', то в графе G имеется, самое большее, $|A|<k$ вершинно-непересекающихся s—t-путей. Это противоречит тому, что любые две вершины графа G связаны k вершинно-непересекающимися путями.

Следовательно, достаточность доказана.

Этот результат получен Уитни [8.12]. Поскольку он является попросту вариацией теоремы 8.9, будем говорить также о нем как о теореме Менгера.

Рассмотрим два частных класса k-связных графов: двусвязные и трехсвязные графы. Существует несколько эквивалентных характеризаций двусвязных графов. Некоторые из них уже приведены в упражнениях к гл. 1.

Татт охарактеризовал трехсвязные графы, исходя из частного класса графов, называемых *колесами*.

Рассмотрим цикл C длины n. Если мы добавим новую вершину и соединим ее со всеми вершинами цикла C, то получим колесо W_{n+1}. Например, колесо W_7 представлено на рис. 8.6.

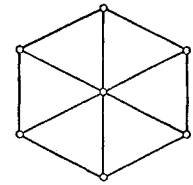


Рис. 8.6. Колесо W_7.

Характеризация Татта [8.13] трехсвязных графов формулируется в следующей теореме:

Теорема 8.11. Простой граф G трехсвязный тогда и только тогда, когда он является колесом или его можно получить из колеса последовательностью операций следующих типов:

1. Введением нового ребра.
2. Заменой вершины v степени не меньше 4 двумя такими смежными вершинами v' и v'' степеней не менее 3, что каждая вершина, смежная первоначально с вершиной v, становится смежной точно с одной из вершин v' и v''.

Закончим этот раздел характеризацией k-реберно-связных графов. Эта характеризация аналогична теореме 8.9, называемой теоремой Менгера, хотя открыта независимо Фордом и Фалкерсоном [8.14], а также Элиасом, Файнстейном и Шенноном [8.15].

Теорема 8.12. Минимальное число ребер, удаление которых из связного графа G разделяет две различные вершины s и t, равна максимальному числу реберно-непересекающихся s—t-путей в графе G.

Доказательство этой теоремы приводится в разд. 15.7

8.5. Паросочетания

В этом и последующих разделах главы будут обсуждаться проблема, называемая *задачей о свадьбах*, и связанные с ней задачи.

Имеется конечное множество юношей, у каждого из которых есть несколько подруг. При каких условиях можно поженить юношей так, чтобы каждый женился на одной из своих подруг? (Разумеется, никакая девушка не выходит замуж более чем за одного юношу!)

Эта задача ставится в теоретико-графовых терминах следующим образом:

Построить двудольный граф G, в котором вершины $x_1, x_2, \ldots, x_n$ представляют юношей, а вершины $y_1, y_2, \ldots, y_m$ — девушек. Ребро (x_i, y_j) присутствует в графе G тогда и только тогда, когда y_j — подруга x_i. Тогда задача о свадьбах равносильна нахождению в графе G такого множества ребер, что никакие два из них не имеют общую вершину и каждая вершина x_i является концевой одного из ребер.

Например, предположим, что имеются четверо юношей: b_1, b_2, b_3 и b_4, а также четыре девушки: g_1, g_2, g_3 и g_4; их взаимоотношения представлены следующим образом: $b_1 \to \{g_1\}$, $b_2 \to \{g_2\}$, $b_3 \to \{g_1, g_2\}$, $b_4 \to \{g_3, g_4\}$.

Двудольный граф, характеризующий данный случай, представлен на рис. 8.7. Легко выяснить, что невозможно поженить всех четырех юношей так, чтоб каждый женился на одной из своих подруг. Однако можно женить трех юношей, не нарушая при этом требований задачи о свадьбах. Например, возможны два следующих набора таких пар: (b_1, g_1), (b_2, g_2), (b_4, g_3); (b_1, g_1), (b_3, g_2), (b_4, g_4). Рассмотрение задачи о свадьбах приводит нас к определению паросочетания графа.

Два ребра называются *независимыми*, если они не имеют общей вершины. Говорят, что ребра e_1, e_2, . . . *независимы*, если каждая их пара не имеет общей вершины.

Паросочетание — это множество независимых ребер графа. Например, паросочетанием является $\{e_1, e_4\}$ в графе на рис. 8.8.

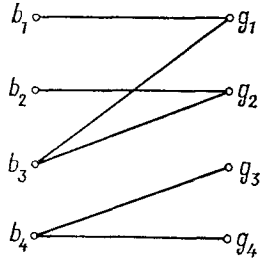


Рис. 8.7

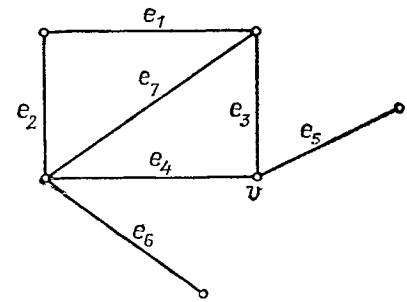


Рис. 8.8.

Очевидно, максимальное по включению паросочетание — это максимальное по включению множество независимых ребер. Так, в графе на рис. 8.8 максимальным по включению паросочетанием является $\{e_1, e_4\}$, в то время как $\{e_5, e_6\}$ таковым не является.

Паросочетание с наибольшим числом ребер называется *максимальным паросочетанием*. Множество $\{e_1, e_5, e_6\}$ — это максимальное паросочетание в графе на рис. 8.8. Число ребер в максимальном паросочетании графа G будет называться *числом паросочетания* графа G и обозначаться через $\alpha_1(G)$.

Вершина называется *насыщенной* в паросочетании M, если она концевая вершина ребра M. Например, вершина v паросочетания $\{e_1, e_5, e_6\}$ в графе на рис. 8.8 насыщенна.

Далее двудольный граф $G=(V, E)$ с разбиением (X, Y) будем обозначать тройкой (X, Y, E).

Будем говорить, что множество X *паросочетается* с Y в двудольном графе (X, Y, E), если существует такое паросочетание M,

что каждая вершина X насыщена в M. Паросочетание M в этом случае называется *полным паросочетанием* X с Y.

В этих терминах задача о свадьбах становится равносильной определению необходимых и достаточных условий существования в двудольном графе (X, Y, E) паросочетания X с Y. В следующем разделе мы приводим результаты, касающиеся паросочетаний в двудольных графах, а также решение задачи о свадьбах.

8.6. Паросочетания в двудольных графах

Рассмотрим двудольный граф $G=(X, Y, E)$. Пусть S — произвольное подмножество X, а $\Gamma(S)$ — множество вершин, смежных с вершинами S. Предположим, что $|S|>|\Gamma(S)|$. Ясно, что тогда не существует полного паросочетания S с $\Gamma(S)$ и что любое паросочетание M в графе G будет насыщать не более $|\Gamma(S)|$ вершин S. Поэтому

$$|M| \leqslant |X| - (|S| - |\Gamma(S)|). \tag{8.10}$$

Поскольку выражение (8.10) справедливо для любого подмножества S множества вершин X, можно заключить, что для любого паросочетания M

$$|M| \leqslant |X| - \max_{S \subseteq X} (|S| - |\Gamma(S)|). \tag{8.11}$$

Введем определение *дефицита* $\sigma(S)$ подмножества S множества X и дефицита $\sigma(G)$ графа G в виде

$$\sigma(S) = |S| - |\Gamma(S)|, \tag{8.12}$$

$$\sigma(G) = \max_{S \subseteq X} (|S| - |\Gamma(S)|). \tag{8.13}$$

Используя выражение (8.13), перепишем теперь неравенство (8.11) в виде

$$|M| \leqslant |X| - \sigma(G). \tag{8.14}$$

Основной результат этого раздела заключается в том, что в двудольном графе $G=(X, Y, E)$ существует паросочетание, содержащее $|X|-\sigma(G)$ ребер. Другими словами, мы покажем, что число ребер в максимальном паросочетании графа G равно $|X|-\sigma(G)$. Доказательство состоит из двух частей. Сначала мы рассмотрим случай $\sigma(G)=0$, затем случай $\sigma(G)>0$. (Заметим, что $\sigma(G)\geqslant 0$, поскольку для пустого множества $\varnothing$ $\sigma(\varnothing)=0$.)

Случай $\sigma(G)=0$ имеет место, когда для каждого подмножества $S \subseteq X$ справедливо неравенство $|S| \leqslant |\Gamma(S)|$, а случай $\sigma(G)>0$, когда для некоторого подмножества справедливо неравенство $|S|>|\Gamma(S)|$.

Следующий результат получен Холлом [8.16]. Используемое нами доказательство предложили Халмос и Вохган [8.17].

Теорема 8.13 (Холл.) В двудольном графе $G=(X, Y, E)$ существует полное паросочетание X с Y тогда и только тогда, когда для любого $S\subseteq X$ справедливо неравенство $|S|\leqslant|\Gamma(S)|$.

Доказательство. Необходимость. Из выражения (8.14) следует, что для полного паросочетания X с Y необходимо, чтобы $\sigma(G)=0$, т. е. $|S|\leqslant|\Gamma(S)|$, $S\subseteq X$.

Достаточность. Доказательство проведем индукцией по $|X|$, числу вершин в X. Если $|X|=1$, то очевидно, что существует полное паросочетание X с Y, так как единственная вершина X должна быть смежна по меньшей мере с одной вершиной Y.

В качестве индуктивного предположения допустим, что достаточность справедлива для любого двудольного графа с $|X|\leqslant m-1$. Рассмотрим двудольный граф $G=(X, Y, E)$ с $|X|=m$. Пусть для каждого подмножества S множества X справедливо неравенство $|S|\leqslant|\Gamma(S)|$. Проверкой следующих случаев покажем, что существует полное паросочетание X с Y.

Случай 1. Для каждого непустого собственного подмножества $S\subseteq X$ $|S|<|\Gamma(S)|$.

Выберем произвольное ребро (x_0, y_0), $x_0\in X$, $y_0\in Y$. Пусть G'— граф, полученный из графа G удалением x_0 и y_0 и инцидентных им ребер. Так как вершины любого подмножества $S\subseteq X-\{x_0\}$ смежны с более чем $|S|$ вершинами Y, то они смежны не менее чем с $|S|$ вершинами $Y-\{y_0\}$. Поэтому по индуктивному предположению существует полное паросочетание $X-\{x_0\}$ с $Y-\{y_0\}$. Это паросочетание вместе с ребром (x_0, y_0) дает полное паросочетание X с Y.

Случай 2. Существует такое непустое подмножество $S_0\subseteq X$, что $|S_0|=|\Gamma(S_0)|$.

Пусть G' — подграф графа G, содержащий вершины множеств S_0 и $\Gamma(S_0)$ и соединяющие эти вершины ребра, а G'' — подграф графа G, содержащий вершины множеств $X-S_0$ и $Y-\Gamma(S_0)$ и соединяющие их ребра.

Покажем, что в подграфе G' существует полное паросочетание S_0 с $\Gamma(S_0)$, а в подграфе G'' — полное паросочетание $X-S_0$ с $Y-\Gamma(S_0)$. Эти паросочетания образуют в совокупности полное паросочетание X с Y.

Рассмотрим сначала подграф G'. Пусть для любого подмножества $S\subseteq S_0$ $\Gamma'(S)$ означает множество вершин $\Gamma(S_0)$, смежных с вершинами S. Тогда ясно, что $\Gamma'(S)=\Gamma(S)$, $S\subseteq S_0$. Поскольку для любого подмножества $|S|\leqslant|\Gamma(S)|=|\Gamma'(S)|$, из индуктивного предположения следует, что существует полное паросочетание S_0 с $\Gamma(S_0)$.

Теперь рассмотрим подграф G''. Пусть для любого подмножества множества $X-S_0$ величина $\Gamma''(S)$ означает множество вершин $Y-\Gamma(S_0)$, смежных с вершинами S. Для любого $S\subseteq X-S_0$ $|S\cup S_0|=|S|+|S_0|\leqslant|\Gamma(S\cup S_0)|=|\Gamma''(S)|+|+|\Gamma(S_0)|$. Поскольку $|S_0|=|\Gamma(S_0)|$, получаем, что $|S|\leqslant|\Gamma''(S)|$, $S\subseteq X-S_0$. Таким образом, из индуктивного предположения снова следует, что существует полное паросочетание $X-S_0$ с $Y-\Gamma(S_0)$. Следовательно, достаточность доказана.

Как мы установили ранее, теорема 8.13 дает решение задачи о свадьбах, оно формулируется следующим образом:

Теорема 8.14 (Холл). Для решения задачи о свадьбах необходимо и достаточно, чтобы любое подмножество из k юношей имело вместе не менее k подруг, $1\leqslant k\leqslant m$, где m — число юношей.

Теорема 8.13 является простым переводом теоремы 8.14 на теоретико-графовый язык.

Теперь мы докажем, что если $\sigma(G)>0$, то число ребер в максимальном паросочетании равно $|X|-\sigma(G)$. Чтобы сделать это, необходимы следующие две леммы:

Лемма 8.1. Пусть S_1 и S_2 — произвольные подмножества X. Тогда $\sigma(S_1\cup S_2)+|+\sigma(S_1\cap S_2)\geqslant\sigma(S_1)+\sigma(S_2)$.

Доказательство. Простым упражнением является доказательство того, что

$$|S_1\cup S_2|+|S_1\cap S_2|=|S_1|+|S_2|. \tag{8.15}$$

Поскольку $|\Gamma(S_1\cup S_2)|=|\Gamma(S_1)\cup\Gamma(S_2)|$ и $|\Gamma(S_1\cap S_2)|\leqslant|\Gamma(S_1)\cap\Gamma(S_2)|$, то

$$|\Gamma(S_1\cup S_2)|+|\Gamma(S_1\cap S_2)|\leqslant|\Gamma(S_1)\cup\Gamma(S_2)|+|+|\Gamma(S_1)\cap\Gamma(S_2)|=|\Gamma(S_1)|+|\Gamma(S_2)|. \tag{8.16}$$

Подставляя выражение (8.16) в выражение (8.15), получим $\sigma(S_1 \cup S_2)+\sigma(S_1 \cap S_2) \geqslant |S_1|+|S_2|-|\Gamma(S_1)|-|\Gamma(S_2)|=\sigma(S_1)+\sigma(S_2)$.

Лемма 8.2. Пусть S_1 и S_2 — такие подмножества X, что $\sigma(S_1)=\sigma(S_2)=\sigma(G)$. Тогда $\sigma(S_1 \cup S_2)=\sigma(S_1 \cap S_2)=\sigma(G)$.

Доказательство. По лемме 8.1 $\sigma(S_1 \cup S_2)+\sigma(S_1 \cap S_2) \geqslant \sigma(S_1)+\sigma(S_2)=2\sigma(G)$. Так как ни $\sigma(S_1 \cup S_2)$, ни $\sigma(S_1 \cap S_2)$ не превышают $\sigma(G)$, получаем $\sigma(S_1 \cup S_2)=\sigma(S_1 \cap S_2)=\sigma(G)$.

Теорема 8.15. Число ребер в максимальном паросочетании двудольного графа $G=(X, Y, E)$ с $\sigma(G)>0$ равно $|X|-\sigma(G)$.

Доказательство. Пусть $S_1, S_2, \ldots, S_k$ — все подмножества X с дефицитом, равным $\sigma(G)$. Пусть также $S_0=S_1 \cap S_2 \cap \ldots \cap S_k$. По лемме 8.2 $\sigma(S_0)=\sigma(G)>0$. Таким образом, S_0 непусто. Следовательно, каждое подмножество X, имеющее дефицит $\sigma(G)$, содержит S_0.

Рассмотрим произвольную вершину x_0 в S_0. Пусть G' — граф, полученный удалением из графа G вершины x_0 и всех инцидентных ей ребер. Очевидно, что $\sigma(G')<\sigma(G)$, так как никакое подмножество множества $X-\{x_0\}$ не содержит S_0. Покажем, что $\sigma(G')=\sigma(G)-1$.

Рассмотрим множество $S_0'=S_0-\{x_0\}$. Имеем $\sigma(S_0')=|S_0'|-|\Gamma(S_0')|=|S_0|-1-|\Gamma(S_0')|$. Поскольку $\sigma(S_0')<\sigma(G)$, получаем $|S_0|-1-|\Gamma(S_0')|<|S_0|-|\Gamma(S_0)|$, т.е.

$$|\Gamma(S_0')| \geqslant |\Gamma(S_0)|. \tag{8.17}$$

С другой стороны, так как S_0' — подмножество S_0, то

$$|\Gamma(S_0')| \leqslant |\Gamma(S_0)|. \tag{8.18}$$

Объединяя выражения (8.17) и (8.18), получим $|\Gamma(S_0')|=|\Gamma(S_0)|$. Поэтому $\sigma(S_0')=|S_0|-1-|\Gamma(S_0)|=\sigma(G)-1$. Поскольку $\sigma(G')<\sigma(G)$, мы можем заключить, что $\sigma(G')=\sigma(G)-1$.

Если мы будем повторять эти рассуждения до тех пор, пока из X не будет удалено соответствующее множество из $\sigma(G)$ вершин вместе с инцидентными им ребрами, получим подграф графа G с дефицитом, равным нулю. По теореме 8.13 существует полное паросочетание такого подграфа. Оно будет паросочетанием графа G, содержащим $|X|-\sigma(G)$ ребер. Очевидно, что из выражения (8.14) следует, что это паросочетание будет максимальным паросочетанием графа G.

Теоремы 8.13 и 8.15 можно объединить в одну. Вследствие ее важности мы представим ее ниже [8.18, 8.19].

Теорема 8.16 (Кёниг). Число ребер в максимальном паросочетании двудольного графа $G=(X, Y, E)$ равно $|X|-\sigma(G)$, где $\sigma(G)$ — дефицит графа G.

Следствие 8.16.1. В непустом двудольном графе $G=(X, Y, E)$ существует полное паросочетание X с Y, если $\min\limits_{x \in X}\{d(x)\} \geqslant \max\limits_{y \in Y}\{d(y)\}$.

Доказательство. Пусть $\min\limits_{x \in X}\{d(x)\}=d_1$ и $\max\limits_{y \in Y}\{d(y)\}=d_2$. Рассмотрим произвольное подмножество A множества X. Пусть E_1 — множество ребер, инцидентных вершинам A, а E_2 — множество ребер, инцидентных вершинам $\Gamma(A)$. Тогда $|E_1| \geqslant |A|d_1$ и $|E_2| \leqslant |\Gamma(A)|d_2$. Поскольку $E_1 \subseteq E_2$, то $|\Gamma(A)|d_2 \geqslant |E_2| \geqslant |E_1| \geqslant |A|d_1$. Поэтому $|\Gamma(A)| \geqslant |A|$, $A \subseteq X$. Таким образом, по теореме Холла существует полное паросочетание X с Y.

Теперь рассмотрим два связанных с теоремой Холла результата. Первый принадлежит теории трансверсалей.

Пусть M — непустое конечное множество, а $S=\{S_1, S_2, \ldots, S_r\}$ — семейство (не обязательно различных непустых подмножеств множества M. Тогда *трансверсаль* (или *система различных представителей*) семейства S есть множество r различных элементов множества M, по одному из каждого множества S_i.

Например, если $M=\{1, 2, 3, 4, 5, 6\}$, $S_1=\{1, 3, 4\}$, $S_2=\{1, 3, 4\}$, $S_3=\{1, 2, 5\}$ и $S_4\{5, 6\}$, то $\{1, 3, 2, 6\}$ является трансверсалью семейства $S=\{S_1, S_2, S_3, S_4\}$. С другой стороны, если $S_1=S_2=\{1, 3\}$, $S_3=\{3, 4\}$ и $S_4=\{1, 4\}$, то для семейства $\{S_1, S_2, S_3, S_4\}$ трансверсаль не существует.

Возникает вопрос: каковы необходимые и достаточные условия того, что семейство подмножеств множества имеет трансверсаль?

Построим такой двудольный граф $G=(X, Y, E)$, что

1) вершина $x_i \in X$ соответствует множеству S_i семейства S;
2) вершина $y_i \in Y$ соответствует элементу i множества M;
3) ребро $(x_i, y_j) \in E$ тогда и только тогда, когда $j \in S_i$.

Теперь становится ясно, что поставленный вопрос эквивалентен задаче нахождения полного паросочетания X с Y в только что построенном двудольном графе. Таким образом, получаем следующую теорему, которая просто переводит теорему Холла на язык теории трансверсалей.

Теорема 8.17. Пусть M — непустое множество, а $S=\{S_1, S_2, \ldots, S_r\}$ — семейство подмножеств множества M. Тогда S имеет трансверсаль тогда и только тогда, когда объединение произвольных k $(1 \leqslant k \leqslant r)$ подмножеств S_i содержит не менее элементов множества M.

Чисто комбинаторное доказательство этой теоремы без использования понятий теории графов дано Радо. Его очень элегантное доказательство можно найти в работе [8.20].

Следующий результат связан с матрицами, имеющими только нулевые и единичные элементы. Такие матрицы называются *(0,1)-матрицами*. Далее под линией матрицы будем понимать ее строку или столбец.

Теорема 8.18 (Кёниг и Эгервари). Минимальное число линий, содержащих все «1» в (0,1)-матрице, равно максимальному числу «1», никакие две из которых не находятся на одной линии матрицы M.

Дана (0,1)-матрица M порядка $m \times n$. Построим такой двудольный граф $G=(X, Y, E)$, что

1) вершины $x_1, x_2, \ldots, x_m \in X$ соответствуют m строкам матрицы M;
2) вершины $y_1, y_2, \ldots, y_n \in Y$ соответствуют n столбцам матрицы M;
3) ребро $(x_i, y_j) \in E$, если элемент (i, j) матрицы M равен 1.

Если мы рассмотрим вершину как покрывающую все инцидентные ей ребра, то теорему Кёнига — Эгервари можно переформулировать следующим образом:

Минимальное число вершин двудольного графа, покрывающих все ребра, равно числу ребер в любом максимальном паросочетании графа.

В гл. 9 (теорема 9.2) мы докажем теорему Кёнига — Эгервари в этой формулировке.

8.7. Паросочетания графов общего вида

В этом разделе мы сформулируем результаты, относящиеся к паросочетаниям графов общего вида.

Рассмотрим граф $G=(V, E)$ и его паросочетание M. *Чередующаяся цепь* графа G — это цепь, ребра которой входят поочередно в M и $(E—M)$. Например, последовательность ребер e_1, e_2, e_3, e_4, e_7, e_6 является чередующейся цепью по отношению к паросочетанию $M=\{e_2, e_4, e_6\}$ графа на рис. 8.9. Те ребра чередующейся цепи, которые принадлежат M, будем называть *темными ребрами*, а те, что принадлежат $E—M$,— *светлыми ребрами*. Следовательно, e_1, e_3, e_7 — светлые ребра, а e_2, e_4, e_6 — темные ребра рассмотренной чередующейся цепи.

Теорема 8.19. Пусть M_1 и M_2 — паросочетания простого графа $G=(V, E)$, а $G'=(V', E')$ — порожденный подграф графа G на множестве ребер $M_1 \oplus M_2 = (M_1—M_2) \cup (M_2—M_1)$. Тогда подграф G' содержит компоненты только следующих двух типов:

1. Цикл четной длины, ребра которого входят поочередно в M_1 и M_2.
2. Путь, в котором ребра входят поочередно в M_1 и M_2, а концевые вершины в одном из паросочетаний не насыщены.

Доказательство. Рассмотрим произвольную вершину $v \in V'$.

Случай 1. $v \in V(M_1—M_2)$ и $v \in V(M_2—M_1)$, где $V(M_i—M_j)$ означает множество вершин, инцидентных ребрам $M_i—M_j$. В этом случае v — концевая вершина

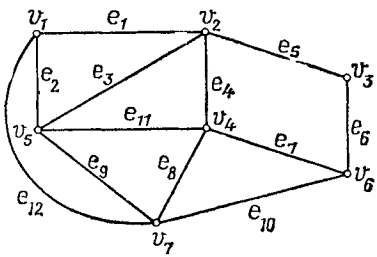


Рис. 8.9.

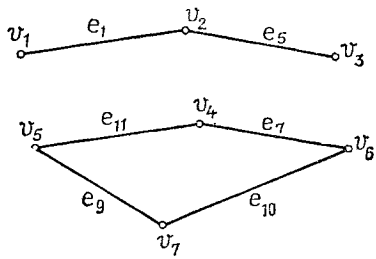


Рис. 8.10.

ребра в $M_1—M_2$. Поскольку M_1 — паросочетание, никакое другое ребро из $M_1—M_2$ не инцидентно вершине v. Более того, никакое ребро из $M_2—M_1$ не инцидентно вершине v, поскольку $v \in V(M_2—M_1)$. Следовательно, в этом случае степень вершины v в G' равна 1.

Случай 2. $v \in V(M_1—M_2)$ и $v \in V(M_2—M_1)$. В этом случае вершине v инцидентны одно ребро $M_1—M_2$ и одно ребро $M_2—M_1$. Следовательно, степень вершины v равна 2.

Поскольку рассмотренные случаи исчерпывают возможные варианты, можно сделать вывод, что максимальная степень в подграфе G' равна 2. Поэтому связная компонента может принадлежать только к одному из описанных типов.

Например, рассмотрим два паросочетания $M_1=\{e_5, e_7, e_9\}$ и $M_2=\{e_1, e_{10}, e_{11}\}$ графа G на рис. 8.9. Тогда $M_1 \oplus M_2=\{e_1, e_5, e_7, e_9, e_{10}, e_{11}\}$, а граф G' будет таким, как показано на рис. 8.10. Граф G' содержит компоненты двух типов, описанных в теореме 8.19.

В следующей теореме мы формулируем характеризацию Бержа [8.21] максимальных паросочетаний в терминах чередующихся цепей.

Теорема 8.20 (Берж). Паросочетание M максимально тогда и только тогда, когда между любыми двумя не насыщенными в нем вершинами не существует чередующейся цепи.

Доказательство.

Необходимость. Предположим, что между двумя не насыщенными в M вершинами имеется чередующаяся цепь P. Тогда, заменяя темные ребра в цепи на светлые, получим паросочетание M_1 с $|M_1|=M+1$. Заметим, что $M_1=(M-P)\cup(P-M)$.

Можно рассмотреть, например, паросочетание $M=\{e_2, e_4\}$ в графе на рис. 8.9. Между ненасыщенными вершинами v_3 и v_7 имеется чередующаяся цепь e_5, e_4, e_{11}, e_2, e_{12}. Если заменить в M темные ребра e_4 и e_2 на светлые e_5, e_{11}, e_{12}, получим паросочетание $\{e_5, e_{11}, e_{12}\}$, содержащее на одно ребро больше, чем M.

Достаточность. Допустим, что M удовлетворяет условиям теоремы. Пусть M' — максимальное паросочетание. Тогда из доказанной необходимости следует, что M' также удовлетворяет условиям теоремы, а именно: не существует чередующейся цепи между произвольными двумя не насыщенными в M' вершинами. Покажем сейчас, что $|M|=|M'|$, доказывая этим достаточность.

Так как $M=(M\cap M')\cup(M-M')$, а $M'=(M\cap M')\cup(M'-M)$, то очевидно, что $|M|=|M'|$ тогда и только тогда, когда $|M-M'|=|M'-M|$.

Пусть G' — граф на множестве ребер $M\oplus M'=(M-M')\cup(M'-M)$.

Рассмотрим сначала произвольный цикл графа G'. По теореме 8.19 он имеет четную длину, ребра в нем поочередно входят в $M-M'$ и $M'-M$. Поэтому каждый цикл графа G' имеет одинаковое число ребер в $M-M'$ и $M'-M$.

Теперь рассмотрим произвольную компоненту графа G', являющуюся путем. Снова по теореме 8.19 ребра в нем входят поочередно в $M-M'$ и $M'-M$. Кроме того, концевые вершины такого пути не насыщены в M или M'. Предположим, что путь имеет нечетную длину, тогда концевые вершины пути инцидентны ребрам, находящимся в одном паросочетании. Это означает, что по отношению к одному из паросочетаний существует чередующаяся цепь, соединяющая ненасыщенные вершины. Но это противоречит тому, что и M, и M' удовлетворяют условиям теоремы. Поэтому каждая компонента графа G', являющаяся путем, содержит четное число ребер и, следовательно, одинаковое число ребер из $M-M'$ и $M'-M$.

Таким образом, каждая компонента графа G' содержит одинаковое число ребер из $M-M'$ и $M'-M$. Поскольку ребра графа G' образуют множество $(M-M')\cup(M'-M)$, получаем $|M-M'|=|M'-M|$ и, следовательно, $|M|=|M'|$.

Пусть M — паросочетание графа G, а P — чередующаяся цепь между произвольными двумя не насыщенными в M вершинами. Тогда, как мы уже увидели, $M\oplus P$ является паросочетанием с числом ребер на единицу больше, чем в M. По этой причине путь P называется *добавляющим путем* по отношению к M.

Теперь мы докажем два интересных результата о двудольных графах, используя теорию чередующихся цепей.

Рассмотрим двудольный граф $G=(X, Y, E)$ с максимальной степенью Δ. Пусть X_1 — множество вершин в X, имеющих степень Δ. Если G' — двудольный граф $(X_1, \Gamma(X_1), E')$, где E' — множество ребер, соединяющих X_1 и $\Gamma(X_1)$, то из следствия 8.16.1 вытекает, что в графе G' имеется полное паросочетание X_1 с $\Gamma(X_1)$. Оно, очевидно, насыщает все вершины X_1. Таким образом, существует паро-

сочетание в графе G, которое насыщает все вершины X степени Δ. Аналогично существует паросочетание в графе G, которое насыщает все вершины Y степени Δ. Сейчас возникает вопрос: существует ли паросочетание двудольного графа, которое насыщает все вершины максимальной степени как X, так и Y? Для ответа на этот вопрос нам необходим следующий результат, полученный Мендельсоном и Далмеджем [8.22].

Теорема 8.21. (Мендельсон и Далмедж). Пусть $G=(X, Y, E)$ — двудольный граф, а M_i — паросочетание, которое «сочетает» $X_i \subseteq X$ с $Y_i \subseteq Y$ $(i=1, 2)$. Тогда существует паросочетание $M' \subseteq M_1 \cup M_2$, которое насыщает X_1 и Y_2.

Доказательство. Рассмотрим двудольный граф $G'=(X_1 \cup X_2, Y_1 \cup Y_2, M_1 \cup M_2)$. Любая вершина этого графа имеет степень 1 или 2, следовательно, любая его компонента является путем или циклом, ребра которых входят поочередно в M_1 и M_2. (См. доказательство теоремы 8.19.)

Любая вершина $y \in Y_2 - Y_1$ имеет в графе G' степень 1. Поэтому она лежит в компоненте, являющейся путем P_y от вершины y к вершине $x \in X_2 - X_1$ или к вершине $z \in Y_1 - Y_2$. В первом случае последнее ребро P_y находится в M_2, поэтому $M_1 \oplus P_y$ паросочетает $X_1 \cup \{x\}$ с $Y_1 \cup \{y\}$. Во втором случае последнее ребро P_y находится в M_2, поэтому $M_1 \oplus P_y$ паросочетает X_1 с $(Y_1 - z) \cup \{y\}$. В обоих случаях $M_1 \oplus P_y$ насыщает $Y_1 \cap Y_2$. Таким образом, $M_1 \oplus P_y$ насыщает $y \in Y_2 - Y_1$ и все вершины X_1 и $Y_1 \cap Y_2$.

Если мы положим $P = \bigcup_{y \in Y_2 - Y_1} P_y$, то легко видеть, что $M_1 \oplus P$ является паросочетанием, насыщающим X_1 и Y_2. Это и есть требуемое паросочетание $M' \subseteq M_1 \cup M_2$.

Теорема 8.22. В двудольном графе имеется паросочетание, насыщающее все вершины максимальной степени.

Доказательство. Рассмотрим двудольный граф $G=(X, Y, E)$. Пусть $X' \subseteq X$ и $Y' \subseteq Y$ содержат все вершины графа G максимальной степени. Как мы видели ранее, существует паросочетание M_1, насыщающее все вершины X', и паросочетание M_2, насыщающее все вершины Y'. По теореме 8.21 существует паросочетание $M' \subseteq M_1 \cup M_2$, насыщающее все вершины X' и Y', являющееся поэтому требуемым паросочетанием, которое насыщает все вершины графа G максимальной степени.

Следствие 8.22.1. Множество ребер двудольного графа с максимальной степенью Δ можно разбить на Δ паросочетаний.

Доказательство. Рассмотрим двудольный граф $G=(X, Y, E)$ с максимальной степенью Δ. По теореме 8.22 существует паросочетание M_1, насыщающее все вершины степени Δ. Тогда двудольный граф $G'=(X, Y, E - M_1)$ имеет максимальную степень $\Delta-1$. Этот граф имеет паросочетание M_2, насыщающее каждую вершину степени $\Delta-1$. Повторяя этот процесс, мы построим последовательность непустых паросочетаний $M_1, M_2, \ldots, M_\Delta$, которые образуют разбиение E.

Применение теорем 8.21, 8.22 и следствия 8.22.1 рассматривается в разд. 15.6.

Вернемся к изучению паросочетаний в графах общего вида.

Паросочетание, насыщающее все вершины графа G, называется *совершенным* паросочетанием этого графа G.

Мы заканчиваем этот раздел теоремой Татта [8.23], дающей условия существования совершенного паросочетания графа. Используемое нами доказательство предложил Андерсон [8.24].

Компонента графа называется *нечетной*, если она содержит нечетное число вершин, и в противном случае — *четной*. Если S —

подмножество вершин графа G, то через $p_0(S)$ будем обозначать число нечетных компонент графа $G-S$.

Теорема 8.23. (Татт). Граф $G=(V, E)$ имеет совершенное паросочетание тогда и только тогда, когда

$$p_0(S) \leqslant |S| \text{ для любого } S \subset V. \tag{8.19}$$

Доказательство. Необходимость. Допустим, что граф G имеет совершенное паросочетание M. Пусть $G_1, G_2, \ldots, G_k$ будут нечетными компонентами графа $G-S$ для некоторого $S \subset V$. Так как каждая компонента G_i нечетная, некоторая вершина v_i в ней должна паросочетаться в M с некоторой вершиной $v_j \in S$. Таким образом, S имеет не менее k вершин, и, следовательно, $p_0(S) \leqslant |S|$.

Достаточность. Прежде всего заметим, что если граф G удовлетворяют условию (8.19), то, выбрав $S=\varnothing$, получим $p_0(\varnothing) \leqslant 0$. Поэтому в графе G нет нечетных компонент. Другими словами, граф G имеет четное число вершин.

Доказательство достаточности проведем индукцией по l, где $2l=|V|$. Используем теорему Холла (8.13) и тот простой факт, что

$$p_0(S) = |S| \pmod 2. \tag{8.20}$$

Случай $l=1$ — тривиален. Допустим, что результат верен для любых графов, имеющих менее $2l$ вершин. Рассмотрим теперь граф G на $2l$ вершинах, удовлетворяющий условию (8.19). Здесь мы имеем два случая.

Случай 1. Допустим, что $p_0(S) < |S|$ для любого S, $2 \leqslant |S| < 2l$. Рассмотрим любое ребро графа $Ge=(a, b)$. Пусть $A=\{a, b\}$, $G_A=G-A$. Пусть $p_0'(T)$ обозначает число нечетных компонент в графе G_A-T для любого подмножества T вершин G_A. Тогда $p_0'(T) \leqslant |T|$; если бы $p_0'(T) > |T|$, то мы получим $p_0(T \cup A) = p_0'(T) > |T| = |TUA| - 2$, что противоречит условию (8.19). Поэтому по индукции графы G_A, а следовательно, и G имеют совершенное паросочетание.

Случай 2. Пусть существует такое множество S, что $p_0(S) = |S| \geqslant 2$. Допустим, что S — такое максимальное множество. Заметим сначала, что в графе $G-S$ нет четных компонент. Если бы в нем была четная компонента, мы могли бы удалить из нее вершину и добавить ее к S. Это привело бы к увеличению по крайней мере на 1 числа четных компонент. Поэтому $p_0(S \cup v) \geqslant p_0(S)+1 = |S|+1$. Условие (8.19) требует, чтобы $p_0(S \cup v) \leqslant |S|+1$. Поэтому $p_0(S \cup v) = |S|+1$. Но это противоречит максимальности S. Следовательно, четных компонент в графе $G-S$ нет.

Пусть $|S|=s$, а $G_1, G_2, \ldots, G_s$ — s нечетных компонент графа $G-S$. Покажем теперь, что мы можем взять по вершине из каждой этой компоненты и паросочетать их с вершинами S. Если бы это было невозможно, тогда по теореме Холла существовало бы k нечетных компонент, которые связывались в графе G только с $h<k$ вершинами S. Но если T — такое множество из h вершин, то мы получаем $p_0(T) \geqslant k > h = |T|$, что противоречит условию (8.19). Таким образом, из каждой компоненты G_i $(1 \leqslant i \leqslant s)$ мы можем взять вершину v_i и паросочетать ее с вершиной S.

Следовательно, каждый подграф $G_i'=G_i-v_i$ имеет четное число вершин. Если мы покажем, что каждый G_i' имеет совершенное паросочетание, доказательство будет завершено.

Если G_i' содержит такое множество вершин R, что $p_0''(R) > |R|$, где $p_0''(R)$ — число нечетных компонент в $G_i'-R$, то по (8.20) $p_0''(R) \geqslant |R|+2$, поэтому

$$p_0(R \cup S \cup \{v_i\}) = p_0''(R) + p_0(S) - 1 \geqslant |R| + |S| + 1 = |R \cup S \cup \{v_i\}|. \tag{8.21}$$

Но условие (8.19) требует, чтобы $p_0(R \cup S \cup \{v_i\}) \leqslant |R \cup S \cup \{v_i\}|$. Поэтому $p_0(R \cup S \cup \{v_i\}) = |R \cup S \cup \{v_i\}|$, что противоречит условию максимальности S. Таким образом, по индукции G_i' имеет совершенное паросочетание.

Другое доказательство с использованием теоремы 8.20 дал Ловац [8.25].

8.8. Замечания, касающиеся литературы

Общей литературой, рекомендуемой для дальнейшего изучения связности и паросочетаний, являются [8.8, 8.26]. Берж [8.26] представляет также детальное обсуждение вопросов реализуемости последовательностей степеней и для случая ориентированных, и для случая неориентированных графов. Харари [8.8] излагает историю теоремы Менгера и дает несколько ее вариаций.

Графы служат моделью сетей связи. При изучении моделируемых графами сетей возникает понятие «уязвимость». Под уязвимостью мы понимаем чувствительность сети к воздействию. Сеть считается «разрушенной», если после удаления нескольких вершин или ребер получившийся в результате граф будет несвязным. Таким образом, уязвимость сети связана с вершинной и реберной связностями сети. Например, сеть N_1 может считаться более уязвимой, чем сеть N_2, если вершинная связность у N_1 меньше, чем у N_2.

Боеш [8.27] опубликовал несколько статей по построению графов с заданными свойствами связности и надежности. Этой же теме посвящены работы [8.28—8.30]. Для дальнейшего изучения этой темы рекомендуется работа [8.21].

Задача проверки связности графа тесно соприкасается с задачей нахождения максимального потока в транспортной сети. Дальнейшее обсуждение этого вопроса и связанные с ним ссылки на литературу можно найти в разд. 15.7, где мы также доказываем теоремы Менгера.

Для дальнейшего изучения паросочетаний и теории трансверсалей очень рекомендуются работы [8.32—8.34]. Применение теории паросочетаний (в частности, к задаче оптимального назначения и задаче составления расписаний) и соответствующие алгоритмы обсуждаются в разд. 15.4—15.6.

Упражнения

8.1. Пусть $d_1 \leqslant d_2 \leqslant \ldots \leqslant d_n$ — степени простого графа G. Покажите, что G — k-связный граф $(k<n)$, если а) $d(v_r) \geqslant r+k-1$, $r \leqslant (n-k)/2$; б) $d(v_{n-k+1}) \geqslant (n+k-2)/2$.

8.2. Покажите, что простой граф G на n вершинах — k-связный, если $\delta(G) \geqslant (n+k-2)/2$.

8.3. Пусть $G=(V, E)$ — простой k-связный граф, $B=\{b_1, b_2, \ldots, b_k\}$ — множество вершин, $|B|=k$. Покажите, что если $a \in V-B$, то существует k вершинно-непересекающихся a—b_i-путей из a в B.

8.4. Пусть G — простой k-связный граф, $k \geqslant 2$. Покажите, что существует цикл, проходящий через два произвольно заданных ребра e_1, e_2 и $k-2$ вершины.

8.5. Покажите, что $\varkappa(H_{k,n})=\varkappa'(H_{k,n})=k$.

8.6. Найдите 5-связный граф на 7 вершинах и 18 ребрах.

8.7. Покажите, что ребра простого графа можно ориентировать таким образом, чтобы получился сильно связный граф, тогда и только тогда, когда граф G—2-реберно-связный.

8.8. *Ассоциированным ориентированным графом* $D(G)$ неориентированного графа G называется граф, полученный заменой каждого ребра e графа G на две противоположно направленные дуги, имеющие те же концевые вершины. Покажите, что

а) существует взаимно-однозначное соответствие между путями графа G и ориентированными путями графа $D(G)$; б) граф $D(G)$ k-реберно-связный тогда и только тогда, когда граф G k-реберно-связный.

Примечание. Ориентированный граф G называется *k-реберно-связным* тогда и только тогда, когда для разрушения всех путей между любыми двумя вершинами s и t графа G необходимо удалить не менее k ребер.

8.9. Найдите простой граф, имеющий последовательность степеней (5, 4, 4, 4, 3, 3, 3, 3, 3), с максимально возможной реберной связностью.

8.10. Покажите, что последовательность $(d_1, d_2, \ldots, d_n)$ неотрицательных целых является последовательностью степеней дерева тогда и только тогда, когда $d_i \geqslant 1$ для всех i и $\sum_{i=1}^{n} d_i = 2(n-1)$.

8.11. Покажите, что последовательность $(d_1, d_2, \ldots, d_n)$ неотрицательных целых является последовательностью степеней графа тогда и только тогда, когда $\sum_{i=1}^{n} d_i$ четна ([8.4]).

8.12. Покажите, что последовательность $(d_1, d_2, \ldots, d_n)$ неотрицательных целых является последовательностью степеней простого графа тогда и только тогда, когда а) $\sum_{i=1}^{n} d_i$ четна и б) $\sum_{i=1}^{k} d_i \leqslant k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\}$ для $1 \leqslant k \leqslant n-1$ [8.7].

8.13. Покажите, что последовательность $(d_1, d_2, \ldots, d_n)$ (реализуемая простым графом) является последовательностью степеней простого k-связного графа тогда и только тогда, когда а) $d_i \geqslant k$ для $1 \leqslant i \leqslant n$; б) $m - \sum_{i=1}^{k-1} d_i + (k-1)(k-2)/2 \geqslant n-k$, где m — число ребер графа G [8.6].

8.14. Докажите или опровергните: для любого паросочетания M существует такое максимальное паросочетание M', что $M \subseteq M'$.

8.15. Квадратная матрица P с действительными неотрицательными элементами называется *бистохастической*, если сумма элементов в каждой строке и в каждом столбце равна 1. Матрица подстановок — это (0,1)-матрица, содержащая точно по одной 1 в каждой строке и каждом столбце. Покажите, что бистохастическую матрицу P можно определить как $P = c_1P_1 + c_2P_2 + \ldots + c_kP_k$, где каждая P_i — матрица подстановок, каждое c_i — действительное неотрицательное число и $\sum_{i=1}^{k} c_i = 1$.

8.16. Пусть G — двудольный граф с двудольным разбиением (X, Y). Покажите, что если граф G имеет полное паросочетание X с Y, то существует такая вершина $x_0 \in X$, что для любой вершины $y \in \Gamma(x_0)$ по крайней мере одно максимальное паросочетание содержит ребро (x_0, y).

8.17. *k-фактором* графа G называется k-однородный остовный подграф графа G. Очевидно, что 1-фактор — это совершенное паросочетание. Граф G называется *k-факторизуемым*, если он является объединением нескольких реберно-непересекающихся k-факторов. Покажите, что $K_{n,n}$ и K_{2n} 1-факторизуемы.

8.18. а) Покажите, что K_{2n+1} можно определить как объединение связных 2-факторов ($n \geqslant 1$).

Примечание. Связный 2-фактор является гамильтоновым циклом.

б) Покажите, что K_{2n+1} есть объединение 1-фактора и $n-1$ связных 2-факторов.

8.19. Покажите, что связный граф G 2-факторизуем тогда и только тогда, когда он однородный с четной степенью.

8.20. Пусть M и N — реберно-непересекающиеся паросочетания графа G, причем $|M| > |N|$. Покажите, что существуют такие непересекающиеся паросочетания M' и N', что $|M'| = |M| - 1$, $|N'| = |N| + 1$ и $M' \cup M' = M \cup N$.

8.21. Покажите, что если $G=(V, E)$ — двудольный граф, а $k \geqslant \Delta$, то существует k таких непересекающихся паросочетаний $M_1, M_2, \ldots, M_k$ графа G, что $E = M_1 \cup M_2 \cup \ldots \cup M_k$ и $\lfloor m/k \rfloor \leqslant |M_i| \leqslant \lceil m/k \rceil$ для $1 \leqslant i \leqslant k$, где m — число ребер G.

8.22. Покажите, что дерево T имеет совершенное паросочетание тогда и только тогда, когда $p_0(v)=1$ для всех вершин T, где $p_0(v)$ — число нечетных компонент в $T-v$.

8.23. Докажите теорему Холла, используя теорему Татта (см. упражнение 5.3.1 в работе [8.35]).

8.24. Докажите теорему Холла, используя теорему Менгера (см. работу [8.36], теорема 28d).

9. Покрытия и раскраски

В предыдущих главах мы определили несколько полезных характеристик, связанных с графом: ранг, цикломатическое число, связность, число паросочетания и т. д. Как было упомянуто ранее, ранг и цикломатическое число используются при изучении электрических цепей, связность — при изучении сетей связи. В этой главе мы изучаем другие важные характеристики графа: число независимости и число вершинного и реберного покрытий, хроматический индекс и хроматическое число. Мы начинаем обсуждение с числа независимости и числа вершинного и реберного покрытий. Эти характеристики мы связываем с числом паросочетания, определенным в предыдущей главе, и устанавливаем эквивалентную формулировку теоремы Холла. Затем мы изучим хроматический индекс и хроматическое число, которые связываем со свойствами вершинной и реберной раскрашиваемости графа.

Обсуждаемые в этой главе характеристики используются при изучении некоторых практических задач, как, например, составление расписаний и исследование сетей связи.

9.1. Независимые множества и вершинные покрытия

Рассмотрим граф $G=(V, E)$. Подмножество $S \subseteq V$ называется *независимым множеством* графа G, если никакие две вершины S не смежны в графе G. Независимое множество называется также *внутренне устойчивым множеством*.

Независимое множество S графа G максимально, если граф G не содержит такого независимого множества S', что $|S'|>|S|$. Число вершин в максимальном независимом множестве графа G называется *числом независимости* (*числом внутренней устойчивости*) графа G и обозначается через $\alpha_0(G)$.

Например, в графе на рис. 9.1 множества $\{b, d\}$, $\{b, f\}$, $\{a, c\}$ и $\{b, d, f\}$ являются независимыми множествами. Из них $\{b, d\}$ и $\{b, f\}$ — не наибольшие независимые множества; $\{a, c\}$ — наибольшее, но не максимальное; $\{b, d, f\}$ — максимальное.

Подмножество K множества вершин V является *вершинным покрытием*, если любое ребро графа G имеет хотя бы одну концевую вершину в подмножестве K. Если рассматривать вершину как по-

крывающую все инцидентные ей ребра, тогда вершинное покрытие графа G есть подмножество V, которое покрывает все ребра графа G.

Вершинное покрытие K графа G *минимально*, если граф G не имеет такого вершинного покрытия K', что $|K'|<|K|$. Число вершин в минимальном вершинном покрытии графа G называется *числом вершинного покрытия* графа G и обозначается через $\beta_0(G)$.

Например, в графе на рис. 9.1 множества $\{a, c, e, f\}$, $\{a, c, d, e\}$, $\{b, d, e, f\}$ и $\{a, c, e\}$ являются вершинными покрытиями. Из них $\{a, c, e, f\}$ и $\{a, c, d, e\}$ — не наименьшие; $\{b, d, e, f\}$ — наименьшее, но не минимальное; $\{a, c, e\}$ — минимальное.

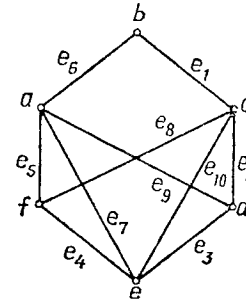


Рис. 9.1.

Когда рассматриваемый граф G понятен из контекста, величины $\alpha_0(G)$ и $\beta_0(G)$ будут обозначаться как α_0 и β_0 соответственно. Напомним, что $\alpha_1(G)$ — это число ребер в максимальном паросочетании графа G.

Независимые множества и вершинные покрытия — тесно связанные понятия, как мы показываем в следующей теореме.

Теорема 9.1. Рассмотрим граф $G=(V, E)$. Подмножество $S\subseteq V$ является независимым множеством графа G тогда и только тогда, когда его дополнение $\bar{S}$ подмножества S в V (т. е. $\bar{S}=V-S$) является вершинным покрытием.

Доказательство. По определению S — независимое множество графа G тогда и только тогда, когда никакое ребро G не содержит в подмножестве S обеих концевых вершин. Другими словами, S — независимое множество тогда и только тогда, когда любое ребро графа G имеет по крайней мере одну концевую вершину в дополнении $\bar{S}$ подмножества S в V. Тогда по определению вершинного покрытия теорема верна.

Следствие 9.1.1. Для простого графа на n вершинах $\alpha_0+\beta_0=n$.

Доказательство. Рассмотрим максимальное независимое множество S^* и минимальное вершинное покрытие K^* графа $G=(V, E)$. Тогда $|S^*|=\alpha_0$ и $|K^*|=\beta_0$.

Согласно теореме 9.1, $\bar{S}^*=V-S^*$ — вершинное покрытие, а $\bar{K}^*=V-K^*$ — независимое множество. Поэтому $|\bar{S}^*|=|V-S^*|=n-\alpha_0\geqslant\beta_0$ и $|\bar{K}^*|=|V-K^*|=n-\beta_0\geqslant\alpha_0$. Объединив эти неравенства, получим $\alpha_0+\beta_0=n$.

Рассмотрим произвольное максимальное паросочетание M^* и произвольное минимальное вершинное покрытие K^* графа G. Поскольку для покрытия ребер M^* требуется по крайней мере $|M^*|$ вершин, вершинное покрытие должно содержать хотя бы $|M^*|$ вершин. Поэтому

$$|M^*|\leqslant|K^*|. \tag{9.1}$$

В общем случае равенство в выражении (9.1) не всегда имеет место. Однако в следующей теореме мы показываем, что $|M^*|=|K^*|$, если G — двудольный граф.

Теорема 9.2. Число ребер в максимальном паросочетании двудольного графа равно числу вершин минимального вершинного покрытия, т. е. $\alpha_1=\beta_0$.

Доказательство. Пусть M^* — максимальное паросочетание, а K^* — минимальное вершинное покрытие двудольного графа $G=(X, Y, E)$.

Рассмотрим произвольное подмножество $A \subseteq X$. Каждое ребро e графа G инцидентно вершине в подмножестве A или в подмножестве $X-A$. Кроме того, любое ребро, инцидентное вершине в подмножестве A, инцидентно также вершине в множестве вершин $\Gamma(A)$, смежных с вершинами подмножества A. Таким образом, множество $(X-A) \cup \Gamma(A)$ является вершинным покрытием графа G и, следовательно, $|(X-A)|+|\Gamma(A)| \geqslant |K^*|$.

Но по теореме Холла 8.13

$$|M^*| = \min_{A \subseteq X} \{|X-A|+|\Gamma(A)|\} \geqslant |K^*|. \qquad (9.2)$$

Теперь, объединяя выражения (9.1) и (9.2), получаем $|M^*|=|K^*|$. Эта теорема принадлежит Кёнигу [9.1].

В теореме 8.20 мы дали характеризацию максимального паросочетания, используя понятие чередующейся цепи. Поскольку независимое множество — это вершинный аналог паросочетания, можно ожидать, что существует также подобная характеризация для максимального независимого множества. Это действительно так, и далее мы получим такую характеризацию. Будем следовать трактовке Бержа [9.2].

Сначала определим чередующуюся последовательность как вершинный аналог чередующейся цепи.

Пусть Y — независимое множество графа $G=(V, E)$. *Чередующаяся последовательность* по отношению к подмножеству Y есть последовательность $\sigma=\{x_1, y_1, x_2, y_2, \ldots\}$ различных вершин, принадлежащих поочередно подмножествам $X=\overline{Y}=V-Y$ и Y, так что выполняются следующие условия:

1) $x_i \in X$, $y_i \in Y$;
2) вершина y_i смежна хотя бы с одной вершиной из множества $\{x_1, x_2, \ldots, x_i\}$;
3) вершина x_{i+1} не смежна ни с одной вершиной из $\{x_1, x_2, \ldots, x_i\}$;
4) вершина x_{i+1} смежна хотя бы с одной вершиной из $\{y_1, y_2, \ldots, y_i\}$.

Чередующаяся последовательность максимальна, если не существует вершин, которые можно добавить к ней без нарушения указанных условий.

Например, в графе на рис. 9.1 последовательность $\{f, a, b, c\}$ является максимальной чередующейся последовательностью по отношению к независимому множеству $\{a, c\}$.

Рассмотрим дерево T. Напомним (упражнение 2.2), что оно является двудольным графом, т. е. множество V его вершин можно разбить на такие два подмножества X и Y, что

1) $V=X \cup Y$;
2) $X \cap Y=\varnothing$;
3) X и Y — независимые множества в дереве.

Будем называть (X, Y) *двудольным* разбиением множества вершин дерева T, а вершины подмножеств X и Y — X- и Y-вершинами соответственно.

В качестве первого шага на пути получения характеризации максимального независимого множества в виде чередующейся последовательности мы покажем, что если в дереве T выполняется неравенство $|X|>|Y|$, то существует чередующаяся последовательность по отношению к подмножеству Y, содержащая все вершины этого множества. Чтобы сделать это, нам необходима следующая лемма:

Лемма 9.1. Пусть (X, Y) — двудольное разбиение множества вершин дерева T. Если $|X|=|Y|+p$, $p\geqslant 0$, то подмножество X содержит по крайней мере $p+1$ висячих вершин дерева T.

Доказательство. Доказательство проводим индукцией по числу n вершин дерева T. Ясно, что лемма справедлива для $n=2, 3, 4$. Пусть этот результат имеет место для всех деревьев, имеющих менее n вершин ($n\geqslant 5$).

Рассмотрим дерево T на n вершинах с $|X|=|Y|+p$. Пусть v — висящая вершина дерева T и $T'=T-\{v\}$. Заметим, что T' — дерево на $n-1$ вершине.

Предположим, что v—X-вершина. Если $p=0$, то v — требуемая X-вершина дерева T. В противном случае по индуктивному предположению не менее p висячих вершин дерева T' являются X-вершинами. Эти p вершин висячие и в дереве T, и, следовательно, они вместе с вершиной v являются требуемыми X-вершинами дерева T.

Доказательство следует немедленно, если v—Y-вершина.

Лемма 9.2. Пусть (X, Y) — двудольное разбиение множества вершин дерева T.

1. Если $|X|=|Y|$ или $|X|=|Y|+1$, то существует чередующаяся последовательность $\{x_1, y_1, x_2, y_2, \ldots\}$, в которой каждая вершина дерева T используется точно один раз.
2. Если $|X|>|Y|+1$, то существует максимальная чередующаяся последовательность нечетной длины $\{x_1, y_1, \ldots\}$, в которой используется точно один раз каждая вершина подмножества Y.

Доказательство.

1. Результат справедлив для дерева на двух вершинах. Допустим, что результат справедлив для любого дерева на $2k$ вершинах. Тогда рассмотрим дерево T на $2k+1$ вершине, в котором $|X|=|Y|+1$. По предыдущей лемме в дереве T существует висячая вершина $x_{k+1}\in X$. Пусть $T'=T-\{x_{k+1}\}$. По индуктивному предположению в дереве T' существует чередующаяся последовательность $\{x_1, y_1, \ldots, y_k\}$, в которую входят все вершины этого дерева. Поэтому $\{x_1, y_1, \ldots, y_k, x_{k+1}\}$ будет требуемой последовательностью для дерева T. (Заметим, что эта последовательность нечетной длины.)

   Теперь допустим, что результат справедлив для любого дерева на $2k+1$ вершине. Рассмотрим дерево T на $2k+2$ вершинах, в котором $|X|=|Y|$. Тогда по предыдущей лемме в дереве T существует висячая вершина $y_{k+1}\in Y$. Пусть $T'=T-\{y_{k+1}\}$. По индуктивному предположению существует чередующаяся последовательность $\{x_1, y_1, \ldots, x_{k+1}\}$, в которую входят все вершины дерева T'. Поэтому $\{x_1, y_1, \ldots, x_{k+1}, y_{k+1}\}$ будет требуемой последовательностью для дерева T.
2. Если $|X|>|Y|+1$, удалим из дерева T столько висячих X-вершин, сколько необходимо для получения дерева T', в котором число X-вершин на 1 больше числа Y-вершин. По предыдущей лемме это всегда возможно.

   Теперь, согласно результату части 1 леммы, существует максимальная чередующаяся последовательность нечетной длины, в которую входят все вершины дерева T', а следовательно, все Y-вершины дерева T.

   Например, в дереве на рис. 9.2 $\{x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5, y_5, x_6, y_6, x_7\}$ — чередующаяся последовательность, в которую входят все Y-вершины дерева T.

Теперь сформулируем и докажем теорему, характеризующую максимальные независимые множества в виде чередующихся последовательностей,

Теорема 9.3. Независимое множество Y графа G максимально тогда и только тогда, когда в графе G не существует чередующейся последовательности нечетной длины по отношению к множеству Y.

Доказательство. Необходимость. Рассмотрим граф $G=(V, E)$. Пусть Y — максимальное независимое множество в графе G и $X=V-Y$.

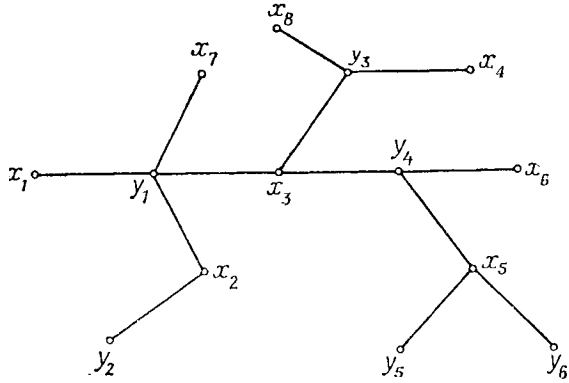


Рис. 9.2.

Допустим, что существует максимальная чередующаяся последовательность σ нечетной длины по отношению к множеству Y и $\sigma=\sigma_1 \cup \sigma_2$, где $\sigma_2=\sigma \cap Y$, а $\sigma_1=\sigma \cap X$.

Заметим, что, поскольку последовательность σ имеет нечетную длину, $|\sigma_1| > |\sigma_2|$ и последняя вершина последовательности σ является X-вершиной. Кроме того, поскольку σ — максимальна, никакую вершину $y_j \in Y-\sigma$ нельзя добавить в последовательность σ без нарушения условия (2) определения чередующейся последовательности.

Таким образом, никакая вершина $y_j \in Y-\sigma_2$ не смежна ни с какой вершиной $x_i \in \sigma_1$. Из того, что σ_1 и $Y-\sigma_2$ сами являются независимыми множествами, следует, что $(Y-\sigma_2) \cup \sigma_1$ — независимое множество. Поэтому $|(Y-\sigma_2) \cup \sigma_1| > |(Y-\sigma_2)| + |\sigma_2| = |Y|$, что противоречит максимальности независимого множества Y.

Достаточность. Пусть X — максимальное независимое множество, а Y — независимое множество с $|Y|<|X|$. Покажем, что существует чередующаяся последовательность нечетной длины по отношению к множеству Y.

Пусть $Y_0=Y-(X \cap Y)$, $X_0=X-(X \cap Y)$, а G_0 — порожденный подграф графа G на множестве вершин $(X_0 \cup Y_0)$. Если G_0 несвязный, тогда пусть $G_1, G_2, \ldots, G_k$ будут компонентами подграфа G_0, причем множество вершин G_i есть $(X_i \cup Y_i)$. Заметим, что $\bigcup_{i=1}^{k} X_i = X_0$ и $\bigcup_{i=1}^{k} Y_i = Y_0$. Так как $|X|>|Y|$, то $|X_0|>|Y_0|$. Поэтому для некоторого i существует неравенство $|X_i|>|Y_i|$. Не нарушая общности, допустим, что $i=1$.

Пусть теперь T — остов графа G_1. Очевидно, что (X_1, Y_1) является двудольным разбиением множества вершин остова T. Поскольку $|X_1|>|Y_1|$, по предыдущей лемме в графе G_1 существует максимальная чередующаяся последовательность σ (по отношению к Y_1) нечетной длины, в которой используются все вершины Y_1. Очевидно, что σ будет чередующейся последовательностью нечетной длины также и по отношению к Y в графе G. Покажем теперь, что σ — наибольшая такая последовательность в графе G.

Заметим, что X- и Y-вершины последовательности σ принадлежат X_0 и Y_0 соответственно. Рассмотрим произвольную Y-вершину y_j, не входящую в последовательность σ. Возможны два случая.

Случай 1. $y_j \in Y \cap X$.
В этом случае вершина y_j не смежна с X-вершинами. В частности, она не смежна ни с одной X-вершиной последовательности σ.

Случай 2. $y_j \in Y_0$.
В этом случае вершина y_j не принадлежит графу G_1, поскольку последовательность σ содержит все Y-вершины графа G_1. Следовательно, вершина y_j не смежна ни с одной вершиной графа G_1. В частности, она не смежна ни с какой X-вершиной последовательности σ.

В обоих случаях вершину y_j нельзя использовать для расширения последовательности σ. Поэтому в графе G σ — максимальная чередующаяся последовательность нечетной длины по отношению к Y.

Рассмотрим, например, независимое множество $Y=\{a, c\}$ в графе на рис. 9.1. Оно не максимально, и можно доказать, что $\{d, c, b, a, f\}$ — максимальная чередующаяся последовательность нечетной длины по отношению к Y.

9.2. Реберные покрытия

Рассмотрим граф $G=(V, E)$. Подмножество $P \subseteq E$ называется *реберным покрытием* графа G, если любая вершина последнего является концевой хотя бы одного из ребер P. Если рассматривать ребро как покрывающее свои концевые вершины, тогда реберное покрытие есть подмножество ребер, покрывающих все вершины графа G.

Реберное покрытие P графа G *минимально*, если он не имеет такого реберного покрытия P', что $|P'|<|P|$. Число ребер в минимальном реберном покрытии графа G называется *числом реберного покрытия* этого графа и обозначается через $\beta_1(G)$. Например, в графе на рис. 9.1 множество $\{e_1, e_3, e_5\}$ образует минимальное реберное покрытие.

Напомним, что числа вершинного и реберного покрытий обозначаются через β_0 и β_1 соответственно; аналогично числа независимости и паросочетания обозначаются через α_0 и α_1 соответственно.

Далее мы используем β_1 для обозначения также и порожденного подграфа на ребрах покрытия.

Предположим, что реберное покрытие P минимально. Тогда легко убедиться в том, что P не имеет циклов и путей длины более 2. Это означает, что каждая компонента покрытия P есть дерево, в котором все ребра инцидентны общей вершине.

В следующей теореме, являющейся реберным аналогом следствия 9.1.1, мы связываем величины α_1 и β_1. Этот результат получен Галлаи [9.3].

Теорема 9.4. Для простого графа G на n вершинах без изолированных вершин $\alpha_1+\beta_1=n$.

Доказательство. Пусть M^* — максимальное паросочетание, P^* — минимальное реберное покрытие G. Если $N(M^*)$ обозначает множество вершин, не насыщенных в M^*, то $|N(M^*)|=n-2\alpha_1$. Теперь для каждой вершины v в $N(M^*)$ выберем ребро, инцидентное вершине v и смежное с ребром из M^*. Пусть P_a — множество из $(n-2\alpha_1)$ выбранных таким образом вершин. Тогда очевидно, что множество $P=M^* \cup P_a$ является вершинным покрытием и $|P|=\alpha_1+n-2\alpha_1=$

$=n-\alpha_1$. Таким образом, $\beta_1=|P^*|\leqslant|P|=n-\alpha_1$, т. е.

$$\alpha_1+\beta_1\leqslant n. \tag{9.3}$$

Пусть теперь P^* имеет r связных компонент, т. е. $\beta_1=|P^*|=n-r$. Выберем по одному ребру из каждой компоненты P^*. Пусть M будет множеством из r выбранных ребер. Очевидно, что M является паросочетанием и поэтому $\alpha_1=|M^*|\geqslant|M|=r=n-\beta_1$. Таким образом,

$$\alpha_1+\beta_1\geqslant n. \tag{9.4}$$

Объединяя выражения (9.3) и (9.4), получаем $\alpha_1+\beta_1=n$.

Рассмотрим максимальное независимое множество S^* и минимальное реберное покрытие P^* графа G, не имеющего изолированных вершин. Так как для покрытия вершин множества S^* требуется $|S^*|$ ребер, любое реберное покрытие должно содержать по крайней мере $|S^*|$ ребер.

Таким образом,

$$|S^*|\leqslant|P^*|. \tag{9.5}$$

В общем случае равенство в выражении (9.5) не имеет места. Однако, когда граф G двудольный, $|S^*|=|P^*|$; это равенство мы докажем в следующей теореме, аналогичной теореме 9.2.

Теорема 9.5. Число вершин максимального независимого множества в двудольном графе без изолированных вершин равно числу ребер минимального реберного покрытия, т. е. $\alpha_0=\beta_1$.

Доказательство. Пусть граф G имеет n вершин. Тогда, согласно следствию 9.1.1, $\alpha_0=n-\beta_0$, а по теореме 9.4 $\beta_1=n-\alpha_1$. Но по теореме 9.2 $\alpha_1=\beta_0$. Поэтому получаем $\alpha_0=\beta_1$.

Теоремы 9.2 и 9.5 являются эквивалентными формулировками теоремы Холла (упражнения 9.5 и 9.6).

9.3. Реберная раскраска и хроматический индекс

Реберной k-раскраской графа называется присвоение ребрам графа k различных цветов. Реберная раскраска называется *правильной*, если никакие два смежных ребра не получают в ней одинакового цвета. Реберная k-раскраска и правильная реберная k-раскраска графа представлены на рис. 9.3. Граф называется *реберно-k-раскрашиваемым*, если он имеет правильную реберную k-раскраску.

Не нарушая общности, допустим, что графы, рассматриваемые в этом разделе, не имеют петель.

Хроматический индекс или *реберное хроматическое число* $\chi'(G)$ графа G — это минимальное число k, для которого граф G имеет правильную реберную k-раскраску. Граф G называется *реберно k-хроматическим*, если $\chi'(G)=k$. Хроматический индекс графа на рис. 9.3, *а* равен 4.

Легко видеть, что реберная k-раскраска графа $G=(V, E)$ влечет разбиение $(E_1, E_2, \ldots, E_k)$ множества E, где E_i — подмножество ребер, которым присвоен в раскраске цвет i. Аналогично любое разбиение множества E на k подмножеств соответствует реберной k-рас-

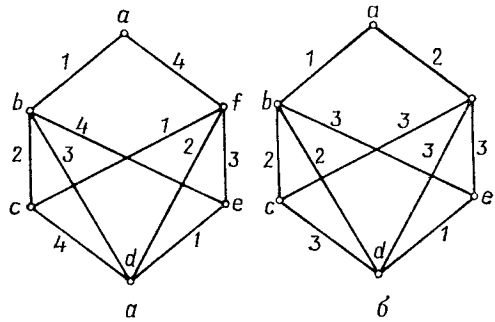


Рис. 9.3. Реберная раскраска (числа помечают цвета).
а — правильная реберная 4-раскраска; *б* — реберная 3-раскраска.

краске графа G. Поэтому мы часто будем обозначать реберную раскраску соответствующим разбиением графа G.

Если раскраска $\mathscr{E}=(E_1, E_2, \ldots, E_k)$ правильная, то каждое E_i является паросочетанием. Поэтому $\chi'(G)$ можно рассматривать как наименьшее число паросочетаний, на которые разбивается множество ребер графа G. Эта интерпретация $\chi'(G)$ будет полезной при доказательстве определенных результатов.

Поскольку в любой правильной раскраске ребра, инцидентные одной вершине, получают разные цвета, то

$$\chi'(G) \geqslant \Delta, \tag{9.6}$$

где Δ — максимальная степень в графе G. В общем случае $\chi'(G) \neq \Delta$, однако в случае двудольных графов $\chi' = \Delta$.

Теорема 9.6. Для двудольного графа $\chi'(G)=\Delta$.

Доказательство. Согласно следствию 8.22.1, множество ребер двудольного графа G можно разбить на Δ паросочетаний. Следовательно, $\chi'(G) \leqslant \Delta$. Объединение этого неравенства с выражением (9.6) доказывает теорему.

Хотя в общем случае Δ цветов недостаточно для того, чтобы правильно раскрасить ребра графа, Визинг [9.4] показал, что для любого простого графа достаточно $\Delta+1$ цветов. Для доказательства теоремы Визинга нам необходимо установить несколько основных результатов.

Будем говорить, что цвет i представлен в раскраске в вершине v, если хотя бы одному ребру, инцидентному вершине v, присвоен цвет i. Число различных цветов, представленных в вершине v, будем обозначать через $c(v)$. Например, в раскраске, показанной на рис. 9.3, *б*, $c(f)=2$.

Теперь рассмотрим вопрос о существовании реберной 2-раскраски графа, в котором $c(v)=2$ для каждой вершины v графа.

Лемма 9.3. Пусть G — связный граф, не имеющий циклов нечетной длины. Тогда он имеет реберную 2-раскраску, в которой оба цвета представлены в каждой вершине степени не менее 2.

Доказательство.

Случай 1. Граф G — эйлеров.

Если G — цикл, тогда по условиям теоремы он должен быть четной длины. В этом случае легко видеть, что граф G имеет реберную 2-раскраску, обладающую требуемым свойством.

Если G — не цикл, тогда он должен иметь вершину v_0 степени не менее 4. Пусть $v_0, e_1, v_1, e_2, v_2, e_3, \ldots, e_m, v_0$ — эйлерова цепь и

$$E_1=\{e_i \mid i \text{ нечетное}\}, \text{ а } E_2=\{e_i \mid i \text{ четное}\}. \tag{9.7}$$

Тогда (E_1, E_2) является реберной 2-раскраской, в которой в каждой вершине представлены оба цвета, поскольку в рассматриваемой цепи каждая вершина, включая вершину v_0, появляется в качестве внутренней вершины.

Случай 2. Граф G — не эйлеров.

В этом случае граф G должен иметь четное число нечетных вершин. Построим сейчас эйлеров граф G', добавляя новую вершину v_0 и соединяя ее в графе G с каждой вершиной нечетной степени. Пусть $v_0, e_1, v_1, \ldots, e_m, v_0$ — эйлерова цепь в графе G'. Если E_1 и E_2 определены, как в выражении (9.7), то (E_1, E_2) есть реберная 2-раскраска графа G', в котором оба цвета представлены в каждой вершине графа G. Сейчас легко видеть, что $(E_1 \cap E, E_2 \cap E)$ является реберной 2-раскраской графа G, в котором оба цвета представлены в каждой вершине степени не менее 2.

Реберная k-раскраска $\mathcal{E}$ графа $G=(V, E)$ называется *оптимальной*, если не существует другой такой реберной k-раскраски $\mathcal{E}'$ графа G, что

$$\sum_{v \in V} c(v) < \sum_{v \in V} c'(v),$$

где $c(v)$ и $c'(v)$ — числа различных цветов, представленных в вершине v в раскрасках $\mathcal{E}$ и $\mathcal{E}'$ соответственно.

Очевидно, что в любой раскраске $\mathcal{E}$ $c(v) \leqslant d(v)$ для каждой вершины v. Более того, для каждой вершины v равенство $c(v)=d(v)$ справедливо тогда и только тогда, когда $\mathcal{E}$ — правильная раскраска.

Следующий полезный результат раскрывает природу оптимальной раскраски.

Лемма 9.4. Пусть $\mathcal{E}=(E_1, E_2, \ldots, E_k)$ — оптимальная k-раскраска графа $G=(V, E)$. Допустим, что в графе G существуют вершина u и такие два цвета i и j, что i не представлен в u, а j представлен в u дважды. Пусть G' — порожденный подграф графа G на множестве ребер $E_i \cup E_j$. Тогда компонента H графа G', содержащая вершину u, является циклом нечетной длины.

Доказательство. Если H не является циклом нечетной длины, то, согласно лемме 9.3, существует реберная 2-раскраска, в которой цвета i и j представлены в каждой вершине H степени не менее 2.

Если мы используем эту реберную 2-раскраску для перекраски ребер компоненты H и оставим цвета других ребер графа G без изменения, тогда получим новую реберную k-раскраску $\mathcal{E}'$, в которой цвета i и j будут представлены в

вершине u. Следовательно, $c'(u)=c(u)+1$, где $c'(u)$ — число различных цветов, представленных в вершине u в раскраске $\mathcal{E}'$. Кроме того, для каждой вершины $v\neq u$, $c'(v)\geqslant c(v)$. Следовательно, $\sum\limits_{v\in V} c'(v) > \sum\limits_{v\in V} c(v)$, что противоречит оптимальности реберной раскраски $\mathcal{E}$. Таким образом, H — цикл нечетной длины.

Реберная k-раскраска $\mathcal{E}'$ называется *улучшением* реберной k-раскраски $\mathcal{E}$, если $\sum\limits_{v\in V} c'(v) > \sum\limits_{v\in V} c(v)$.

Сейчас мы докажем теорему Визинга [9.4]. Используемое нами доказательство предложено Фурнье [9.5]. В его изложении будем следовать Бонди и Мурти [9.6].

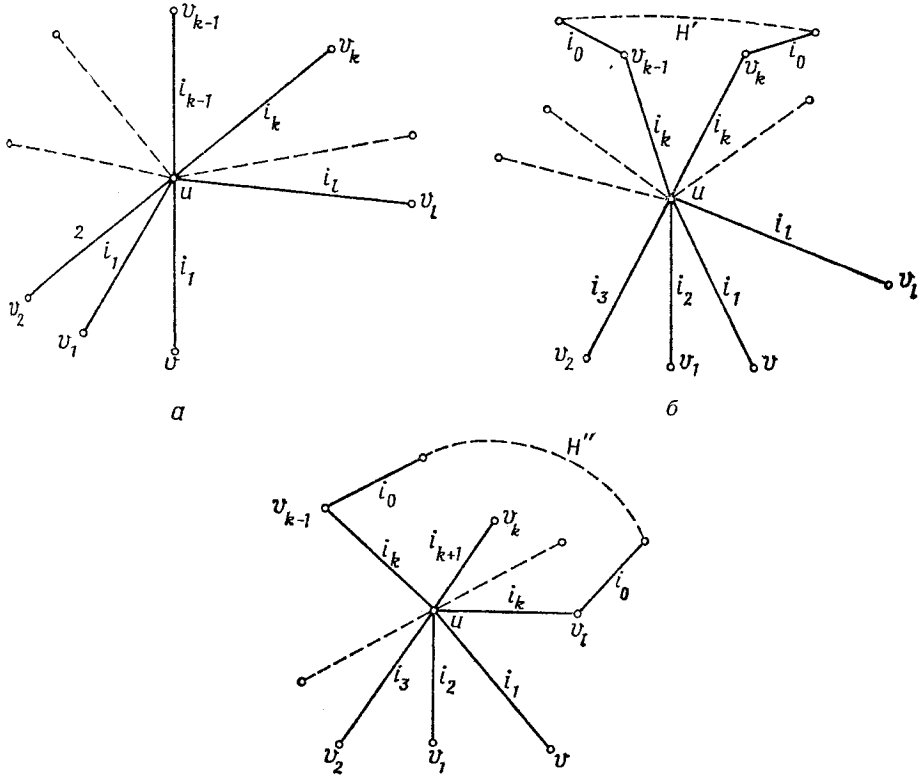


Рис. 9.4.

Теорема 9.7 (Визинг). Если $G=(V, E)$ — простой граф, то либо $\chi'(G)=\Delta$, либо $\chi'(G)=\Delta+1$.

Доказательство. Поскольку $\chi'(G)\geqslant\Delta$, будет достаточно, если мы докажем, что $\chi'(G)\leqslant\Delta+1$.

Пусть $\varepsilon=\{E_1, E_2, \ldots, E_{\Delta+1}\}$ — оптимальная реберная $(\Delta+1)$-раскраска графа G. Допустим, что $\chi'(G)>\Delta+1$, т. е. $\mathcal{E}$ — неправильная реберная $(\Delta+1)$-раскраска. Тогда в графе G существует такая вершина u, что $c(u)<d(u)$. Отсюда

следует, что существуют такие цвета i_0 и i_1, что i_0 не представлен в вершине u, а i_1 представлен в ней дважды. Пусть ребро (u, v_1) имеет цвет i_1 (рис. 9.4, *а*).

Поскольку максимальная степень графа G равна Δ, то в раскраске $\mathcal{E}$ представлены не все $\Delta+1$ цветов. Пусть в ней не представлен цвет i_2. Тогда i_2 должен быть представлен в вершине u, иначе мы можем перекрасить ребро (u, v_1) в цвет i_2 и получить новую раскраску графа G, являющуюся улучшением раскраски $\mathcal{E}$. Пусть ребро (u, v_2) имеет цвет i_2. Снова не все цвета представлены в вершине v_2.

Предположим, что в вершине v_2 не представлен цвет i_3. Тогда он должен быть представлен в вершине u, в противном случае мы можем перекрасить ребро (u, v_1) в цвет i_2, а ребро (u, v_2) — в i_3, получая улучшение раскраски $\mathcal{E}$. Пусть ребро (u, v_3) имеет цвет i_3.

Повторяя эти рассуждения, можно построить такие последовательности вершин $v_1, v_2, \ldots$ и цветов $i_1, i_2, \ldots$, что 1) (u, v_j) имеет цвет i_j; 2) i_{j+1} не представлен в вершине v_j.

Легко видеть, что существует такое наименьшее целое число l, что для некоторого $k<l$ справедливо равенство $i_{l+1}=i_k$ (рис. 9.4, *а*).

Для получения противоречия перекрасим теперь двумя разными способами некоторые ребра, инцидентные u, и получим две новые оптимальные реберные $(\Delta+1)$-раскраски: $\mathcal{E}'$ и $\mathcal{E}''$.

Раскраска $\mathcal{E}'=\{E'_1, E'_2, \ldots, E'_{\Delta+1}\}$ получается перекраской ребра (u, v_j) в цвет i_{j+1} $(1\leqslant j\leqslant k-1)$. Каждое из оставшихся ребер графа G получает в раскраске $\mathcal{E}'$ тот же цвет, что и в раскраске ε (рис. 9.4, *б*).

Раскраска $\mathcal{E}''=\{E''_1, E''_2, \ldots, E''_{\Delta+1}$ получается перекраской ребра (u, v_j) в цвет i_{j+1} $(1\leqslant j\leqslant l-1)$, а ребра (u, v_l) — в цвет i_k. Каждое из оставшихся ребер графа G получает в раскраске $\mathcal{E}''$ тот же цвет, что и в раскраске $\mathcal{E}$ (рис. 9.4, *в*).

Пусть G' — подграф графа G на множестве ребер $(E'_{i_0} \cup E'_{i_k})$, а H' — его компонента, содержащая вершину u. Аналогично G'' — подграф графа G на множестве ребер $(E''_{i_0} \cup E''_{i_k})$, а H'' — его компонента, содержащая вершину u.

Как в раскраске $\mathcal{E}'$, так и в $\mathcal{E}''$ цвет i_0 не представлен в вершине u, а цвет i_k представлен в этой вершине дважды. Поэтому из леммы 9.4 следует, что H' и H'' — циклы.

Единственным ребром цикла H', которого нет в цикле H'', является ребро (u, v_k). Поэтому вершины u и v_k связаны также и в подграфе G''. Следовательно, они лежат в одной и той же компоненте подграфа G'', а именно в компоненте H''. Тогда степень v_k в H'' равна 1, иначе бы степень ее в H' была больше 2. Это противоречит тому, что H'' — также цикл. Таким образом, $\mathcal{E}$ — правильная реберная $(\Delta+1)$-раскраска.

Более общий результат по сравнению с выводом этой теоремы можно найти в упражнении 9.8.

9.4. Вершинная раскраска и хроматическое число

Вершинная k-раскраска графа — это присвоение его вершинам k различных цветов. Вершинная раскраска *правильная*, если никакие две смежные вершины в ней не получают одного цвета. Правильная вершинная 3-раскраска представлена на рис. 9.5. Граф называется *вершинно k-раскрашиваемым*, если он имеет правильную вершинную k-раскраску.

Не нарушая общности, мы можем допустить, что графы, рассматриваемые в этом разделе, простые.

Хроматическим числом $\chi(G)$ графа G называется минимальное число k, для которого G вершинно k-раскрашиваемый. Граф G называется *k-хроматическим*, если $\chi(G)=k$. Например, на рис. 9.5 хроматическое число графа равно 3.

Далее вместо «правильная вершинная k-раскраска» будем говорить просто «k-раскраска». Аналогично «вершинно k-раскрашиваемый» будем заменять на «k-раскрашиваемый».

Заметим, что k-раскраска графа $G=(V, E)$ порождает разбиение $(V_1, V_2, \ldots, V_k)$ множества V, где каждое V_i — подмножество вершин, которым присвоен цвет i и поэтому является независимым множеством. Аналогично каждое разбиение множества V на k независимых множеств соответствует k-раскраске графа G.

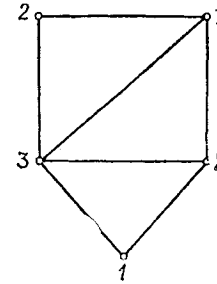


Рис. 9.5. Правильная вершинная 3-раскраска.

В предыдущем разделе (теорема 9.7) мы доказали, что для правильной раскраски ребер простого графа требуется, самое большее, $\Delta+1$ цветов. Докажем аналогичный результат для вершинной раскраски.

Теорема 9.8. Простой граф G—$(\Delta+1)$-раскрашиваемый.

Доказательство. Даны $\Delta+1$ различных цветов, получим $(\Delta+1)$-раскраску графа G следующим образом.

Возьмем произвольную вершину v_0 и присвоим ей любой из $\Delta+1$ цветов. Затем выберем нераскрашенную вершину, например v_1. Присвоим вершине v_1 цвет, который не был присвоен смежным с ней вершинам. Это всегда возможно, поскольку $d(v_1)\leqslant\Delta$ и, следовательно, вершинам, смежным с вершиной v_1, будет присвоено, самое большее, Δ цветов. Повторим этот процесс до тех пор, пока не раскрасятся все вершины. В результате получится правильная $(\Delta+1)$-раскраска.

Очевидно, что $\chi=\Delta+1$ для полных графов и циклов нечетной длины. Очень интересно то, что для всех остальных графов $\chi\leqslant\Delta$. Этот результат получен Бруксом [9.7] и доказывается ниже. Приводимое здесь доказательство предложили Мельников и Визинг [9.8]. Другое доказательство можно найти в работе [9.9].

Теорема 9.9 (Брукс). Пусть G — связный простой граф. Если он не цикл нечетной длины и не полный граф, то $\chi(G)\leqslant\Delta$.

Доказательство. Теорема очевидна для $\Delta=0, 1, 2$.

Чтобы доказать теорему для $\Delta\geqslant 3$, допустим противное, т. е. что существуют графы, не являющиеся полными, для которых $\Delta\geqslant 3$ и $\chi=\Delta+1$. Выберем такой граф $G=(V, E)$ с минимальным числом вершин.

Пусть $v_0\in V$, а G' — граф, полученный с результате удаления из графа G вершины v_0. Из выбора графа G следует, что G' — Δ-раскрашиваемый. Следовательно, $d(v_0)\geqslant\Delta$, иначе бы для раскраски вершины v_0 можно было использовать один из Δ цветов, применяемых для раскраски графа G', что противоречит равенству $\chi(G)=\Delta+1$. Другим важным следствием является следующее:

Свойство 1. В любой Δ-раскраске графа G' вершины, смежные с вершиной v_0, раскрашиваются по-разному

Пусть $u_1, u_2, \ldots, u_\Delta$ — вершины, смежные с вершиной v_0. Пусть $u_1, u_2, \ldots, u_\Delta$ получают в раскраске графа G' цвета 1, 2, . . ., Δ соответственно. Обо-

значим через $G(i, j)$ порожденный подграф G' на вершинах, которым присвоены цвета i и j.

Свойство 2. Вершины u_i и u_j находятся в одной компоненте связности $G(i, j)$.

В противном случае, заменяя цвета i и j в компоненте, содержащей u_i, мы получим новую Δ-раскраску графа G', в которой u_i и u_j присвоен одинаковый цвет, что противоречит свойству 1.

Пусть C_{ij}— компонента $G(i, j)$, содержащая вершины u_i и u_j.

Свойство 3. C_{ij}— путь от вершины u_i к вершине u_j.

Предположим, что степень u_i в C_{ij} больше 1. Тогда u_i смежна не менее чем с двумя вершинами цвета j. Поскольку $d(u_i) \leqslant \Delta - 1$ в графе G', мы можем перекрасить u_i в цвет $k \neq i, j$, так что в получившейся новой раскраске вершины u_i и u_j имеют одинаковый цвет, что противоречит свойству 1.

Аналогично можно показать, что степень u_j в C_{ij} равна 1.

Степень всех остальных вершин C_{ij} равна 2. Допустим противное. Пусть u — первая вершина степени (в C_{ij}) больше 2 на пути от вершины u_i к вершине u_j. Если u раскрашена в цвет i, то она смежна по крайней мере с тремя вершинами цвета j. Поскольку $d(u) \leqslant \Delta$, можно перекрасить u в цвет $k \neq i, j$, поэтому в новой раскраске вершины u_i и u_j будут в разных компонентах, что противоречит свойству 2.

Таким образом, C_{ij} является путем из вершины u_i в вершину u_j.

Свойство 4. C_{ij} и C_{ik} не имеют общих вершин, за исключением u_i.

Пусть $u \neq u_i$ — общая вершина C_{ij} и C_{ik}. Тогда u раскрашена в цвет i и смежна по крайней мере с двумя вершинами цвета j и двумя вершинами цвета k. Так как $d(u) \leqslant \Delta$, существует цвет $l \neq i, j, k$, в который можно перекрасить u. Но это разделит вершины u_i и u_j, что противоречит свойству 2.

Продолжим доказательство теоремы. Установим теперь противоречие со свойством 4.

Поскольку G — не полный граф на $\Delta+1$ вершине, существуют две несмежные вершины, например u_1 и u_2. Путь C_{12} содержит вершину $u \neq u_2$, смежную с вершиной u_1. Допустим, что мы поменяли местами цвета 1 и 3 в пути C_{13} (который присутствует, поскольку $\Delta \geqslant 3$), в результате чего в новой раскраске G' вершина u_1 получает цвет 3, а вершина u_3 — цвет 1. Но тогда новые компоненты C'_{12} и C'_{23} содержат общую вершину $u \neq u_2$, что противоречит свойству 4.

Это завершает доказательство теоремы.

9.5. Хроматические полиномы

В этом разделе мы обсуждаем вопрос подсчета различных правильных λ-раскрасок графа.

Если граф λ-раскрашиваемый, то его можно раскрасить в λ цветов более чем одним способом. Две раскраски графа считаются различными, если хотя бы одной вершине графа присваиваются разные цвета.

Хроматический полином $P(G, \lambda)$ имеет значение для каждого целого λ, равное числу различных правильных λ-раскрасок графа G.

Рассмотрим, например, граф, представленный на рис. 9.6. Среди данных λ цветов мы можем выбрать любой для раскраски вершины a. Вершину b можно раскрасить в один из оставшихся $\lambda-1$ цветов. Для каждой раскраски вершины b существует $\lambda-1$ различных способов раскраски вершины c. Таким образом, граф на рис. 9.6 можно раскрасить $\lambda(\lambda-1)^2$ различными способами. Другими сло-

вами, хроматический полином графа равен $\lambda(\lambda-1)^2$. Мы можем повторить эти рассуждения, чтобы показать, что хроматический полином пути на n вершинах равен $\lambda(\lambda-1)^{n-1}$.

В качестве другого примера рассмотрим полный граф K_n на n вершинах $v_1, v_2, \ldots, v_n$. При наличии λ цветов вершину v_1 можно

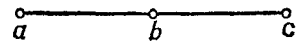


Рис. 9.6.

раскрасить в любой из них, вершину v_2 — в любой из оставшихся $\lambda-1$ цветов, а вершину v_3 — в любой из оставшихся $\lambda-2$ цветов и т. д. Таким образом, $P(K_n, \lambda)=\lambda(\lambda-1)\ldots(\lambda-n+1)$. Теперь выведем формулу для определения хроматического полинома графа G.

Теорема 9.10. Пусть u и v — несмежные вершины простого графа G. Пусть $e=(u, v)$. Если $G\cdot e$ — простой граф, полученный из графа G замыканием вершин u и v и заменой получившегося множества параллельных ребер на одно ребро, а $G+e$ — граф, полученный добавлением к графу G ребра e, то $P(G, \lambda)=P(G+e, \lambda)+P(G\cdot e, \lambda)$.

Доказательство. Любая λ-раскраска графа G, в которой вершинам u и v присваиваются различные цвета, соответствует λ-раскраске графа $G+e$, и наоборот. Аналогично любая λ-раскраска графа G, в которой вершинам u и v присвоен один цвет, соответствует λ-раскраске графа $G\cdot e$, и наоборот. Следовательно, $P(G, \lambda)=P(G+e, \lambda)+P(G\cdot e, \lambda)$.

Этот результат можно сформулировать в другой форме.

Следствие 9.10.1. Если $e=(u, v)$ — ребро простого графа G, то $P(G, \lambda)=P(G-e, v)-P(G\cdot e, \lambda)$, где $G-e$ получается из графа G удалением ребра e, а $G\cdot e$ определяется из теоремы 9.10.

Если мы повторно применим формулу, приведенную в теореме 9.10, к графу G, то процесс закончится на полных графах, например $H_1, H_2, \ldots, H_k$, поэтому $P(G, \lambda)=P(H_1, \lambda)+P(H_2, \lambda)+\ldots+P(H_k, \lambda)$.

С другой стороны, если мы используем формулу, приведенную в следствии 9.10.1, процесс закончится на пустых графах (т. е. графах, не имеющих ребер), поэтому хроматический полином есть линейная комбинация хроматических полиномов пустых графов.

Обе процедуры иллюстрируются рис. 9.7.

Теорема 9.11. Хроматический полином $P(G, \lambda)$ графа n на вершинах имеет степень n с главным членом λ^n и константой, равной 0. Кроме того, все коэффициенты целые и чередуются по знаку.

Доказательство. Доказательство проведем индукцией по числу ребер m. Очевидно, что теорема справедлива для $m=0$, так как хроматический полином пустого графа на n вершинах равен λ^n.

Допустим, что теорема верна для всех графов, имеющих менее m ребер. Рассмотрим граф G на n вершинах с m ребрами. Пусть e — ребро графа G. Тогда $G-e$ — граф на n вершинах с $m-1$ ребрами, а $G\,e$ — граф на $n-1$ вершинах с $m-1$ или менее ребрами.

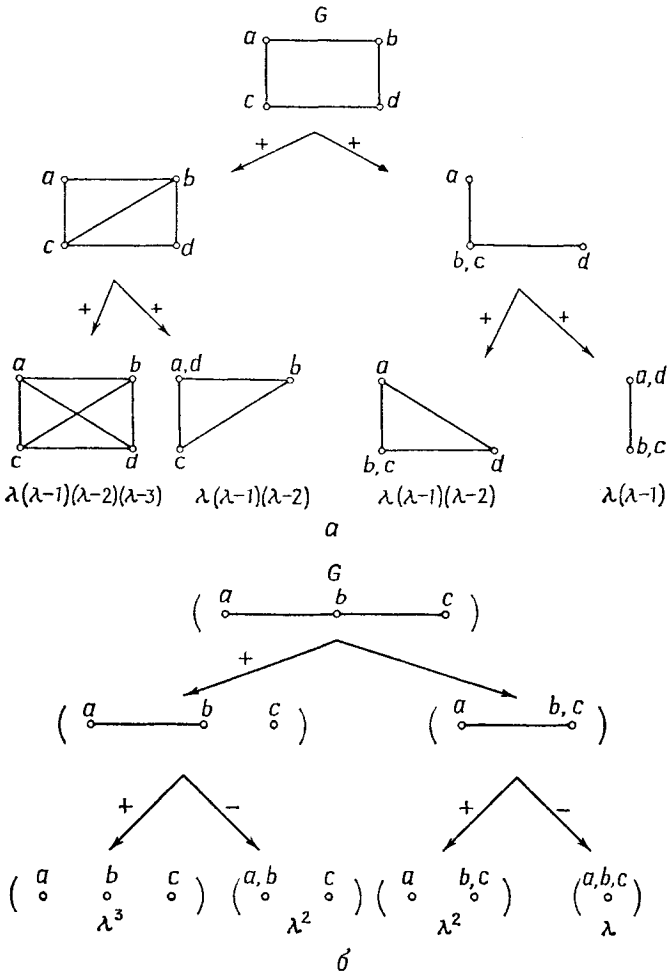


рис. 9.7. Вычисление хроматического полинома графа G.

а — с использованием теоремы 9.10; $P(G, \lambda)=\lambda(\lambda-1)(\lambda-2)(\lambda-3)+2\lambda(\lambda-1)(\lambda-2)+\lambda(\lambda-1)$; *б* — с использованием следствия 9.10.1; $P(G, \lambda)=\lambda^3-2\lambda^2+\lambda$.

Из индуктивного предположения следует, что существуют такие неотрицательные целые коэффициенты $a_1, a_2, \ldots, a_{n-1}$ и $b_1, b_2, \ldots, b_{n-2}$, что $P(G-e, \lambda)=\lambda^n-a_{n-1}\lambda^{n-1}+a_{n-2}\lambda^{n-2}-\ldots+(-1)^{n-1}a_1\lambda$ и $P(G\cdot e, \lambda)=\lambda^{n-1}-b_{n-2}\lambda^{n-2}+b_{n-3}\lambda^{n-3}-\ldots+(-1)^{n-2}b_1\lambda$. Согласно следствию 9.10.1, $P(G, \lambda)=P(G-e, \lambda)-P(G\cdot e, \lambda)=\lambda^n-(a_{n-1}+1)\lambda^{n-1}+\sum_{i=2}^{n-1}(-1)^i(a_{n-i}+b_{n-i})\lambda^{n-i}$. Таким образом граф G также удовлетворяет теореме.

9.6. Проблема четырех красок

При изготовлении карт в целях различения отдельных областей необходимо их раскрашивать таким образом, чтобы никакие две смежные области не были раскрашены одинаково. Эта задача правильной раскраски областей планарного графа равносильна задаче правильной раскраски вершин двойственного графа. Легко привести пример, показывающий, что трех цветов в общем случае недостаточно для правильной раскраски планарного графа. В следующей теореме, известной под названием *теоремы о пяти красках*, мы показываем, что пяти цветов достаточно.

Теорема 9.12. Любой планарный граф 5-раскрашиваемый.

Доказательство. Допустим, что теорема верна для всех планарных графов порядка менее n, покажем, что она верна также для планарного графа $G=(V, E)$ порядка n.

Согласно следствию 7.3.5, в графе G существует вершина v_0 степени не более 5. Пусть G' — порожденный подграф графа G на $V-\{v_0\}$.

Раскрасим подграф G' в пять цветов: $\alpha_1, \alpha_2, \ldots, \alpha_5$. (Это возможно в силу индуктивного предположения.) Очевидно, что если $d(v_0)\neq 5$, то вершине v_0 в

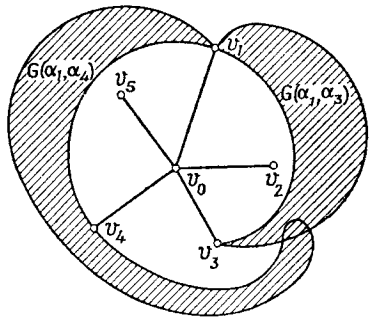


Рис. 9.8.

правильной 5-раскраске графа G можно присвоить один из этих цветов. В противном случае всем 5 вершинам $v_1, v_2, \ldots, v_5$, смежным с вершиной v_0, можно присвоить различные цвета.

Допустим, что вершине $v_i (1\leqslant i\leqslant 5)$ присвоен цвет α_i. Кроме того, пусть вершины $v_1, v_2, \ldots, v_5$ располагаются по часовой стрелке относительно вершины v_0, как показано на рис. 9.8.

Пусть $G(\alpha_i, \alpha_j)$ — подграф графа G на вершинах, которым присвоены цвета α_i или α_j.

Компонента $G(\alpha_1, \alpha_3)$, содержащая вершину v_1, должна содержать и вершину v_3, в противном случае цвета α_1 и α_3 в этой компоненте можно было бы поменять местами и раскрасить вершину v_0 в цвет α_1.

Аналогично компонента $G(\alpha_1, \alpha_4)$, содержащая вершину v_1, должна содержать и вершину v_4 (рис. 9.8).

Тогда в $G(\alpha_2, \alpha_5)$ вершины v_2 и v_5 не будут соединены путем. Поэтому если в компоненте $G(\alpha_2, \alpha_5)$, содержащей вершину v_2, поменять местами цвета α_2 и α_5, то вершину v_0 можно будет раскрасить в цвет α_2. Таким образом, граф G — 5-раскрашиваемый.

Этот результат был получен Хивудом [9.10].

Возникает вопрос: можно ли улучшить теорему о пяти красках? Предполагалось, что любой планарный граф 4-раскрашиваемый. Эта гипотеза известна как *гипотеза четырех красок*. Она оставалась неразрешенной более 100 лет и была предметом экстенсивных исследований. Недавно Аппель и Хакен [9.11] доказали справедливость гипотезы. Этот результат формулируется в следующей теореме:

Теорема 9.13 (теорема о четырех красках). Каждый планарный граф 4-раскрашиваемый.

Имеется очень обширная литература, касающаяся проблемы четырех красок. Обзор проблемы с историческими подробностями можно найти в работах [9.12—9.14]. См. также работу [9.15].

9.7. Замечания, касающиеся литературы

Общие ссылки на литературу, охватывающую тему данной главы,— это работы [9.2, 9.6, 9.16].

Классический результат теории графов, принадлежащий Турану [9.17], определяет, какое минимальное число ребер должен содержать граф на n вершинах с числом независимости не более k (упражнение 9.2). См. работу [9.18]. Этот результат является отправной точкой в ветви теории графов, называемой *теорией экстремальных графов*. Книга Болобаша [9.15] посвящена исключительно изучению задач, связанных с экстремальными графами. Характеристики экстремальных графов образуют основу проектирования сетей связи с заданными свойствами надежности и уязвимости. См. работы [9.20—9.23].

Применение числа независимости в теории информации можно найти в работах [9.2] (гл. 16), [9.24] (гл. 9), [9.25].

Рид [9.26] дал прекрасный обзор результатов по хроматическим полиномам. См. также работы [9.27—9.29]. Лю [9.24] обсуждает применение раскраски при разложении графов на планарные подграфы. Эта задача возникает при проектировании печатных плат. Применение раскраски к задачам составления расписания можно найти в работе [9.30], а алгоритм раскраски — в работе [9.31].

Множество вершин D графа G называется *доминирующим*, если любая вершина, не входящая в множество D, смежна с вершиной того же множества. Обзор результатов, касающихся доминирующих множеств, можно найти в работе [9.32]. Экстремальные результаты, касающиеся доминирующих множеств, даны в работе [9.33].

Упражнения

9.1. Доказать или опровергнуть: любое вершинное покрытие содержит минимальное вершинное покрытие.

9.2. Даны целые n и k, $n \geqslant k > 0$. Пусть $q = \lfloor n/k \rfloor$ и r — такое целое число, что $n = kq + r$, $0 \leqslant r < k$.

Пусть $G_{n,k}$ — простой граф, состоящий из k непересекающихся полных графов, r из которых имеют $q+1$ вершин, а остальные $k-r$ имеют q вершин. Покажите, что любой граф G на n вершинах с $\alpha_0(G) \leqslant k$, имеющий минимальное число ребер, изоморфен $G_{n,k}$ [9.17]. Легким следствием из этого результата является следующее:

Если G — простой граф на n вершинах и m ребрах с $\alpha_0(G)=k$, то $m \geqslant q(n-k/2-kq/2)$.

9.3. Покажите, что если G — простой граф на n вершинах с m ребрами, то $\alpha_0(G) \geqslant n^2/2(m+n)$.

9.4. Покажите, что максимальное число ребер, которые может иметь простой граф G на n вершинах, не содержащий треугольников, равно $\lfloor n^2/4 \rfloor$. Постройте граф с этим свойством.

9.5. Получите теорему Холла 8.13 из теоремы 9.2.

9.6. Получите теорему Холла 8.13 из теоремы 9.5.

9.7. Покажите, что

$$\chi'(K_n) = \begin{cases} n-1, & \text{если } n \text{ четно;} \\ n, & \text{если } n \text{ нечетно.} \end{cases}$$

9.8. Покажите, что для графа G без петель $\chi'(G) \leqslant \Delta + k$, где k — максимальное число параллельных ребер между любыми двумя вершинами графа G [9.4].

9.9. Пусть G — граф без петель. Покажите, что если $\Delta(G)=3$, то $\chi'(G)=3$ или 4.

9.10. Покажите, что для однородного непустого графа с нечетным числом вершин $\chi'(G)=\Delta+1$.

9.11. Покажите, что для произвольной ориентации ребер простого графа G существует ориентированный путь длиной $\chi(G)-1$.

9.12. Покажите, что если два любых цикла нечетной длины графа имеют общую вершину, то $\chi(G) \leqslant 5$.

9.13. Покажите, используя следствие 7.3.5, что любой планарный граф 6-раскрашиваемый.

9.14. Покажите, что если простой граф G имеет последовательность степеней $(d_1, d_2, \ldots, d_n)$, $d_1 \geqslant d_2 \geqslant \ldots \geqslant d_n$, то $\chi(G) \leqslant \max\limits_i \min\{d_i+1, i\}$ [9.34].

9.15. Покажите, что для простого графа G на n вершинах а) $\chi(G)\alpha_0(G) \geqslant n$; б) $\chi(G)+\alpha_0(G) \leqslant n+1$; в) $\chi(G)+\chi(\overline{G}) \leqslant n+1$.

9.16. Пусть G — простой однородный граф степени k на n вершинах. Покажите, что $\chi(\overline{G}) \geqslant n/(n-k)$.

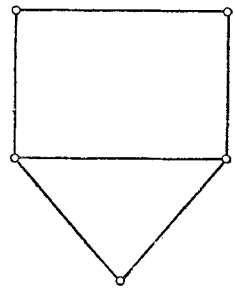


Рис. 9.9.

9.17. Пусть G — планарный граф, в котором каждая область ограничена точно тремя ребрами. Покажите, что граф G — 3-раскрашиваемый тогда и только тогда, когда граф G — эйлеров.

9.18. Покажите, что простой граф непланарен, если он имеет семь вершин и степень каждой вершины равна 4 [9.24].

9.19. а) Покажите, что граф 2-раскрашиваемый тогда и только тогда, когда все его циклы имеют четную длину.

б) Покажите, что области планарного графа G можно правильно раскрасить в два цвета тогда и только тогда, когда граф G — эйлеров.

9.20. Найти хроматический полином графа, представленного на рис. 9.9.

9.21. а) Покажите, что граф G на n вершинах является деревом тогда и только тогда, когда $P(G, \lambda)=\lambda(\lambda-1)^{n-1}$.

б) Покажите, что если граф G на n вершинах связный, тогда $P(G, \lambda)\leqslant\lambda(\lambda-1)^{n-1}$.

9.22. Покажите, что если G — цикл длины n, то $P(G, \lambda)=(\lambda-1)^n+(-1)^n(\lambda-1)$.

9.23. Покажите, что если G — колесо на $n+1$ вершинах, то $P(G, \lambda)=\lambda(\lambda-2)^n+(-1)^n\lambda(\lambda-2)$.

9.24. Покажите, что если простой граф G имеет n вершин, m ребер и k компонент, то

а) коэффициент при λ^{n-1} в P-полиноме (G, λ) равен $(-m)$;

б) наименьший показатель степени λ в полиноме $P(G, \lambda)$ с ненулевым коэффициентом равен k.

9.25. *Клика* графа $G=(V, E)$ есть такое множество $S\subseteq V$, что порожденный подграф G на S является полным графом. Пусть $P=\{P_1, P_2, \ldots, P_k\}$ — разбиение множества V на клики. Пусть $\theta(G)=\min\limits_{P}\{|P|\}$.

Таким образом, $\theta(G)$ есть наименьшее возможное число клик, на которые можно разбить множество V.

Покажите, что для простого графа G существует неравенство $\alpha_0(G)\leqslant\theta(G)$. Кроме того, покажите, что если S — независимое множество, а P — такое разбиение множества V на клики, что $|S|=|P|$, то S — максимальное независимое множество, а P — минимальное разбиение V на клики.

10. Матроиды

Рассмотрим конечное множество векторов S над произвольным полем. Хорошо известно, что любое подмножество S либо линейно зависимо, либо линейно независимо. Кроме того, набор независимых множеств векторов обладает некоторыми свойствами. Например:
1. Любое подмножество независимого множества независимо.
2. Если I_p и I_{p+1} — такие независимые множества, что $|I_{p+1}|=|I_p+1$, то I_p вместе с некоторым элементом I_{p+1} образует независимое множество из $|I_{p+1}|$ элементов.

Интересно, что существует несколько алгебраических систем, обладающих этими свойствами. Например, ими обладает набор подмножеств ребер графа, не содержащих циклов. Изучая свойства таких систем, Уитни [10.1] ввел понятие матроида.

В этой главе мы даем введение в теорию матроидов. Мы изучим их некоторые фундаментальные свойства. Особый интерес будет проявлен к тому, что замеченная нами «двойственность» циклов и разрезающих множеств графа не случайна. Как мы увидим, эта двойственность является следствием того, что и набор подграфов, не имеющих циклов, и набор подграфов, не имеющих разрезающих множеств, имеют структуру матроида. Мы изучим также теорему о раскрашивании и лемму о раскраске дуг, которые находят применение в анализе сетей. Главу мы завершаем обсуждением «жадного» алгоритма, являющегося обобщением известного алгоритма Краскала нахождения остова минимальной стоимости во взвешенном связном графе.

10.1. Основные определения

Существует несколько эквивалентных систем аксиом, характеризующих матроид. Мы начнем обсуждение с системы, известной под названием аксиом независимости. В разд. 10.3 мы приведем другие эквивалентные системы аксиом.

Матроид M — это конечное множество S и набор $\mathcal{F}$ таких подмножеств множества S, что выполняются следующие условия, называемые аксиомами независимости:

I.1. $\varnothing \in \mathcal{F}$.

I.2. Если $X \in \mathcal{F}$ и $Y \subseteq X$, то $Y \in \mathcal{F}$.

I.3. Если X и Y — члены $\mathcal{F}$ и $|X|=|Y|+1$, то существует такое $x \in X-Y$, что $Y \cup x \in \mathcal{F}$.

Элементы множества S называются *элементами матроида* M.

Члены набора $\mathcal{F}$ называются *независимыми множествами* матроида M. Максимальное по включению независимое множество матроида M называется *базой* матроида M. Множество баз матроида M обозначается $\mathcal{B}(M)$ или просто $\mathcal{B}$.

Подмножество S, не принадлежащее набору $\mathcal{F}$, называется *зависимым*. Минимальное по включению зависимое подмножество S называется *циклом* матроида M. Элемент $x \in S$ называется *петлей* матроида M, если $\{x\}$ зависимо. Набор циклов матроида M обозначается $\mathcal{C}(M)$ или просто $\mathcal{C}$.

Функция ранга ρ матроида M связывает со всяким подмножеством $A \subseteq S$ неотрицательное целое число, определяемое следующим образом: $\rho(A) = \max\{|X| : X \subseteq A,\ X \in \mathcal{F}\}$, где $\rho(A)$ — ранг подмножества A. *Ранг матроида* M, обозначаемый $\rho(M)$,— это ранг множества S.

Рассмотрим несколько примеров.

Пусть S — конечное подмножество векторного пространства. Как мы видели ранее, семейство всех подмножеств линейно независимых векторов в подмножестве S удовлетворяет аксиомам независимости I.1—I.3. Следовательно, эти подмножества S образуют набор независимых множеств матроида на подмножестве S. Ранг подмножества $X \subseteq S$ в этом матроиде равен размерности векторного пространства, порождаемого X.

Пусть G — неориентированный граф с множеством ребер E. Определим на E два матроида.

Сначала рассмотрим набор $\mathcal{F}$ всех подмножеств E, не содержащих циклов. Очевидно, что $\mathcal{F}$ удовлетворяет аксиомам I.1 и I.2. Нетрудно показать (упражнение 4.8), что $\mathcal{F}$ удовлетворяет и I.3. Таким образом, $\mathcal{F}$ является набором независимых множеств матроида M^* на E. Каждая база матроида M — это остовный лес графа G. Ранг любого подмножества $X \subseteq E$ в этом матроиде равен рангу подграфа графа G, порождаемого X. Кроме того, каждый цикл матроида M является циклом графа G. По этой причине M называется *циклическим матроидом* графа G.

Рассмотрим теперь семейство $\mathcal{F}^*$ всех подмножеств E, не содержащих разрезающих множеств графа G. Можно показать (упражнение 4.9), что $\mathcal{F}^*$ удовлетворяет аксиомам I.1—I.3 и поэтому является семейством независимых множеств матроида M^* на E. Любая база матроида M^* — это коостовный лес графа G. В этом матроиде ранг подмножества $X \subseteq E$ равен цикломатическому числу порождаемого X подграфа графа G. Кроме того, каждый цикл M^* является разрезающим множеством графа G. Матроид M^* называется *матроидом разрезов* или *матроидом связей* графа G.

Определенные таким образом матроиды M и M^* обладают интересным свойством, заключающимся в том, что базы одного из них являются дополнениями в E баз другого. Это утверждение справедливо для любого матроида на произвольном конечном S (не обяза-

тельно множестве ребер графа). Говоря другими словами, для любого матроида M на множестве S существует такой матроид M^* на S, что базы матроида M^* являются дополнениями баз матроида M. Этот результат мы обсуждаем в разд. 10.4.

Другим примером матроида является *матроид паросочетаний*, определенный на множестве вершин графа.

Теорема 10.1. Пусть G — неориентированный граф с множеством вершин V. Пусть $\mathcal{F}$ — набор всех таких подмножеств $I \subseteq V$, что элементы I насыщаются в некотором паросочетании графа G. Тогда $\mathcal{F}$ — набор независимых множеств матроида на V.

Доказательство. Очевидно, что $\mathcal{F}$ удовлетворяет аксиомам I.1 и I.2.

Чтобы показать, что $\mathcal{F}$ удовлетворяет аксиоме I.3, рассмотрим два произвольных члена I_p и I_{p+1} набора $\mathcal{F}$, содержащие p и $p+1$ вершин соответственно. Пусть X_p и X_{p+1} — произвольные паросочетания, насыщающие элементы I_p и I_{p+1} соответственно. Возможны два случая.

Случай 1. Предположим, что некоторый элемент $x \in I_{p+1} - I_p$ насыщается в X_p. Тогда X_p насыщает $I_p \cup x$ и аксиома I.3 выполняется.

Случай 2. Допустим, что никакой элемент $x \in I_{p+1}$ не насыщается в X_p. Тогда рассмотрим подграф G' на множестве ребер $X_p \oplus X_{p+1} = (X_p - X_{p+1}) \cup (X_{p+1} - X_p)$. По теореме 8.19 каждая компонента подграфа G' является 1) циклом, ребра которого входят поочередно в X_p и X_{p+1}, либо 2) путем, ребра которого входят поочередно в X_p и X_{p+1}, а его концевые вершины в одном из паросочетаний не насыщены.

Так как $|I_{p+1} - I_p| > |I_p - I_{p+1}|$, в подграфе G' существует путь P от вершины $v \in I_{p+1} - I_p$ к вершине, лежащей вне $I_p - I_{p+1}$. Тогда $X_p \oplus P$ будет паросочетанием, насыщающим v и все элементы I_p. Таким образом, $I_p \cup v$ входит в $\mathcal{F}$ и выполняется аксиома I.3.

В качестве примера рассмотрим граф G, представленный на рис. 10.1, *а*. Множества $I_4 = \{v_1, v_2, v_3, v_4\}$ и $I_6 = \{v_1, v_2, v_4, v_5, v_6\}$ насыщаются в паросочетаниях $X_4 = \{e_1, e_2\}$ и $X_6 = \{e_2, e_3, e_4\}$ соответственно. Подграф на множестве ребер $X_4 \oplus X_6 = \{e_1, e_3, e_4\}$ показан на рис. 10.1, *б*. Существует путь P от вершины $v_5 \in I_6 - I_4$ к вершине $v_6 \notin I_4 - I_6$. Паросочетание $X_4 \oplus P = \{e_2, e_3, e_4\}$ насыщает множество $I_4 \cup v_5$.

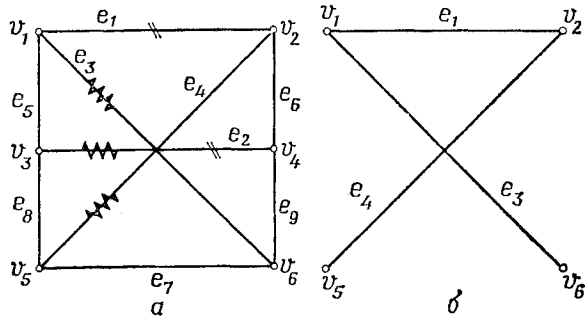


Рис. 10.1.

а — граф G с помеченными паросочетаниями X_4 и X_6; *б* — подграф G на множестве ребер $X_4 \oplus X_6$.

Два матроида M_1 и M_2 на множествах S_1 и S_2 соответственно называются изоморфными, если существует взаимно однозначное соответствие между элементами множеств S_1 и S_2, сохраняющее независимость.

Матроид M на множестве S называется *графическим*, если он изоморфен циклическому матроиду графа G. Если матроид M изоморфен матроиду разрезов графа G, он называется *кографическим*.

10.2. Фундаментальные свойства

В этом разделе мы установим несколько фундаментальных свойств матроидов. Все они известны в теории графов.

Сначала рассмотрим независимые множества и базы матроида M.

Теорема 10.2 (теорема добавления). Пусть X и Y — произвольные независимые подмножества в матроиде M. Если $|X|<|Y|$, то существует такое $Z\subseteq Y-X$, что $|X\cup Z|=|Y|$ и $X\cup Z$ независимо в M.

Доказательство. Пусть Z_0 — такое множество, что 1) $Z_0\subseteq Y-X$; 2) $X\cup Z_0$ независимо в M; 3) если для любого $Z\subseteq Y-X$ $X\cup Z$ независимо в M, то $|X\cup Z_0|\geqslant |X\cup Z|$. Допустим, что $|X\cup Z_0|<|Y|$. Тогда существует множество $Y_0\subseteq Y$ с $|Y_0|=|X\cup Z_0|$. Поскольку Y_0 независимо в M, по аксиоме I.3 найдется такой элемент $y\in Y_0-(Y-Z)$, что $X\cup Z\cup y$ независимо в M. Множество $Z_0\cup y$ противоречит выбору Z_0. Следовательно, $|X\cup Z_0|\geqslant|Y|$, что доказывает теорему.

Следствие 10.2.1. Все базы матроида M на множестве S имеют одинаковую мощность, равную рангу M.

Доказательство. Допустим противное. Пусть B_1, B_2 — базы и $|B_1|<|B_2|$. Тогда по теореме добавления существует такое $Z\subseteq B_2-B_1$, что $B_1\cup Z$ — независимое множество. Однако это противоречит максимальности B_1 в $\mathcal{F}$.

С помощью теоремы добавления можно доказать также обобщение этого результата.

Следствие 10.2.2. Пусть M — матроид на множестве S, $A\subseteq S$. Тогда все максимальные независимые подмножества A имеют одинаковую мощность.

Другое следствие из теоремы 10.2 сформулируем без доказательства.

Следствие 10.2.3. Если B_1 и B_2 — базы матроида M и $x\in B_1-B_2$, то существует такое $y\in B_2-B_1$, что $(B_1\cup y)-x$ является базой матроида M.

Теперь изучим свойства функции ранга матроида.

Теорема 10.3. Функция ранга ρ матроида M на множестве обладает следующими свойствами:

1. $0\leqslant\rho(A)\leqslant A$.
2. Если $A\subseteq B$, то $\rho(A)\leqslant\rho(B)$.
3. $\rho(A)\leqslant\rho(A\cup x)\leqslant\rho(A)+1$.
4. Подмодульное неравенство $\rho(A)+\rho(B)\geqslant\rho(A\cup B)+\rho(A\cap B)$.
5. Если $\rho(A\cup y)=\rho(A\cup x)=\rho(A)$, то $\rho(A\cup y\cup x)=\rho(A)$.

Доказательство. Первые три свойства следуют непосредственно из определения функции ранга.

Докажем свойство 4. Пусть X — максимальное независимое подмножество $A\cap B$. По теореме добавления существует максимальное независимое подмножество $Y\subseteq A\cup B$, содержащее X. Пусть $Y=X\cup V\cup W$, где $V\subseteq A-B$, а $W\subseteq B-A$. Так как $X\cup V$ — независимое подмножество A, а $X\cup W$ — независимое подмножество B, то получим $\rho(A)\geqslant|X\cup V|$ и $\rho(B)\geqslant|X\cup W|$. Следовательно, $\rho(A)+\rho(B)\geqslant|X\cup V|+|X\cup W|=2|X|+|V|+|W|=|X|+(|X|+|V|+|W|)=\rho(A\cap B)+\rho(A\cup B)$.

Свойство 5 является следствием свойства 4.

Перейдем к рассмотрению циклов матроида. Сформулируемые ниже свойства непосредственно следуют из определения цикла.

1. Любое собственное подмножество цикла — независимо. Поэтому если C_1 и C_2 — различные циклы, то $C_1 \not\subseteq C_2$.
2. Если C — цикл, то $\rho(C)=|C|-1$.
3. Матроид M на множестве S не содержит циклов тогда и только тогда, когда все подмножества S — независимы. Таким образом, множество S является единственной базой такого матроида M.

Теорема 10.4. Если C_1 и C_2 — различные циклы матроида M и $x \in C_1 \cap C_2$, то существует такой цикл C_3, что $C_3 \subseteq (C_1 \cup C_2)-x$.

Доказательство. Пусть $C'=(C_1 \cup C_2)-x$. Покажем, что C' — зависимо, т. е. $\rho(C')<|C'|$, доказывая тем самым теорему.

Поскольку C_1 и C_2 — различные циклы, $C_1 \cap C_2$ является собственным подмножеством как C_1, так и C_2. Поэтому $C_1 \cap C_2$ независимо. Таким образом, $\rho(C_1 \cap C_2)=|C_1 \cap C_2|$. Кроме того, $\rho(C_1)=|C_1|-1$ и $\rho(C_2)=|C_2|-1$. Теперь, используя этот результат и подмодульность функции ранга (теорема 10.3), получим

$$\begin{aligned}\rho(C_1 \cup C_2) &\leqslant \rho(C_1)+\rho(C_2)-\rho(C_1 \cap C_2)= \\ &=|C_1|-1+|C_2|-1-|C_1 \cap C_2|=|C_1 \cup C_2|-2<|C'|. \end{aligned} \tag{10.1}$$

Поскольку $C' \subseteq C_1 \cup C_2$, имеем также

$$\rho(C') \leqslant \rho(C_1 \cup C_2). \tag{10.2}$$

Объединив выражения (10.1) и (10.2), получим $\rho(C')<|C'|$.

Следствие 10.4.1. Если A — независимое множество матроида M на множестве S, то $A \cup x$ для любого $x \in S$ содержит не более одного цикла.

Доказательство. Допустим, что для некоторого $x \in S$ существуют два таких различных цикла C_1 и C_2, что $x \subseteq C_1 \cup C_2$, $C_1 \subseteq A \cup x$ и $C_2 \subseteq Ax$. Тогда по предыдущей теореме существует цикл $C' \subseteq (C_1 \cup C_2)-x$. Следовательно, $(C_1 \cup C_2)-x$ — зависимое множество. Но $(C_1 \cup C_2)-x \subseteq A$, что противоречит независимости множества A.

Следствие 10.4.2. Если B — база матроида M на множестве S и $x \in S-B$, то существует такой единственный цикл $C=C(x, B)$, что $x \in C \subseteq B \cup x$.

Доказательство. Поскольку B — максимальное независимое множество, в $B \cup x$ содержится цикл. По предыдущему следствию этот цикл единственный.

Определенный в этом следствии цикл $C(x, B)$ называется *базисным циклом* элемента x по отношению к базе B.

Докажем теперь более сильный результат, чем теорема 10.4.

Теорема 10.5. Если C_1 и C_2 — различные циклы матроида и $x \in C_1 \cap C_2$, то для любого $y \in C_1-C_2$ существует такой цикл C, что $y \in C \subseteq (C_1 \cup C_2)-x$.

Доказательство. Допустим, что C_1, C_2, x, y таковы, что теорема неверна и что $|C_1 \cup C_2|$ минимально среди всех пар различных циклов, не удовлетворяющих теореме.

По теореме 10.4 существует цикл $C_3=(C_1 \cup C_2)-x$. Из выбора x и y очевидно, что $y \notin C_3$. Кроме того, $C_3 \cap (C_2-C_1) \neq \varnothing$, в противном случае $C_3 \leqslant C_1$, что противоречит тому факту, что C_1 — минимальное зависимое множество. Пусть тогда $z \in C_3 \cap (C_2-C_1)$.

Теперь для циклов C_2 и C_3 справедливы следующие утверждения: 1) $z \in C_2 \cap C_3$; 2) $x \in C_2-C_3$; 3) $C_2 \cup C_3$ — собственное подмножество $C_1 \cup C_2$, поскольку $y \notin C_2 \cup C_3$. Из выбора C_1 и C_2 следует, что существует такой цикл C_4, что $x \in C_4 \subseteq (C_2 \cup C_3)-z$.

Для циклов C_1 и C_4 имеем 1) $x \in C_1 \cap C_4$; 2) $y \in C_1-C_4$, так как $y \notin C_2 \cup C_3$; 3) $C_1 \cup C_4$ — собственное подмножество $C_1 \cup C_2$, поскольку $z \notin C_1 \cup C_4$.

Из выбора C_1 и C_2 также следует, что существует такой цикл C_5, что $y \in C_5 \subseteq (C_1 \cup C_4)-x$.

Поскольку $C_1 \cup C_4$ — собственное подмножество $C_1 \cup C_2$, цикл C_5 таков, что $y \in C_5 \subseteq (C_1 \cup C_2)-x$, а это противоречит выбору C_1 и C_2.

10.3. Эквивалентные системы аксиом

В этом разделе мы сформулируем несколько альтернативных систем аксиом, определяющих матроид. Эти эквивалентные характеризации помогут нам в более глубоком проникновении в структуру матроидов. Начнем с эквивалентного аксиомам независимости набора.

Теорема 10.6. (Аксиомы независимости.) Набор $\mathcal{F}$ подмножеств множества S составляет множество независимых множеств матроида тогда и только тогда, когда $\mathcal{F}$ удовлетворяет аксиомам I.1, I.2 и I.3′. Если A — произвольное подмножество множества S, то все максимальные подмножества $Y \subseteq A$, входящие в $\mathcal{F}$, имеют одинаковую мощность.

Доказательство предлагается читателю в качестве упражнения. См. следствие 10.2.3.

Теорема 10.7. (Аксиомы баз.) Пусть $\mathcal{B}$ — множество баз матроида. Тогда

B.1. $\mathcal{B} \neq \varnothing$ и никакое множество в $\mathcal{B}$ не содержит в качестве собственного подмножества другое.

B.2. Если $B_1, B_2 \in \mathcal{B}$ и $x \in B_1$, тогда существует $y \in B_2$, такой, что $(B_1 - x) \cup y \in \mathcal{B}$.

Обратно, если $\mathcal{B}$ — набор подмножеств S, удовлетворяющих аксиомам B.1 и B.2, то $\mathcal{F} = \{I \mid I \subseteq B$ для некоторого $B \in \mathcal{B}\}$ является набором независимых множеств матроида на S.

Доказательство. Мы уже доказали, что базы матроида удовлетворяют аксиомам B.1 и B.2. (Вспомните определение базы и следствие 10.2.3.)

Пусть дан набор $\mathcal{B}$, удовлетворяющий аксиомам B.1 и B.2. Покажем, что набор $\mathcal{F}$, определенный в формулировке теоремы, является набором независимых множеств матроида на S.

Прежде всего заметим, что из аксиом B.1 и B.2 следует существование для $x \in B_1 - B_2$ такого $y \in B_2 - B_1$, что $(B_1 - x) \cup y$ входит в $\mathcal{B}$.

Очевидно, что $\mathcal{F}$ удовлетворяет аксиомам I.1 и I.2. Для доказательства удовлетворения аксиоме I.3 мы сначала покажем, что все члены $\mathcal{B}$ имеют одинаковую мощность.

Допустим, что существуют $B_1, B_2 \in \mathcal{B}$ с $|B_1| > |B_2|$. Тогда, повторяя применение аксиомы B.2, мы сможем удалить $B_2 - B_1$ из B_2 и заменить это подмножество на такое множество $A \subset B_1 - B_2$ с $|A| = |B_2 - B_1|$, что $B_3 = (B_2 \cap B_1) \cup A$ будет членом $\mathcal{B}$. Но $B_3 \subset B_1$, что противоречит аксиоме B.1. Таким образом, все члены набора $\mathcal{B}$ имеют одинаковую мощность.

Рассмотрим теперь два любых различных члена X и Y набора $\mathcal{F}$ с $|Y| = |X| + 1$. Пусть $X \subseteq B_1$, $Y \subseteq B_2$, где $B_1, B_2 \in \mathcal{B}$. Пусть $X = \{x_1, x_2, \ldots, x_k\}$; $B_1 = \{x_1, x_2, \ldots, x_k, b_1, b_2, \ldots, b_q\}$; $Y = \{y_1, y_2, \ldots, y_k, y_{k+1}\}$; $B_2 = \{y_1, y_2, \ldots, y_{k+1}, c_1, c_2, \ldots, c_{q-1}\}$.

Рассмотрим множество $B_1 - b_q$. По аксиоме B.2 существует такой $z \in B_2$, что $B' = (B_1 - b_q) \cup z$ входит в $\mathcal{B}$. Если $z \in Y$, то $(X \cup z) \subseteq B'$ и, следовательно, является членом $\mathcal{F}$. Поэтому аксиома I.3 в этом случае удовлетворяется.

Если $z \notin Y$, то рассмотрим $B'' = B' - b_{q-1}$. Снова по аксиоме B.2 существует такой $z' \in B_2$, что $B'' = (B' - b_{q-1}) \cup z'$ является членом $\mathcal{B}$. Если $z' \in Y$, то $Y \cup z' \subseteq B''$. Следовательно, $(X \cup z') \in \mathcal{F}$ и аксиома I.3 удовлетворяется.

Если $z' \notin Y$, удалим из B'' член b_{q-2} и т. д. Поскольку $|\{b_1, b_2, \ldots, b_q\}| > |\{c_1, c_2, \ldots, c_{q-1}\}|$, не более чем через q шагов мы достигнем ситуации, в которой заменим некоторый b_i на элемент Y, посредством чего выполним аксиому I.3.

Теорема 10.8. (Аксиомы ранга.) Функция ранга ρ матроида M на множестве удовлетворяет аксиомам

R.1. $0 \leqslant \rho \leqslant |A|$ для любого $A \subseteq S$.

R.2. Если $A \subseteq B \subseteq S$, то $\rho(A) \leqslant \rho(B)$.

R.3. Для любых A, $B \subseteq S$ $\rho(A) + \rho(B) \geqslant \rho(A \cup B) + \rho(A \cap B)$.

Обратно, если определенная на конечном множестве S целочисленная функция ρ удовлетворяет аксиомам R.1 — R.3, то множество $\mathcal{F}=\{I|\rho(I)=|I|,\ I\subseteq S\}$ является набором независимых множеств матроида на S.

Доказательство. Мы уже доказали в теореме 10.3, что функция ранга матроида удовлетворяет аксиомам R.1 — R.3.

Для доказательства обратного предположим, что функция ρ удовлетворяет аксиомам R.1 — R.3. Из аксиомы R.1 следует, что $\rho(\varnothing)=0$. Поэтому $\varnothing\in\mathcal{F}$ и аксиома I.1 удовлетворяется.

Пусть $B\in\mathcal{F}$ и $A\subseteq B$. Тогда по аксиоме R.3 получаем

$$\rho(A)+\rho(B-A)\geqslant\rho(A\cup(B-A))+\rho(A\cap(B-A))=\rho(B)+\rho(\varnothing)=|B|. \quad (10.3)$$

Если $\rho(A)<|A|$, то по аксиоме R.1 $\rho(A)+\rho(B-A)<|A|+|B-A|=|B|$, что противоречит формуле (10.3). Следовательно, $\rho(A)=|A|$ и A также является членом $\mathcal{F}$, вследствие чего удовлетворяется аксиома I.2.

Для доказательства аксиомы I.3 заметим сначала, что из аксиом R.2 и R.3 следует аксиома

R.3′. Если $\rho(A\cup x)=\rho(A\cup y)=\rho(A)$, то $\rho(A\cup x\cup y)=\rho(A)$. Пусть X и Y — различные члены $\mathcal{F}$ с $|X|=k$ и $|Y|=k+1$. Предположим, что для любого $y\in Y-X$ $\rho(X\cup y)=k$. Тогда, применяя повторно аксиому R.3′, получим $\rho(X\cup(Y-X))=\rho(X\cup Y)=k$ в противоречие с $\rho(Y)=k+1$. Поэтому существует элемент $y\in Y-X$, для которого $\rho(X\cup y)=k+1$, и аксиома I.3, таким образом, удовлетворяется.

Теорема 10.9. (Аксиомы циклов.) Пусть $\mathcal{C}$ — множество циклов матроида на множестве S. Тогда

C.1. $\varnothing\notin\mathcal{C}$ и никакое множество в $\mathcal{C}$ не содержит другого в качестве собственного подмножества.

C.2. Если C_1, $C_2\in\mathcal{C}$, $C_1\neq C_2$ и $x\in C_1\cap C_2$, то существует такой $C_3\in\mathcal{C}$, что $C_3\subseteq(C_1\cup C_2)-x$.

Обратно, если набор $\mathcal{C}$ подмножеств множества S удовлетворяет аксиомам C.1 и C.2, то множество $\mathcal{F}=\{I|C\not\subseteq I$ для всех $C\in\mathcal{C}\}$ является набором независимых множеств матроида на S.

Доказательство. Мы уже доказали, что циклы матроида удовлетворяют аксиомам C.1 и C.2 (теорема 10.4).

Пусть дан набор $\mathcal{C}$, удовлетворяющий аксиомам C.1 и C.2. Покажем, что набор $\mathcal{F}$, определенный, как в формулировке теоремы, является набором независимых множеств матроида на S. Сделаем это, показывая последовательно, что $\mathcal{F}$ удовлетворяет аксиомам I.1, I.2 и I.3′ (теорема 10.6).

Заметим сначала, что $\mathcal{F}$ есть набор всех подмножеств S, не содержащих ни одного члена $\mathcal{C}$. Поэтому он удовлетворяет аксиомам I.1 и I.2.

Для доказательства того, что $\mathcal{F}$ удовлетворяет аксиоме I.3′, рассмотрим произвольное подмножество $A\subseteq S$. Пусть S_1 и S_2 — различные максимальные подмножества A, принадлежащие $\mathcal{F}$. Допуская $|S_2|>|S_1|$, установим противоречие.

Так как S_1 и S_2 — различные подмножества, $S_1-S_2\neq\varnothing$ и $S_2-S_1=\varnothing$. Пусть $x\in S_2-S_1$. Тогда, очевидно, $S_1\cup x\notin\mathcal{F}$. Следовательно, существует $C\in\mathcal{C}$, для которого $x\in C\subseteq S_1\cup x$. Более того, $C\cap(S_1-S_2)\neq\varnothing$, иначе бы $C\subset S_2$, что противоречит принадлежности S_2 к $\mathcal{F}$.

Пусть $x'\in C\cap(S_1-S_2)$ и $S_3=(S_1-x')\cup x$. Заметим, что $|S_3|=|S_1|$. Сначала докажем:

1. $S_3\in\mathcal{F}$. Очевидно, что $S_1-x'\in\mathcal{F}$. Если бы $S_3\notin\mathcal{F}$, тогда существовал бы такой $C'\in\mathcal{C}$, что $x\in C'\subseteq S_3\subset S_1\cup x$. Более того, $C\neq C'$, так как $x'\in C'$. Таким образом, C и C' — такие различные члены $\mathcal{C}$, что а) $x\in C\cap C'$ и б) $C\subset S_1\cup x$ и $C'=S_1\cup x$. Тогда из аксиомы C.2 следует, что существует такой $C''\in\mathcal{C}$, что $C''\subseteq(C\cup C')-x\subseteq S_1\subseteq\mathcal{F}$. Это противоречит тому, что никакой член $\mathcal{F}$ не содержит членов $\mathcal{C}$. Следовательно, $S_3\in\mathcal{F}$.

Теперь докажем:

2. $S_3\in\mathcal{F}$ — *максимальное подмножество множества* A.

Допустим противное. Пусть $S' \in \mathcal{F}$ — такое максимальное подмножество множества A, что $S_3 \subset S'$. Теперь $x' \in S'$, так как в противном случае $S_1 \subset S'$, что противоречит максимальности S_1 в A. Тогда $S' \cup x' \notin \mathcal{F}$. Следовательно, существует такой $C'' \in \mathcal{C}$, что $x' \in C'' \subseteq S' \cup x'$. Более того, $C'' \cap (S' - S_1) \neq \varnothing$, так как иначе бы $C'' \subset S_1$. Пусть $x'' \in C \cap (S' - S_1)$. Тогда можно показать (как в доказательстве п. 1), что $(S' - x'') \cup x' \in \mathcal{F}$. Но $S_1 \subset (S' - x'') \cup x' \in \mathcal{F}$, что противоречит максимальности S_1 в A. Таким образом, S_3— максимальное подмножество в A.

Заметим, что S_3 построено из S_1 заменой элемента $x' \in S_1 - S_2$ на элемент $x \in S_2 - S_1$. Поскольку $|S_2| > |S_1|$, мы можем повторить эту конструкцию конечное число раз, получив максимальное такое подмножество $S_n \subseteq A$, что $S_n \in \mathcal{F}$ и $S_n \subset S_2$, что противоречит равенству $\rho(Y) = k+1$. Поэтому существует такой элемент $y \in Y - X$, что $\rho(X \cup y) = k+1$. Поэтому существует такой элемент $y \in Y - X$, что $\rho = (X \cup y) = k+1$ и аксиома I.3, таким образом, удовлетворяется.

10.4. Двойственность матроидов и графоиды

В этом разделе мы сначала введем понятие двойственности матроидов и обсудим некоторые результаты, связывающие матроид с двойственным к нему матроидом. Затем мы сформулируем теорему о «раскрашивании», которая приведет нас к понятию графоида. Мы завершим раздел развитием самодвойственной системы аксиом Минти для матроидов, основанной на графоидах.

Начнем с теоремы, предложенной Уитни [10.1], в которой определяется двойственный матроид.

Теорема 10.10. Пусть $\mathcal{B}$ — множество баз матроида M на множестве S. Тогда $\mathcal{B}^* = \{S - B \mid B \in \mathcal{B}\}$ есть множество баз матроида M^* на S.

Доказательство. Очевидно, что $\mathcal{B}^*$ удовлетворяет аксиоме B.1. Чтобы показать, что аксиома B.2 также выполняется, рассмотрим любые такие два произвольных члена B_1^* и B_2^* набора $\mathcal{B}^*$, что $B_1^* = S - B_1$, $B_2^* = S - B_2$. Пусть $x \in B_1^* - B_2^*$. Тогда $x \in B_2 - B_1$. Из упражнения 10.4 следует, что существует такое $y \in B_1 - B_2$, что $B_3 = (B_1 - y) \cup x$ является базой M. Теперь $y \in B_2^* - B_1^*$ и $(B_1^* - y) \cup x = S - [(B_1 - y) \cup x] = S - B_3 = B_3^*$. Таким образом, аксиома B.2 выполняется и, следовательно, $\mathcal{B}^*$— множество баз матроида M на S.

Определенный в теореме матроид M^* называется *двойственным матроидом* к M. Легко видеть, что M — двойственный к M^* матроид. Поэтому будем говорить, что M и M^*— двойственные матроиды.

Определенные в разд. 10.1 матроид разрезов и циклический матроид графа, как нетрудно заметить, являются двойственными матроидами.

База M^* называется *кобазой* M. Аналогично цикл M^*— *коцикл* M, петля M^* — *копетля* M, функция ранга M^* — *функция коранга* M и т. д. Функция ранга M^* обозначается через ρ^*. Если B — база M, то кобаза $S - B$ обозначается через B^*. Набор кобаз M обозначается через $\mathcal{B}^*(M)$, а набор коциклов — через $\mathcal{C}^*(M)$.

Теперь сформулируем результаты, связывающие матроид с двойственным к нему матроидом. Многие из них хорошо известны в теории графов (гл. 2).

Очевидно, что из определения двойственного матроида следует

$$\rho^*(M) = |S| - \rho(M). \tag{10.4}$$

В следующей теореме раскрывается взаимосвязь ρ и ρ^*.

Теорема 10.11. Если M и M^* — двойственные матроиды на множестве S, тогда для любого $A \subseteq S$ справедлива формула

$$\rho^*(A) = |A| + \rho(S-A) - \rho(M). \tag{10.5}$$

Доказательство. Рассмотрим $A \subseteq S$; пусть B^*— такая база M^*, что $|B^* \cap A|$ — максимально. Тогда B — такая база M, что $|B \cap (S-A)|$ — максимально. Из определения функции ранга следует, что

$$\rho^*(A) = |B^* \cap A|, \tag{10.6}$$

$$\rho(S-A) = |B \cap (S-A)|. \tag{10.7}$$

Тогда $|B^* \cap A| = |A| - |B \cap A|$ и $|B \cap (S-A)| = |B| - |B \cap A| = \rho(M) - |B \cap M|$. Таким образом, из выражений (10.6) и (10.7) следует $\rho^*(A) = |A| + \rho(S-A) - \rho(M)$.

Заметим, что для любого утверждения о циклах, базах и т. д. матроида существует двойственное утверждение о коциклах, кобазах и т. д. матроида. Это обусловлено тем, что коциклы, кобазы и т. д. сами являются циклами, базами и т. д. матроида.

Все леммы и теоремы этого раздела мы дополняем, где это возможно, двойственными утверждениями. Доказательства этих двойственных утверждений очевидны.

Лемма 10.1. Пусть M — матроид на множестве S. Если $A \subseteq S$ — независимое множество M, то $S-A$ содержит кобазу M. Двойственное утверждение: если $A^* \subseteq S$ — независимое множество M^*, то $S-A^*$ содержит базу M.

Доказательство. Существует такая база B матроида M, что $A \subseteq B$. Поэтому $S-A$ содержит соответствующую кобазу B^*.

Теорема 10.12. Пусть M — матроид на множестве S. Если A и A^* — такие подмножества S, что $A \cap A^* = \varnothing$, A — независимое множество M, а A^*— независимое множество M^*, то существует такая база B матроида M, что $A \subseteq B$, а $A^* \subseteq B^*$.

Доказательство. По предыдущей лемме $S-A^*$ содержит базу матроида M. Поскольку $A \subseteq S-A^*$— независимое множество M, то по теореме 10.2 существует такая база B матроида M, что $A \subseteq B \subseteq S-A^*$, а, следовательно, $A^* \subseteq B^*$.

Лемма 10.2. Пусть M — матроид на множестве S. Тогда любая база B матроида M имеет непустое пересечение с любым коциклом M.

Аналогично любая кобаза M имеет непустое пересечение с любым коциклом M.

Доказательство. Если бы для некоторого коцикла C^* матроида M $C^* \cap B = \varnothing$, то $B^* = S-B$ содержало бы S^*, что противоречит независимости B^* в M^*.

Пусть B^*— произвольная кобаза матроида на множестве S. По двойственному утверждению к следствию 10.4.2 для любого $x \in B$ $B^* \cup x$ содержит в точности один коцикл. Он называется *базисным коциклом* элемента x по отношению к B. Заметим, что этот коцикл содержит точно один элемент из B, а именно x.

Лемма 10.3. Пусть M — матроид. Для любого независимого множества A матроида M существует коцикл, имеющий точно один элемент из A. Кроме того, если $|A| < \rho(M)$, то существует коцикл, имеющий пустое пересечение с A.

Аналогично для любого независимого множества A^* матроида M^* существует цикл, имеющий точно один элемент из A^*. Если $|A^*| < \rho^*(M)$, то существует цикл, имеющий с A^* пустое пересечение.

Доказательство. Пусть B — такая база M, что $A \subseteq B$, а C^*— базисный коцикл элемента $x \in B$ по отношению к B. Тогда $C^* \cap A = \{x\}$, если $x \in A$, и $C^* \cap A = \varnothing$, если $x \in B-A$. Таким образом, лемма доказана.

Теорема 10.13. Пусть M — матроид на множестве S. Подмножество $X \subseteq S$ является базой M тогда и только тогда, когда X — минимальное подмножество, имеющее непустое пересечение со всеми коциклами M.

Аналогично подмножество $X^* \subseteq S$ — кобаза M тогда и только тогда, когда оно является минимальным подмножеством, имеющим непустое пересечение со всеми циклами M.

Доказательство. Необходимость следует из лемм 10.2 и 10.3.

Для доказательства достаточности положим, что X — минимальное подмножество S, имеющее непустое пересечение со всяким коциклом M. Тогда $S-X$ — максимальное подмножество, не содержащее коциклов M. Поэтому $S-X$ является по определению кобазой M, а следовательно, X — базой M.

Получим сейчас несколько новых характеризаций циклов и коциклов.

Теорема 10.14. Пусть M — матроид на множестве S. Подмножество $X\subseteq S$ есть цикл M тогда и только тогда, когда оно является минимальным подмножеством, имеющим непустое пересечение со всякой базой M.

Аналогично подмножество $X^*\subseteq S$ есть коцикл M тогда и только тогда, когда оно является минимальным подмножеством, имеющим непустое пересечение со всякой базой M.

Доказательство аналогично доказательству теоремы 2.13.

Пусть C^* — коцикл матроида M на S. По предыдущей теореме C^*— минимальное подмножество, имеющее непустое пересечение со всякой базой M. Поэтому $S-C^*$ — максимальное подмножество, не содержащее баз M. Таким образом, получаем следующее утверждение.

Лемма 10.4. Если C^* — коцикл матроида M на множестве S, то $(S-C^*)\cup x$ содержит базу B матроида M для любого $x\in C$.

Аналогично если C — цикл матроида M на множестве S, то $(S-C)\cup x$ содержит кобазу M для любого $x\in C$.

Теорема 10.15. Пусть M — матроид на множестве S. Подмножество $X\subseteq S$ есть цикл M тогда и только тогда, когда оно является минимальным подмножеством с $|X\cap C^*|\neq 1$ для всякого коцикла C^* матроида M.

Аналогично подмножество $X^*\subseteq S$ есть коцикл M тогда и только тогда, когда оно является минимальным подмножеством с $|X^*\cap C|=1$ для всякого цикла C матроида M.

Доказательство. Необходимость. Допустим противное. Тогда для любого собственного подмножества C' цикла C по лемме 10.3 существует такой коцикл C^*, что $|C^*\cap C'|=1$.

Допустим, что $|C\cap C^*|=1$ для некоторого цикла C и некоторого коцикла C^*, пусть $C\cap C^*=\{x\}$. Рассмотрим теперь $S'=S-C^*$ и $C'=C-x$. Очевидно, что $C'\subseteq S'$. По лемме 10.4 $S'\cup x$ содержит базу. Пусть $B\subseteq S'\cup x$ — такая база, что $C'\subseteq B$. Заметим, что $x\in B$. Поэтому цикл $C=C'\cup x$ содержится в B; получили противоречие.

Достаточность. Если X — такое подмножество S, что $|X\cap C^*|\neq 1$ для всякого коцикла C^*, то X должно содержать цикл, иначе бы X было независимым множеством M и по лемме 10.3 существовал коцикл, имеющий в точности один элемент их X.

Пусть C — содержащийся в X цикл. Из доказанного необходимого условия теоремы очевидно, что $|C\cap C^*|\neq 1$ для всякого коцикла C^*. Поэтому $X=C$, иначе минимальность X приводит к противоречию.

Теперь введем понятие раскрашивания конечного множества и сформулируем теорему о «раскрашивании».

Раскрашивание множества S — это разбиение S на три таких подмножества R, G и B, что $|G|=1$. Для облегчения представления мы считаем, что элементы R «окрашены» в красный цвет, элемент G — в зеленый, а элементы B — в голубой.

Теорема 10.16. (теорема о раскрашивании). Пусть M — матроид на множестве S. Для любого раскрашивания S существует

1) цикл C матроида M, содержащий зеленый элемент и не имеющий голубых элементов, либо
2) коцикл C^* матроида M, содержащий зеленый элемент и не имеющий красных элементов.

Доказательство. Допустим, что зеленый элемент входит в некоторый цикл, иначе он был бы копетлей (почему?), и утверждение теоремы становится тривиальным.

Предположим, что для некоторого раскрашивания S утверждение 1 не истинно. Рассмотрим тогда подмножество $S'=R\cup G$. Очевидно, что любой содержащийся в S' цикл состоит только из красных элементов. Пусть R'— минимальное подмножество красных элементов, для которого $S''=S'-R'$ не содержит циклов, а B — база, для которой $S''\subseteq B$. Тогда легко видеть, что базисный цикл любого элемента $y\in R'$ по отношению к B состоит только из красных элементов.

Пусть теперь C^*— базисный коцикл зеленого элемента по отношению к B. Допустим, что C^* содержит красный элемент x. Тогда $x\in R'$ и $|C^*\cap C_x|=1$, где C_x— базисный цикл x по отношению к B. Это противоречит теореме 10.15, следовательно, C^* — коцикл, содержащий зеленый элемент и не имеющий красных элементов. Таким образом, утверждение 2 истинно.

Теоремы 10.15 и 10.16 приводят нас к определению графоида, введенному Минти [10.2].

Графоид — это структура $(S, \mathscr{C}, \mathscr{D})$, состоящая из конечного множества S и двух наборов $\mathscr{C}$ и $\mathscr{D}$ непустых подмножеств S, удовлетворяющих следующим условиям:

G.1. Если $C\in\mathscr{C}$ и $D\in\mathscr{D}$, то $|C\cap D|=1$.

G.2. Для любого раскрашивания S существует а) член $\mathscr{C}$, содержащий зеленый элемент и не имеющий голубых элементов, либо б) член $\mathscr{D}$, содержащий зеленый элемент и не имеющий красных элементов.

G.3. Никакой член $\mathscr{C}$ не содержит другого члена $\mathscr{C}$ в качестве собственного подмножества; никакой член $\mathscr{D}$ не содержит другого члена $\mathscr{D}$ в качестве собственного подмножества.

Теорема 10.17. Пусть M — матроид на множестве S. Тогда $[S, \mathscr{C}(M), \mathscr{C}^*(M)]$ — графоид.

Доказательство следует из теорем 10.15 и 10.16.

Установим сейчас обратную теорему.

Теорема 10.18. Пусть $(S, \mathscr{C}, \mathscr{D})$ — графоид. Тогда $B\subseteq S$ — максимальное подмножество, не содержащее члена $\mathscr{C}$, тогда и только тогда, когда B не содержит членов $\mathscr{C}$, а $S-B$ — членов $\mathscr{D}$.

Доказательство.

Необходимость. Пусть B — максимальное подмножество S, не содержащее членов $\mathscr{C}$. Допустим, что $S-B$ содержит $D\in\mathscr{D}$. Пусть $x\in D$. Тогда $B\cup x$ содержит $C\in\mathscr{C}$ и $x\in C$. Поэтому $C\cap D=\{x\}$, что противоречит аксиоме G.1, Таким образом, $S-B$ не содержит членов $\mathscr{D}$.

Достаточность. Пусть $B\subseteq S$ не содержит членов $\mathscr{C}$, а $S-B$ — членов $\mathscr{D}$. Покажем, что для любого $x\in S-B$ $B\cup x$ содержит $C\in\mathscr{C}$.

Рассмотрим раскрашивание S, в котором все элементы B — красные, x — зеленый, а все оставшиеся элементы $S-B$ — голубые. Очевидно, что не существует члена $\mathscr{D}$, содержащего только зеленый и голубые элементы. Таким образом, по аксиоме G.2 $B\cup x$ содержит $C\in\mathscr{C}$.

Теорема 10.19. Пусть $(S, \mathscr{C}, \mathscr{D})$ — графоид. Если B — максимальное подмножество S, не содержащее элементов $\mathscr{C}$, то $S-B$ — максимальное подмножество S, не содержащее элементов $\mathscr{D}$.

Доказательство следует из теоремы 10.18 и двойственной к ней.

Теорема 10.20. Пусть $(S, \mathscr{C}, \mathscr{D})$ — графоид. Тогда $\mathscr{C}$ — набор циклов матроида M на S, а $\mathscr{D}$ — набор его коциклов.

Доказательство. Покажем сначала, что $\mathscr{C}$ — набор циклов матроида M_1 на S. Очевидно, что аксиома C.1 удовлетворяется.

Пусть C_1 и C_2— произвольные различные члены $\mathscr{C}$, для которых $x\in C_1\cap C_2$, $y\in C_1-C_2$. Рассмотрим раскрашивание S, в котором y — зеленый, x — голубой, оставшиеся элементы $C_1\cup C_2$— красные, а остальные элементы S — голубые (рис. 10.2).

Члена $D\in\mathscr{D}$, содержащего зеленый элемент и не имеющего красных элементов, не существует. Если бы такой член существовал, то, как легко показать, выполнялось бы условие $|C_1\cap D|=1$ или $|C_2\cap D|=1$, что противоречит аксиоме G.1. Поэтому по аксиоме G.2 существует $C_3\in\mathscr{C}$, содержащий зеленый элемент

и не имеющий голубых. По аксиоме G.3$\{y\}$ не является членом $\mathscr{C}$. Отсюда следует, что $y \in C_3 \subseteq (C_1 \cup C_2) - x$. Таким образом, аксиома C.2 удовлетворяется. Следовательно, $\mathscr{C}$ — набор циклов матроида M_1 на S.

Аналогично можно показать, что $\mathscr{D}$ есть набор циклов матроида M_2 на S.

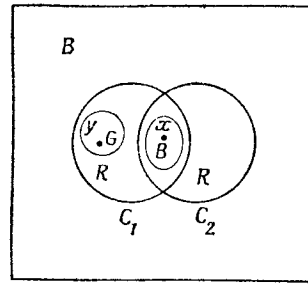


Рис. 10.2.

По теореме 10.19 базы M_1 являются дополнениями баз M_2. Поэтому M_1 и M_2 — двойственные матроиды.

Из теорем 10.17 и 10.20 следует эквивалентность системы аксиом для графоида и системы аксиом для циклов. Этот вывод сделан Минти [10.2], давшим элегантное изложение теории матроидов на основе графоидов.

10.5. Ограничение, сужение и миноры матроида

Рассмотрим граф G со множеством ребер E. По отношению к любому подмножеству $T \subseteq E$ мы можем определить два графа, обозначаемые $G|T$ и $G \cdot T$. Граф $G|T$, называемый *ограничением* G на T, получается из графа G уничтожением (размыканием) ребер, принад-

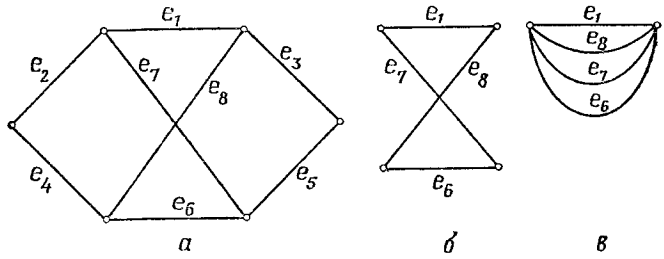


Рис. 10.3.

а — граф G; *б* — $G|T$, ограничение G на T, $T=\{e_1, e_6, e_7, e_8\}$; *в* — $G \cdot T$, сужение G на T.

лежащих $E-T$, и удалением образовавшихся изолированных вершин. Фактически граф $G|T$ является порожденным графом G на T. Граф $G \cdot T$, называемый *сужением* G на T, получается стягиванием ребер, принадлежащих $E-T$. На рис. 10.3 представлены граф G, графы $G|T$ и $G \cdot T$ для $T=\{e_1, e_6, e_7, e_8\}$.

Легко убедиться в следующем:

1. Граф G', порожденный подмножеством ребер X графа G, является ациклическим подграфом $G|T$ тогда и только тогда, когда G' ациклический подграф графа G.
2. Граф G', порожденный подмножеством ребер X графа G, является ациклическим подграфом $G\cdot T$ тогда и только тогда, когда существует такое подмножество $Y\subseteq E-T$, что граф, порожденный подмножеством Y, является остовным лесом $G|(E-T)$, а граф, порожденный подмножеством $X\cup Y$,— ациклическим подграфом графа G.

Эти положения мотивируют введение двух понятий подматроидов, порожденных на матроиде подмножествами его элементов.

Если $\mathcal{F}(M)$ — множество независимых множеств матроида M на S и $T\subseteq S$, тогда пусть $\mathcal{F}(M|T)=\{X|X\subseteq T,\ X\in\mathcal{F}(M)\}$. Легко видеть, что $\mathcal{F}(M|T)$ — множество независимых множеств матроида на T. Этот матроид обозначается $M|T$ и называется *ограничением* M на T. Очевидны следующие утверждения:

1. $X\subseteq T$ — независимое множество в $M|T$ тогда и только тогда, когда X — независимое множество в M.
2. $X\subseteq T$ — цикл $M|T$ тогда и только тогда, когда X — цикл матроида M.
3. Если λ — функция ранга $M|T$, тогда для любого $X\subseteq T$

$$\lambda(X)=\rho(X). \tag{10.8}$$

4. Если $M(G)$ — циклический матроид графа G с множеством ребер E, то $M(G)|T=M(G|T)$ для любого $T\subseteq E$.

Определим теперь подматроид $M\cdot T$. Пусть $\mathcal{F}(M\cdot T)$ — набор всех таких подмножеств $X\subseteq T$, что существует максимальное независимое подмножество $Y\subseteq S-T$, для которого $X\cup Y\in\mathcal{F}(M)$.

Теорема 10.21. $\mathcal{F}(M\cdot T)$ — набор независимых множеств матроида на T.

Доказательство. Пусть X_1 и X_2— такие члены $\mathcal{F}(M\cdot T)$, что $|X_2|=|X_1|+1$. Тогда существуют максимальные независимые подмножества Y_1, $Y_2\subseteq S-T$, для которых $X'=X_1\cup Y_1$ и $X''=X_2\cup Y_2$— независимые множества M. Очевидно, что $|X''|=|X'|+1$. Следовательно, по аксиоме I.3 существует такой элемент $x\in X''-X'$, что $X'\cup x$ — независимое множество в M. Заметим, что $X''-X'=(X_2-X_1)\cup(Y_2-Y_1)$. Кроме того, $x\notin Y_2-Y_1$, поскольку Y_1— максимальное независимое подмножество $S-T$. Поэтому $x\in X_2-X_1$ и $(X_1\cup x)\subseteq T$ входит в $\mathcal{F}(M\cdot T)$. Таким образом, $\mathcal{F}(M\cdot T)$ удовлетворяет аксиоме I.3. Легко видеть, что аксиомы I.1 и I.2 также выполняются.

Определенный в теореме матроид называется *сужением* M на T и обозначается $M\cdot T$. Заметим, что если $M(G)$ — циклический матроид графа G с множеством ребер E, то $M(G\cdot T)=M(G)\cdot T$ для любого $T\subseteq E$.

Пусть ρ^T— функция ранга $M\cdot T$. Тогда из определения $M\cdot T$ получим, что для любого $A\subseteq T$

$$\rho^T(A)=\rho(A\cup(S-T))-\rho(S-T). \tag{10.9}$$

В гл. 7 мы заметили, что размыкание или удаление ребер и стягивание ребер графа — двойственные операции. В частности, если G и G' — двойственные графы, а T и T' — соответствующие подмножества ребер G и G' $G|T$ — двойственный к $G\cdot T$ граф (теорема 7.10). Следующая теорема — матроидный аналог этой взаимосвязи.

Теорема 10.22. Если M — матроид на S и $T \subseteq S$, тогда 1) $(M|T)^* = M^* \cdot T$; 2) $(M \cdot T)^* = M^*|T$.

Доказательство. 1. Пусть λ — функция ранга $M|T$, а λ^* — функция ранга $(M|T)^*$. Тогда из выражения (10.5) следует, что $\lambda^*(X) = |X| - \lambda(T) + \rho(T - X)$ для любого $X \subseteq T$.

Пусть ρ^T — функция ранга $M \cdot T$, а $(\rho^*)^T$ — функция ранга $M^* \cdot T$. Тогда из выражений (10.5) и (10.9) следует, что

$$(\rho^*)^T(X) = \rho^*[X \cup (S-T)] - \rho^*(S-T) = \rho^*[S-(T-X)] - \rho^*(S-T) = |S| - |T| + |X| - \rho(S) + \rho(T-X) - [|S| - |T| - \rho(S) + \rho(T)] = |X| - \rho(T) + \rho(T-X) = |X| - \lambda(T) + \lambda(T-X) = \lambda^*(X),$$

согласно выражению (10.8).

Так как $(M|T)^*$ и $M^* \cdot T$ имеют одну функцию ранга, то $(M|T)^* = M^* \cdot T$.

2. В п. 1 заменяя M^* на M и взяв двойственные матроиды, получим п. 2.

Пусть M — матроид на S и $T \subseteq S$, матроид N на T называется *минором* M, если N получен последовательностью ограничений и сужений M. Тема, связанная с минорами матроидов, глубоко изучена Таттом [10.3].

Используя терминологию теории матроидов, можно сформулировать множество результатов, касающихся планарных и двойственных графов. Например, теорему 7.5 можно сформулировать следующим образом:

Граф планарен тогда и только тогда, когда циклический матроид $M(G)$ графа не содержит в качестве миноров матроидов $M(K_5)$ и $M(K_{3,3})$.

10.6. Представимость матроидов

Пусть M — матроид на множестве S. Пусть $B = \{b_1, b_2, \ldots, b_r\}$ — база M, а $B^* = S - B = \{e_1, e_2, \ldots, e_q\}$ — соответствующая кобаза M. Напомним, что $B^* \cap C(e_j) = e_j$, $j = 1, \ldots, q$, где $C(e_j)$ — базисный цикл элемента e_j. Аналогично $B \cap C^*(b_j) = b_j$, $j = 1, \ldots, r$, где $C^*(b_j)$ — базисный коцикл элемента b_j. Как и в случае неориентированных графов (гл. 6), мы можем сейчас определить базисную цикломатическую матрицу D_f^* и базисную коцикломатическую матрицу D_f матроида M по отношению к базе B. Очевидно, что обе эти матрицы будут иметь вид $[U\ D]$, где U — единичная матрица, столбцы которой соответствуют элементам B. Матрица такого вида называется *стандартным представлением* матроида M по отношению к базе B.

Рассмотрим граф G. Присвоим произвольным образом ориентацию ребрам графа G, получив в результате ориентированный граф G'. Тогда по теореме 6.9 усеченная матрица инциденций G' есть представление циклического матроида $M(G)$ над произвольным полем F. Можно заметить также (теорема 6.10), что любая базисная матрица разрезающих множеств G' является стандартным представлением $M(G)$ над произвольным полем F. Далее, по теореме 6.11 любая базисная цикломатическая матрица G' является стандартным представлением циклического матроида $M(G)$ над произвольным полем F. Таким образом, приходим к следующему утверждению:

Теорема 10.23. Циклический матроид и матроид разрезов графа представимы над произвольным полем F.

Сейчас мы докажем основной результат этого раздела.

Теорема 10.24. Если матроид M на S представим над полем F, тогда двойственный матроид M^* на S также представим над F.

Доказательство. Допустим, M имеет ранг r и содержит n элементов. Пусть матрица A порядка $r\times n$ является представлением M над F.

Пусть X — множество всех таких векторов-столбцов x, что $Ax=0$. Величина X называется *нуль-пространством* матрицы A. Из линейной алгебры известно, что размерность X равна $n-r$. Выберем теперь из X множество из $n-r$ линейно независимых векторов-столбцов и образуем из этих векторов, как из столбцов, матрицу B порядка $n\times(n-r)$. Заметим, что $AB=0$.

Покажем сейчас, что B^t является представлением над F двойственного матроида M^*. Для этого докажем, что произвольные r столбцов матрицы A линейно независимы тогда и только тогда, когда дополняющее множество из $n-r$ столбцов B^t— линейно независимо. Выберем первые r столбцов матрицы A. Очевидно, что такой выбор не влечет потери общности.

Первые r столбцов матрицы A линейно зависимы тогда и только тогда, когда существует ненулевой вектор-столбец $x=[x_1, x_2, \ldots, x_r, 0, 0, \ldots, 0]^t$, принадлежащий X. Такой вектор $x\in X$, в свою очередь, существует тогда и только тогда, когда существует вектор-столбец $y\neq 0$ такой размерности $n-r$, что $x=By$. Записывая теперь B в виде $\begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$, где B_1 имеет порядок $r\times(n-r)$, а B_2— $-(n-r)\times(n-r)$, легко видеть, что $B_2y=0$. Поскольку $y\neq 0$, то B_2— вырожденная матрица. Таким образом, строки B_2, а следовательно, и последние $n-r$ столбцов B^t зависимы, что доказывает теорему.

Легким следствием этой теоремы является следующий результат:

Следствие 10.24.1. Пусть M — матроид ранга r на множестве $S=\{s_1, s_2, \ldots, s_n\}$. Если M имеет стандартное представление

$$\begin{array}{c} s_1 s_2 \cdots s_r \quad\quad s_{r+1} \cdots s_n \\ [\ U_r \quad | \quad A\ ], \end{array}$$

то M^* имеет стандартное представление

$$\begin{array}{c} s_1 s_2 \cdots s_r \quad\quad s_{r+1} \cdots s_n \\ [\ -A \quad | \quad U_{n-r}\ ], \end{array}$$

где U_k — единичная матрица порядка $k\times k$.

Из этого следствия легко получается выражение (6.13), связывающее базисную цикломатическую матрицу с базисной матрицей разрезающих множеств.

Дальнейшее обсуждение проблемы представимости матроидов можно найти в работе [10.4].

10.7. Бинарные матроиды

Матроид называется *бинарным*, если он представим над $GF(2)$, полем целых чисел по mod 2.

Очевидно, что циклический матроид $M(G)$ и матроид разрезов $M^*(G)$ графа G бинарные. Нами уже доказана теорема 4.6, в которой каждый цикл $M(G)$ выражается в виде суммы по mod 2 некоторых базисных циклов графа G. Подобный результат справедлив и для разрезающих множеств графа G. Это свойство циклов и разрезающих множеств (коциклов) сохраняется и для случая бинарных матроидов. Однако в общем случае произвольных матроидов оно не выполняется. Пусть, например, $S=1, 2, 3, 4$, а M в качестве циклов имеет все подмножества из трех элементов S. Тогда сумма по mod 2

циклов $\{1, 2, 3\}$ и $\{1, 2, 4\}$ даст $\{3, 4\}$, являющееся независимым множеством матроида M.

В этом разделе мы устанавливаем некоторые свойства бинарных матроидов, которые приведут нас к разработке альтернативной характеризации бинарных матроидов.

Пусть M — матроид на множестве S. Пусть $B=\{b_1, b_2, \ldots, b_r\}$ — база M, а $B^*=S-B=\{e_1, e_2, \ldots, e_q\}$ — соответствующая кобаза M. Напомним, что $B^* \cap C(e_j)=e_j$, $j=1, \ldots, q$, где $C(e_j)$ — базисный цикл e_j. Аналогично $B \cap C^*(b_j)=b_j$, $j=1, \ldots, r$, где $C^*(b_j)$ — базисный коцикл b_j. Как и в случае неориентированных графов (гл. 6), определим *базисную коцикломатическую матрицу* D_f и *базисную цикломатическую матрицу* D_f^* матроида M по отношению к базе B. Очевидно, что матрицы будут иметь вид

$$D_f=\begin{array}{c} \begin{matrix} b_1 b_2 \cdots b_r & & e_1 e_2 \cdots e_q \end{matrix} \\ \left[\begin{array}{c|c} U_r & F \end{array}\right] \end{array},$$

$$D_f^*=\begin{array}{c} \begin{matrix} b_1 b_2 \cdots b_r & & e_1 e_2 \cdots e_q \end{matrix} \\ \left[\begin{array}{c|c} G & U_q \end{array}\right] \end{array}.$$

Заметим, что $D_f=[d_{ij}]$ и $D_f^*=[d_{ij}^*]$ — (0,1)-матрицы.

Допустим, что матроид M — бинарный. Тогда он имеет стандартное представление на $GF(2)$ вида

$$\begin{array}{c} \begin{matrix} b_1 b_2 \cdots b_r & & e_1 e_2 \cdots e_r \end{matrix} \\ \left[\begin{array}{c|c} U_r & A \end{array}\right] \end{array}. \tag{10.10}$$

Пусть базисный цикл $C(e_j)=\{e_j, b_1, b_2, \ldots, b_k\}$. Тогда сумма по mod 2 столбцов матрицы (10.10), соответствующих элементам $C(e_j)$, равна нулю. Поскольку сумма по mod 2 векторов аналогична кольцевой сумме соответствующих множеств, легко видеть, что

$$a_{ij}=\begin{cases} 1, & \text{если } b_i \in C(e_j); \\ 0, & \text{если } b_i \in C(e_j). \end{cases}$$

Другими словами,

$$A=G^t. \tag{10.11}$$

Таким образом, матрица $[U_r G^t]$ является стандартным представлением над $GF(2)$ M и по отношению к базе B.

Исходя из стандартного представления для двойственного матроида M^*, можно аналогично показать, что $[F^t U_q]$ является стандартным представлением над GF (2) матроида M^*.

Поскольку матроид имеет по отношению к данной базе единственное стандартное представление, из следствия 10.24.1 вытекает

$$F=G^t. \tag{10.12}$$

Таким образом, мы доказали следующую теорему:

Теорема 10.25. Пусть M — бинарный матроид на множестве S.

1. Базисная цикломатическая матрица M по отношению к произвольной базе является стандартным представлением этого матроида.
2. Базисная коцикломатическая матрица матроида M по отношению к произвольной базе является стандартным представлением матроида M^*.

Из формулы (10.12) мы получаем также следующий интересный результат.

Теорема 10.26. Пусть M — бинарный матроид. Пусть D_f и D_f^* — базисные коцикломатическая и цикломатическая матрицы матроида M по отношению к общей базе. Тогда

$$D_f(D_f^*)=0. \tag{10.13}$$

Пусть C — цикл матроида M; мы можем связать с ним (0, 1)-вектор-строку, каждый элемент которого соответствует элементу матроида M, а элемент вектора равен 1, если соответствующий элемент множества S входит в цикл C. Например, строки D_f^* —это векторы, соответствующие базисным циклам. Матрица, содержащая все циклические векторы матроида M, называется *цикломатической матрицей* матроида M и обозначается $D^*(M)$. Аналогичным образом определяется *коцикломатическая матрица* $D(M)$ матроида M.

Допустим, что x — вектор, соответствующий циклу C бинарного матроида M. Так как D_f— стандартное представление матроида M, сумма по mod 2 столбцов D_f, соответствующих элементам C, равна 0. Другими словами,

$$D_f x^t=0. \tag{10.14}$$

Аналогично если x — коциклический вектор, то

$$D_f^* x^t=0. \tag{10.15}$$

Заметим, что мы считали расстановку столбцов D_f^*, D_f и x соответствующей одному порядку элементов.

Теорема 10.27. Пусть M — бинарный матроид на S. Пусть B — база, а C — цикл матроида M, $C-B=\{x_1, x_2, \ldots, x_k\}$. Если $C(x_i)$ будет обозначать базисный цикл элемента x_i по отношению к B, то $C=C(x_1)\oplus C(x_2)\oplus \ldots \oplus C(x_k)$.

Доказательство основано на выражениях (10.12) и (10.14). См. теорему 6.7.

Для доказательства обратной теоремы мы нуждаемся в следующем утверждении:

Теорема 10.28. Пусть M — матроид. Пусть для любых базы B и цикла C матроида M имеем $C=C(x_1)\oplus C(x_2)\oplus \ldots \oplus C(x_k)$, где $\{x_1, x_2, \ldots, x_k\}=C-B$, а $C(x_i)$ — базисный цикл x_i по отношению к B. Тогда сумма по mod 2 $C_1\oplus C_2$ двух любых различных циклов матроида M содержит цикл этого матроида.

Доказательство. Допустим, что $C_1\oplus C_2$ не содержит цикла. Пусть $C_1\cap C_2=\{x_1, x_2, \ldots, x_p\}$. Тогда $C_1\oplus C_2=(C_1\cup C_2)-\{x_1, \ldots, x_p\}$ — независимое множество в матроиде M. Поэтому существует база $B\supseteq C_1\oplus C_2$. Но тогда $C_1-B=C_2-B=\{x_1, \ldots, x_p\}$. Следовательно, по предположению теоремы $C_1=C(x_1)\oplus C(x_2)\oplus \ldots \oplus C(x_p)=C_2$, что противоречит условию $C_1\neq C_2$.

Теорема 10.29. Пусть M — матроид. Пусть для любых базы B и цикла C матроида M имеем $C=C(x_1)\oplus C(x_2)\oplus \ldots \oplus C(x_k)$, где $\{x_1, x_2, \ldots, x_k\}=C-B$, а $C(x_i)$ — базисный цикл x_i по отношению к B. Тогда M — бинарный матроид.

Доказательство. Пусть $B=\{b_1, b_2, \ldots, b_r\}$ — произвольная база матроида M, $S-B=\{e_1, e_2, \ldots, e_q\}$. Базисная цикломатическая матрица $D_f^*(M)$ матроида M по отношению к B имеет вид

$$\begin{array}{c} b_1 b_2 \cdots b_r \quad e_1 e_2 \cdots e_q \\ [\quad A \quad | \quad U_q \quad]. \end{array}$$

Для доказательства теоремы покажем, что матрица

$$\begin{matrix} & b_1 b_2 \cdots b_r & & e_1 e_2 \cdots e_q \\ [& U_r & | & A^t &] \end{matrix} \qquad (10.16)$$

является стандартным представлением матроида M над $GF(2)$.

Пусть C — цикл матроида M. Если $C-B=\{e_{i_1}, e_{i_2}, \ldots, e_{i_k}$, то по условиям теоремы $C=C(e_{i_1})\oplus C(e_{i_2})\oplus \ldots \oplus C(e_{i_k})$. Из определения A следует, что сумма по mod 2 столбцов матрицы в формуле (10.16), соответствующих членам цикла C, равна 0. Таким образом, эти столбцы линейно зависимы над $GF(2)$.

Допустим, что $\{b_1, \ldots, b_t, e_1, \ldots, e_p\}$ — цикл матроида M', порожденного на S линейной зависимостью соответствующих вектор-столбцов матрицы (10.16). Тогда $W=C(e_1)\oplus C(e_2)\oplus \ldots \oplus C(e_p)$. По теореме 10.28 W содержит цикл C' матроида M. Но C' не может быть собственным подмножеством цикла W, так как это противоречило бы тому, что W — цикл матроида M'. Таким образом, $W=C'$.

Следовательно, матрица (10.16) есть представление матроида M над $GF(2)$.

Другие характеризации бинарного матроида даны в следующей теореме.

Теорема 10.30. Пусть M — матроид. Следующие утверждения эквивалентны:

1. Для любых цикла C и коцикла C^* матроида M $|C\cap C^*|$ — четно.
2. Сумма по mod 2 произвольного набора различных циклов матроида M есть объединение непересекающихся циклов того же матроида.
3. Если для любых базы B и цикла C матроида M имеем $C-B=\{e_1, e_2, \ldots, e_q\}$, $C(e_i)$ — базисный цикл e_i по отношению к базе B, то $C=C(e_1)\oplus C(e_2)\oplus \ldots \oplus C(e_q)$.
4. M — бинарный матроид.

Доказательство.

$1\Rightarrow 2$. Пусть $C_1, C_2, \ldots, C_k$ — различные циклы матроида M, а $A=C_1\oplus C_2\oplus \ldots \oplus C_k$. Не нарушая общности, допустим, что A не содержит петель.

Поскольку для любого коцикла C^* $|C^*\cap C|$ — четно $(1\leqslant i\leqslant k)$, то легко видеть, что $|A\cap C^*|$ — также четно. Допустим, что A — независимое множество. Тогда приходим к противоречию, поскольку по лемме 10.3 существует коцикл, имеющий точно один элемент из A. Таким образом, A — зависимое множество и содержит цикл C.

Если $A=C$, теорема доказана. Допустим противное: пусть $A_1=A\oplus C$. Заметим, что для любого коцикла C^* имеем, что $|C^*\cap A_1|$ четно. Поэтому мы можем повторить рассуждения для A_1. Поскольку A_1 — конечно и $A_1=A-C$, этот процесс в конце концов закончится, приводя в результате к набору непересекающихся циклов, объединение которых равно A.

$2\Rightarrow 3$. См. доказательство теоремы 4.6.

$3\Rightarrow 4$. Аналогично теореме 10.29.

$4\Rightarrow 1$. По теореме 10.27 всякий цикл C можно выразить через базисные циклы $C=C(x_1)\oplus C(x_2)\oplus \ldots \oplus C(x_k)$. По формуле (10.15) имеем, что $|C^*\cap C(x_i)|$ — четно для всех $1\leqslant i\leqslant k$. Поэтому легко видеть, что $|C^*\cap C|$ — четно.

Очевидно, что заменой циклов на коциклы в теореме 10.30 достигается альтернативная характеризация в терминах коциклов.

10.8. Ориентируемые матроиды

Матроид M называется *ориентируемым*, если некоторым ненулевым элементам в цикломатической матрице $D^*=D^*(M)$ и в коцикломатической матрице $D=D(M)$ можно присвоить отрицательные знаки таким образом, что после этого произведение любой строки цикломатической матрицы на любую строку коцикломатической матрицы будет равно нулю над кольцом целых чисел. Очевидно, что циклический матроид и матроид разрезов графа ориентируемы.

Раскрашивание ориентируемого матроида M — это разбиение его элементов на три множества — R, G и B — и выделение одного из элементов — G. Визуально можно представить это разбиение как раскраску элементов матроида M в три цвета: красный, зеленый и голубой, причем точно один зеленый элемент раскрашен в темно-зеленый цвет. Однако заметим, что этот элемент рассматривается как зеленый.

Основной результат этого раздела — «лемма о раскраске дуг» — получен Минти [10.2].

Теорема 10.31. (Лемма о раскраске дуг.) Пусть M — ориентируемый матроид. Для любого раскрашивания элементов матроида M справедливо точно одно из следующих утверждений:

1. Существует цикл, содержащий темно-зеленый элемент, но не имеющий голубых. Все зеленые элементы в нем ориентированы одинаково (т. е. имеют одинаковый знак в цикломатической матрице).
2. Существует коцикл, содержащий темно-зеленый элемент, но не имеющий красных. Все зеленые элементы в нем ориентированы одинаково.

Доказательство. Докажем индукцией по числу зеленых элементов. Если имеется единственный зеленый элемент, результат следует из аксиомы G.2.

Допустим, что теорема справедлива, когда число зеленых элементов равно m. Тогда рассмотрим раскрашивание, в котором число зеленых элементов равно $m+1$. Выберем зеленый элемент x, отличный от темно-зеленого (dg) (рис. 10.4).

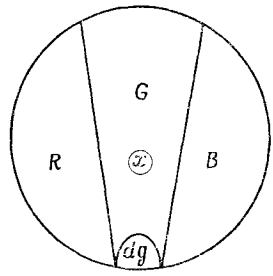


Рис. 10.4.

Раскрасим элемент x в красный цвет. В получившемся раскрашивании стало m зеленых элементов. Если теперь найдется коцикл типа 2, то теорема доказана.

Допустим, мы раскрасили x в голубой цвет. Если в получившемся раскрашивании имеется цикл типа 1, то теорема доказана.

Предположим, что ни одно из этих допущений не имеет места. Тогда по индуктивному предположению мы имеем:

а) существует коцикл типа 2, если x раскрашен в голубой цвет;

б) существует цикл типа 1, если x раскрашен в красный цвет.

Теперь пусть соответствующие строки цикломатической матрицы и коцикломатической матрицы имеют следующий вид:

	dg	R	B	G	x
Коцикл	$+1$	$0\ 0\cdots 0\ \ 0$	$1\ -1\cdots 0\ 1$	$1\ 1\cdots 1\ 0$	?
Цикл	$+1$	$-1\ 1\cdots 0\ -1$	$0\ \ 0\cdots 0\ 0$	$0\ 1\cdots 1\ 0$	?

Не нарушая общности, мы здесь допустили, что в позиции темно-зеленого элемента в обоих векторах стоит $+1$.

По определению произведение двух этих векторов равно нулю. Вклад в произведение от темно-зеленого элемента равен 1; от всех красных и голубых элементов — нулю; от зеленых элементов — неотрицательному целому p и от x — неизвестному целому q, которое равно 0, 1 или —1. Таким образом, $1+p+q=0$. Это равенство выполняется только при $p=0$ и $q=-1$. Поэтому в одном из векторов под знаком вопроса скрывается $+1$, а в другом —1. Выбрав вектор, в котором знак вопроса скрывает 1, получим требуемый цикл или коцикл.

Таким образом, выполняется либо утверждение 1, либо утверждение 2. Оба одновременно они выполняться не могут, поскольку тогда произведение соответствующих циклу и коциклу столбцов не равнялось бы нулю.

Лемма о раскраске дуг в частном случае графов [10.5, 10.6] очевидна, она используется в гл. 11 в доказательстве свойства «неусиления» резисторных схем.

10.9. Матроиды и «жадный» алгоритм

Рассмотрим множество S, элементам которого s_i присвоены неотрицательные веса $w(s_i)$. Вес подмножества S определяется равным сумме весов всех входящих в подмножество элементов. Пусть $\mathcal{F}$ — набор подмножеств S. Многие задачи комбинаторной оптимизации сводятся к следующей задаче:

Найти член $\mathcal{F}$ максимального веса.

Например, к этой задаче сводится нахождение остова максимального веса во взвешенном связном графе G, если $\mathcal{F}$ — набор всех остовов графа G.

Для решения этой задачи естественно применить следующий алгоритм, называемый «жадным».

Алгоритм 10.1. «Жадный» алгоритм.

Шаг 1. Выбрать такой элемент s_1, что $\{s_1\} \in \mathcal{F}$ и $w(s_1) \geqslant w(s)$ для всех таких s, что $\{s\} \in \mathcal{F}$. Если такого s_1 не существует — останов.

Шаг 2. Выбрать такой элемент s_2, что $\{s_1, s_2\} \in \mathcal{F}$ и $w(s_2) \geqslant w(s)$ для всех таких $s \neq s_1$, что $\{s_1, s\} \in \mathcal{F}$. Если такого s_2 не существует, останов.

Шаг 3. Выбрать такой отличный от $s_1, s_2, \ldots, s_{k-1}$ элемент s_k, что $\{s_1, s_2, \ldots, s_{k-1}, s_k\} \in \mathcal{F}$ и $w(s_k)$ — максимально среди всех таких s. Если такого s_k не существует — останов.

Очевидно, что алгоритм заканчивает работу, получая максимальный по включению член $\mathcal{F}$. Однако этот член может иметь не максимальный вес в $\mathcal{F}$.

Например, если $S=\{a, b, c, d\}$, $w(a)=4$, $w(b)=3$, $w(c)=2$, $w(d)=2$ и $\mathcal{F}=\{\{a\}, \{a, c\}, \{b, c, d\}, \{b, d\}\}$, то алгоритм получит $\{a, c\}$, но членом $\mathcal{F}$ максимального веса является $\{b, c, d\}$. Однако если мы изменим веса так, чтобы $w(a)=6$, $w(b)=3$, $w(c)=2$, $w(d)=2$, то алгоритм снова получит $\{a, c\}$, являющееся теперь членом $\mathcal{F}$ максимального веса.

Исследуем взаимосвязь получаемого «жадным» алгоритмом решения со структурой $\mathcal{F}$.

Рассмотрим матроид M на множестве S. Пусть $\mathcal{F}$ — набор независимых множеств M. Пусть $I_1=\{a_1, a_2, \ldots, a_m\}$ и $I_2=\{b_1, b_2, \ldots, b_n\}$ — независимые множества, элементы которых расставлены в порядке невозрастания весов. Таким образом, $w(a_1)\geqslant w(a_2)\geqslant\ldots\geqslant w(a_m)$ и $w(b_1)\geqslant w(b_2)\geqslant\ldots\geqslant(b_n)$. В этом случае I_1 *лексикографически больше* I_2, если существует такое k, что $w(a_i)=w(b_i)$ для $1\leqslant i\leqslant k-1$ и $w(a_k)>w(b_k)$ или $w(a_i)=w(b_i)$ для $1\leqslant i\leqslant n$ и $m>n$. Множество, которое лексикографически не меньше любого другого множества, называется *лексикографически максимальным*. Из этого определения следует, что лексикографически максимальное независимое множество должно быть базой.

Множество $B\in\mathcal{F}$ называется *оптимальным по Гейлу* в $\mathcal{F}$, если для всякого $I\in\mathcal{F}$ существует такое взаимно однозначное соответствие между I и B, что для всех $a\in I$ $w(a)\leqslant w(b)$, где b — элемент B, соответствующий a. Очевидно, что оптимальными по Гейлу могут быть только базы. Кроме того, если база оптимальна по Гейлу, то она должна иметь максимальный вес.

Далее мы принимаем, что элементы всякого множества упорядочены по невозрастанию весов.

Теорема 10.32. Пусть $\mathcal{F}$ — набор независимых множеств матроида M на S, а B — база M. Для произвольного взвешивания элементов эквивалентны следующие утверждения:

1. B лексикографически максимально в $\mathcal{F}$.
2. B оптимально по Гейлу в $\mathcal{F}$.
3. B член $\mathcal{F}$ максимального веса.

Доказательство.

$1\Rightarrow 2$. Пусть $B=\{b_1, b_2, \ldots, b_r\}$ — лексикографически максимальная база матроида M. Допустим, что B не оптимальна по Гейлу. Тогда существует такое независимое множество $I=\{a_1, a_2, \ldots, a_k\}$, что $w(a_i)=w(b_i)$ для $1\leqslant i\leqslant k-1$ и $w(a_k)>w(b_k)$. Рассмотрим независимые множества $B_{k-1}=\{b_1, b_2, \ldots, b_{k-1}\}$ и $I=\{a_1, a_2, \ldots, a_k\}$. По аксиоме 1.3 существует такой $a_j\in I$, что $I'=\{b_1, b_2, \ldots, b_{k-1}, a_j\}$ — независимое множество. Но I' лексикографически больше B, поскольку $w(a_j)\geqslant w(a_k)>w(b_k)$, что противоречит лексикографической максимальности B в $\mathcal{F}$.

$2\Rightarrow 3$. Очевидно.

$3\Rightarrow 1$. Пусть $B=\{b_1, b_2, \ldots, b_r\}$ — лексикографически максимальная база, а B' — база максимального веса. Поскольку справедливо $1\Rightarrow 2$, то $w(b_i)\geqslant w(b_i')$ для всех $1\leqslant i\leqslant r$. Из того, что B' — база максимального веса, следует, что $w(b_i)=w(b_i')$. Таким образом, B' также лексикографически максимальна.

Легко показать (аналогично тому, как это сделано в доказательстве теоремы 10.32), что применение «жадного» алгоритма дает лексикографически максимальную базу, которая по только что доказанной теореме имеет максимальный вес. Таким образом, справедливо следующее утверждение:

Теорема 10.33. Пусть $\mathcal{F}$ — набор независимых множеств матроида на S, элементам которого присвоены неотрицательные веса. Применение «жадного» алгоритма к набору $\mathcal{F}$ дает член набора $\mathcal{F}$ максимального веса.

Из этой теоремы очевидно, что база максимального веса получается, если выбирать элементы матроида в порядке невозрастания весов, отвергая только те элементы, выбор которых нарушает условие независимости получаемого множества. Применение «жадного» алгоритма для получения базы минимального веса очевидно.

Рассмотрим, например, взвешенный граф G на рис. 10.5. Веса ребер приведены на рисунке. Для применения «жадного» алгоритма к получению остова графа G максимального веса необходимо сначала расположить ребра в порядке невозрастания весов. Таким образом, ребра будут упорядочены следующим образом: a, b, e, f, d, c, g, h.

С помощью алгоритма будут получены сначала три ребра a, b и e, поскольку они не образуют цикла. Ребро f будет отклонено, так как множество $\{a, b, e\} \cup \cup \{f\}$ содержит цикл. Следовательно, будет выбрано ребро d. Ребро c будет отклонено, поскольку оно образует цикл с уже выбранными ребрами e и d. По этой

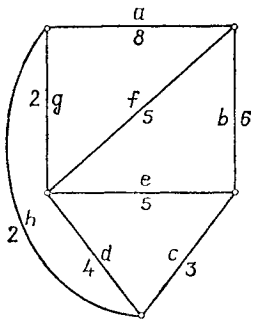


Рис. 10.5. Взвешенный граф.

же причине будут отклонены ребра g и h. Таким образом, «жадный» алгоритм порождает множество $\{a, b, e, d\}$, являющееся остовом графа G максимального веса.

Докажем сейчас теорему, обратную теореме 10.33.

Теорема 10.34. Пусть $\mathcal{F}$ — набор подмножеств множества S, обладающий таким свойством, что если $A \in \mathcal{F}$ и $B \subseteq A$, то $B \in \mathcal{F}$. Тогда применение к набору F «жадного» алгоритма дает член набора $\mathcal{F}$ максимального веса, только если $\mathcal{F}$ — набор независимых множеств матроида на S.

Доказательство. Из аксиом независимости очевидно, что нам необходимо показать, что если $A = \{a_1, a_2, \ldots, a_k\} \in \mathcal{F}$, а $B = \{b_1, b_2, \ldots, b_{k+1}\} \in \mathcal{F}$, то такой существует $b_i \notin A$, что $A \cup b_i \in \mathcal{F}$.

С этой целью определим веса элементов S следующим образом: $w(a_i) = 1$, $1 \leqslant i \leqslant k$, $w(b_i) = x$, $b_i \in B - A$, $w(e) = 0$, $e \in S - (A \cup B)$, где $0 < x < 1$. Тогда «жадный» алгоритм выберет сначала элементы $a_1, a_2, \ldots, a_k$. Если не существует такого b_i, что $\{b_i, a_1, \ldots, a_k\} \in \mathcal{F}$, алгоритм выберет элементы из $S - (A \cup B)$. Поэтому, когда алгоритм закончит работу, в результате получится множество, имеющее вес, равный весу A.

Если $|B \cap A| = t$, то $w(A) = k$ и $w(B) = t + (k + 1 - t)\, x$. Очевидно, что можно выбрать $0 < x < 1$ таким, что $w(A) < w(B)$. Но тогда «жадный» алгоритм не породил члена набора $\mathcal{F}$ максимального веса, т. е. мы пришли к противоречию.

Применение «жадного» алгоритма к задаче нахождения остова максимального веса было впервые предложено в работе [10.7]. Расширение применения этого алгоритма на матроиды было предложено в работах [10.8—10.11].

10.10. Замечания, касающиеся литературы

Прекрасными работами, содержащими богатую информацию по теории матроидов, являются работы [10.4, 10.12]. В работе [10.13] имеется глава по матроидам. Уилсон [10.14] дает элегантное введение в теорию матроидов. Он раскрывает силу общности матроидов,

включая простые доказательства двух теорем по реберно непересекающимся остовам графа. Кроме первоначальной статьи Уитни [10.1] мы очень рекомендуем для дальнейшего изучения работы [10.15—10.27].

Татт [10.3, 10.17, 10.18] разработал теорию цепных групп и матроидов. Он определил матроид регулярным, если тот изоморфен матроиду регулярной цепной группы. Можно показать, что матроид регулярен тогда и только тогда, когда он представим над всяким полем. В другой характеризации матроид *регулярен* тогда и только тогда, когда он ориентируем [10.2]. Татт также предложил необходимые и достаточные условия того, что матроид графический. Бинарный матроид является графическим тогда и только тогда, когда он не содержит в качестве минора матроид Фано или двойственный ему, или $M^*(K_5)$, или $M^*(K_{3,3})$. Дальнейшее обсуждение этого вопроса можно найти в работах [10.4, 10.19]. (Определение матроида Фано дано в упражнении 10.22.)

Теория связности матроидов дана в работе [10.20]. Теория ориентируемости в применении к матроидам общего вида дается в работах [10.28, 10.29].

Леман [10.16] дал решение игры переключений Шеннона, используя понятие матроида. См. также работы [10.4, 10.30, 10.31].

Красота теории матроидов заключается в ее универсальной природе, приводящей к простым доказательствам результатов теории трансверсалей и теории графов. См. также работы [10.14, 10.24, 10.30].

Работа [10.27] интересна рассмотрением алгоритмов, связанных с матроидами. См. также работы [10.4, 10.32, 10.33].

Теория матроидов находит в настоящее время все возрастающее применение в теории электрических схем. См. работы [10.34—10.43]. Бруно и Вейнберг [10.37] дают хорошее введение в теорию матроидов.

Упражнения

10.1. Пусть M — матроид на S, а $A \subseteq S$. Определим $\mathcal{F}'$ как набор таких подмножеств $X \subseteq S$, что X — независимое множество в M и $X \cap A = \varnothing$. Докажите, что $\mathcal{F}'$ — набор независимых множеств матроида на S.

10.2. Пусть множество S имеет n элементов. Покажите, что набор $\mathcal{F}$ всех подмножеств S, имеющих k или менее элементов, есть множество независимых множеств матроида. Этот матроид называется *однородным матроидом* ранга k и обозначается через $U_{k,n}$.

10.3. Пусть M — матроид на S, а B_1, B_2 — различные базы M. Докажите, что существует взаимно однозначное соответствие между B_1 и B_2, при котором для любого $e \in B_1$ $(B_2 - e') \cup e$ является базой M, где $e' \in B_2$ соответствует e.

10.4. Докажите, что если B_1, B_2 — базы матроида M и $X_1 \subseteq B_1$, то существует такое $X_2 \subseteq B_2$, что $(B_1 - X_1) \cup X_2$ и $(B_2 - X_2) \cup X_1$ являются базами M [10.44]; см. также [10.12], глава V.

10.5. Пусть D — набор таких непустых подмножеств S, что для двух любых членов X и Y набора $\mathcal{D}$ $(x \in X \cap Y,\ y \in X - Y)$ существует такое $Z \in \mathcal{D}$, что $y \in Z \subseteq$

$\subseteq(X \cup Y)-x$. Докажите тогда, что набор $\mathscr{D}'$ минимальных членов $\mathscr{D}$ образует множество циклов матроида [10.3].

10.6. Докажите, что если C — цикл матроида M, а $a \in C$, то существует такая база B, что $C=C(a, B)$.

10.7. Докажите, что если B — база матроида M и $x \in B$, то имеется точно один такой коцикл C^* матроида M, что $C^* \cap (B-x)=\varnothing$.

10.8. Докажите, что если C — цикл матроида M и x, y — его различные элементы, то существует коцикл C^*, содержащий x, y и никакого другого элемента цикла C [10.2].

10.9. Пусть M — матроид на S, а x, y, z — различные элементы S. Докажите, что если существуют циклы C_1, содержащий x и y, и C_2, содержащий y и z, то существует и цикл C_3, содержащий x и z.

10.10. Множество $A \subseteq S$ называется *замкнутым* в матроиде M на S, если для всех $x \in S-A$ $\rho=(A \cup x)=\rho(A)+1$. Покажите, что пересечение двух замкнутых множеств есть замкнутое множество.

10.11. Пусть M — матроид на множестве S. *Замыканием* $\sigma(A)$ подмножества $A \subseteq S$ называется множество всех таких элементов $x \in S$, что $\rho(A \cup x)=\rho(A)$. Докажите, что а) если x входит в $\sigma(A \cup y)$, но не входит в $\sigma(A)$, то y входит в $\sigma(A \cup x)$. б) элемент x принадлежит $\sigma(A)$ тогда и только тогда, когда $x \in A$ или существует цикл C матроида M, для которого $C-A=\{x\}$.

10.12. *Гиперплоскостью* матроида M на S называется максимальное собственное замкнутое подмножество S. Покажите, что H является гиперплоскостью тогда и только тогда, когда $S-H$ — коцикл матроида M (систему аксиом для матроида, исходя из гиперплоскостей, можно найти в работе [10.4]).

10.13. Покажите, что если M — матроид на S и $A \subseteq S$, то сужение $M \cdot T$ матроида M на T — матроид, коциклы которого являются в точности коциклами матроида M, содержащимися в A.

10.14. Докажите, что если M — матроид на S и $T \subseteq X \subseteq S$, то а) $M|T=(M|X)|T$; б) $M \cdot T=(M \cdot X) \cdot T$; в) $(M|X) \cdot T=(M \cdot (S-(X-T)))|T$; г) $(M-X)|T=(M|S-(X-T)) \cdot T$.

10.15. Матроид M на S называется *связным* или *неразделимым*, если для любой пары различных элементов x и y, принадлежащих S, существует цикл матроида M, содержащий x и y. В противном случае M называется *несвязным* или *разделимым*. Покажите, что матроид является связным тогда и только тогда, когда связен двойственный к нему матроид M^*.

Примечание. Если G — граф, то $M(G)$ связный тогда и только тогда, когда граф G является 2-связным.

10.16. Покажите, что матроид M на S связный тогда и только тогда, когда существует такое собственное подмножество $A \subset S$, что $\rho(A)+\rho(S-A)=\rho(S)$ [10.1].

10.17. Доказать или опровергнуть: граф G стягивается к H тогда и только тогда, когда $M(G)$ содержит $M(H)$ в качестве минора сужения.

10.18. Докажите, что однородный матроид $U_{2,4}$ представим над всяким полем, за исключением $GF(2)$.

10.19. Докажите, что матроид бинарный тогда и только тогда, когда для любых цикла C и коцикла C^* справедливо $|C \cap C^*| \neq 3$.

10.20. Пусть $\mathscr{F}$ — семейство непустых подмножеств множества S. Трансверсаль подсемейства $\mathscr{F}$ называется *частичной трансверсалью* $\mathscr{F}$. Покажите, что если $\mathscr{F}$ — набор непустых подмножеств множества S, то набор частичных трансверсалей $\mathscr{F}$ образует множество независимых множеств матроида на S. (Определение трансверсали дано в разд. 8.6.) Найдите функцию ранга этого матроида. Матроид M на S называется *трансверсальным*, если существует такое семейство $\mathscr{F}$ подмножеств S, что $\mathscr{F}(M)$ есть семейство частичных трансверсалей $\mathscr{F}$. Найдите функцию ранга трансверсального матроида (см. теорему 8.15).

10.21. Покажите, что любой однородный матроид ранга k является трансверсальным.

10.22. Матроид Фано F — это матроид, определенный на множестве $S=\{1, 2, 3, 4, 5, 6, 7\}$, базами которого являются все подмножества S, содержащие по три

элемента, за исключением {1, 2, 3}, {1, 4, 5}, {1, 6, 7}, {2, 4, 7}, {2, 5, 6}, {3, 4, 6} и {3, 5, 7}. Покажите, что матроид F является а) бинарным, б) трансверсальным, в) неграфическим, г) некографическим.

10.23. Покажите, что циклический матроид K_4 не является трансверсальным.

10.24. Матроид M на S называется *эйлеровым*, если S можно представить в виде объединения непересекающихся циклов. Матроид называется *двудольным*, если всякий его цикл содержит четное число элементов. Докажите, что матроид M двудольный тогда и только тогда, когда матроид M^* эйлеров.

10.25. Пусть D — ориентированный граф без петель, а X и Y — непересекающиеся множества вершин D. Подмножество $A \subseteq X$ называется независимым, если существует $|A|$ вершинно непересекающихся цепей от A к Y. Покажите, что эти независимые множества образуют независимые множества матроида на X (такой матроид называется *гаммоидом*) [10.45].

10.26. Матроид M на S называется *правильным*, если для любых баз B_1, B_2 матроида M существует такое взаимно однозначное соответствие между B_1 и B_2, что для любого $x \in B_1$ $(B_1 - x) \cup x'$ и $(B_2 - x') \cup x$ являются базами матроида M, где x' — элемент B_2, соответствующий x. Покажите, что а) $M(K_4)$ — неправильный матроид; б) если M — правильный матроид на S и $T \subseteq S$, то матроид $M|T$ правильный; в) если M — правильный матроид, то любой его минор правильный.

10.27. Пусть M_1 и M_2 — матроиды на множестве S.

а) Покажите, что множество всех объединений $I \cup J$ независимых множеств I матроида M_1 и независимых множеств J матроида M_2 образует набор независимых множеств нового матроида. (Этот матроид называется *объединением* M_1 и M_2 и обозначается $M_1 \cup M_2$.)

б) Покажите, что если ρ_1 и ρ_2 — функции ранга матроидов M_1 и M_2 на множестве S, то $\rho(A) = \min\limits_{X \subseteq A} \{\rho_1(X) + \rho_2(X) + |A - X|\}$, где $A \subseteq S$, ρ — функция ранга $M_1 \cup M_2$ [10.14].

10.28. Пусть M — матроид на S. Докажите, что а) M содержит k непересекающихся баз тогда и только тогда, когда для любого $A \subseteq S$ $\rho(A) + |S - A| \geqslant k\rho(S)$; б) S можно выразить в виде объединения не более чем k независимых множеств тогда и только тогда, когда для любого $A \subseteq S$ $k\rho(A) \geqslant |A|$.

Примечание. Рассмотрите объединение k копий матроида M и используйте результат упражнения 10.27 [10.14].

10.29. Покажите, что если «жадным» алгоритмом выбраны k элементов, то они имеют максимальный вес среди всех независимых множеств, состоящих из k или менее элементов.

10.30. Необходимо выполнить на ЭВМ множество заданий. Все задания требуют для выполнения одинакового времени. Каждому заданию присваивается крайний срок выполнения.

а) Покажите, что набор всех подмножеств заданий, которые можно выполнить по расписанию, образует множество всех независимых множеств матроида.

б) Допустим, за каждое не выполненное в срок задание необходимо заплатить штраф. В каком порядке следует выполнять задания, чтобы общий штраф был минимальным?

10.31. Пусть M — матроид, элементам которого присвоены неотрицательные веса. Докажите, что

а) никакой элемент базы максимального веса не имеет наименьшего веса в любом цикле M;

б) каждый элемент базы максимального веса имеет наибольший вес по крайней мере в одном коцикле матроида M.

Используя п. б., предложите процедуру построения базы матроида максимального веса. (В работе [10.46] описывается такая процедура для построения остова минимального веса в связном графе.)

10.32. Пусть M — матроид на S, элементам которого присвоены неотрицательные веса. Пусть $\mathcal{B}$ — набор баз матроида M, а $\mathcal{C}^*$ — набор коциклов этого же матроида M. Докажите, что $\min\limits_{B \in \mathcal{B}} \max\limits_{e \in B} w(e) = \max\limits_{C^* \in \mathcal{C}^*} \min\limits_{e \in C^*} w(e)$.

Часть II. Теория электрических цепей

11. Графы и электрические цепи

Электрическая цепь есть соединение таких элементов электрической цепи, как резисторы, емкости, индуктивности, а также источники напряжения и тока. Каждому элементу цепи сопоставляются две переменные: переменная напряжения $v(t)$ и переменная тока $i(t)$. Для этих переменных необходимо определить исходные направления, так как они являются функциями времени и могут принимать положительные и отрицательные значения. Это делается указанием стрелкой, названной *направлением*, для каждого элемента цепи (рис. 11.1). Эта стрелка означает, что величина $i(t)$ положительна,

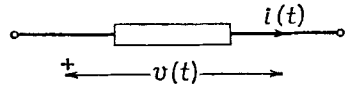


Рис. 11.1. Представление элемента цепи.

если ток течет по направлению стрелки. Далее принимаем, что полярность напряжения $v(t)$ считается положительной у оперения стрелки. Таким образом, величина $v(t)$ является положительной, когда падение напряжения в элементе цепи направлено вдоль стрелки. Элементы цепи характеризуются физическими соотношениями между переменными тока и напряжения. Иногда может оказаться необходимым, чтобы для некоторых элементов цепи были заданы значения напряжения, а для других — значения токов. Такие элементы называются *источниками напряжения* и *тока* соответственно.

Двумя фундаментальными законами теории электрических цепей являются законы Кирхгофа, которые можно сформулировать следующим образом:

Закон Кирхгофа для токов (ЗКТ). Алгебраическая сумма токов, вытекающих из узла, равна нулю.

Закон Кирхгофа для напряжений (ЗКН). Алгебраическая сумма напряжений в любом замкнутом контуре равна нулю.

Например, для цепи, показанной на рис. 11.2, *а*, уравнения ЗКТ и ЗКН приведены ниже:

Уравнения ЗКТ

Узел a $\quad i_1 - i_5 + i_6 = 0,$

Узел c $\quad -i_2 + i_4 - i_6 = 0,$

Узел b $\quad -i_1 + i_2 + i_3 = 0.$

Уравнения ЗКН

Контур $\{1, 3, 5\}$ $v_1+v_3+v_5=0$,
Контур $\{2, 4, 3\}$ $v_2+v_4-v_3=0$,
Контур $\{1, 6, 2\}$ $-v_1+v_6-v_2=0$.

Для заданной электрической цепи N задача анализа состоит в том, чтобы определить напряжения и токи в элементах, которые удовлетворяют законам Кирхгофа, и соотношения ток — напряжение, характеризующие различные элементы, образующие цепь.

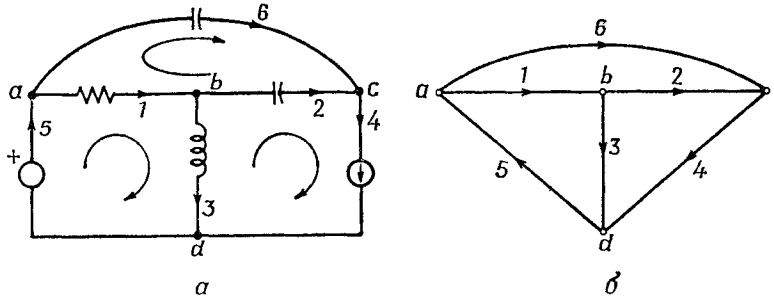


Рис. 11.2. Представление цепи ориентированным графом.
а — цепь N; *б* — ориентированный граф N.

Отметим, что уравнения, которые возникают при применении законов Кирхгофа, носят алгебраический характер и зависят только от взаимосвязи элементов в цепи, а не от их природы. Существует несколько свойств электрической цепи, которые зависят от ее структуры. Изучая такие свойства, удобнее рассматривать каждый элемент цепи как ориентированное ребро, связанное с двумя переменными: $v(t)$ и $i(t)$. Таким образом, можно рассматривать схему электрической цепи как ориентированный граф, в котором каждому ребру сопоставлены две переменные: $v(t)$ и $i(t)$, необходимые для выполнения законов Кирхгофа и некоторых определенных физических соотношений. Например, ориентированный граф, соответствующий цепи, представленной на рис. 11, 2, *а*, изображен на рис. 11.2, *б*.

Легко видеть, что уравнения ЗКТ и ЗКН для цепи N можно записать как $Q_c I_e=0$ и $B_c V_e=0$ соответственно, где Q_c и B_c — матрица разрезов и цикломатическая матрица ориентированного графа, сопоставленного цепи N, а I_e и V_e — вектор-столбцы токов и напряжений соответственно на элементах цепи N. Во всех рассуждениях в этой и последующих главах будем обозначать как электрическую цепь, так и соответствующий ей ориентированный граф одним и тем же символом. Чаще всего граф будет пониматься как цепь и наоборот. Таким образом, такие относящиеся к графу определения, как связность, цикломатическое число, ранг, оказываются применимыми и к цепи. Также будем понимать под *узлом* вершину.

В этой главе мы обсудим некоторые аспекты анализа цепей, которые в значительной степени зависят от теории графов. Нашей основной задачей здесь будет подчеркнуть полезность теории графов при систематическом построении уравнений цепи и раскрытии некоторых фундаментальных свойств электрических цепей.

11.1. Преобразование контуров и сечений

В этом разделе исследуем соотношения между токами и напряжениями на элементах цепи N. Эти соотношения вытекают из законов Кирхгофа и отношения ортогональности между цикломатической матрицей и матрицей сечений цепи N. Не нарушая общности, можно считать, что граф N — связный.

Пусть T является остовом цепи N, а B_f и Q_f обозначают фундаментальную цикломатическую матрицу и матрицу сечений по отношению к T. Тогда уравнения Кирхгофа для токов и напряжений будут иметь вид

$$Q_f I_e = 0, \tag{11.1}$$

$$B_f V_e = 0. \tag{11.2}$$

Предположим, что I_e и V_e разделены следующим образом:

$$I_e = \begin{bmatrix} I_c \\ I_t \end{bmatrix} \quad \text{и} \quad V_e = \begin{bmatrix} V_c \\ V_t \end{bmatrix},$$

где векторы, отвечающие хордам и ветвям остова T, отличаются подстрочными индексами c и t соответственно. Тогда выражения (11.1) и (11.2) можно записать в виде

$$[Q_{fc} \quad U] \begin{bmatrix} I_c \\ I_t \end{bmatrix} = 0, \tag{11.3}$$

$$[U \quad B_{ft}] \begin{bmatrix} V_c \\ V_t \end{bmatrix} = 0. \tag{11.4}$$

Заметим, что [выражение (6.13)]

$$Q_{fc} = -B_{ft}^t. \tag{11.5}$$

Сначала рассмотрим выражение (11.3). Из него получаем

$$I_t = -Q_{fc} I_c = B_{ft}^t I_c. \tag{11.6}$$

Таким образом, ток I_e можно представить в виде

$$I_e = \begin{bmatrix} I_c \\ I_t \end{bmatrix} = \begin{bmatrix} U \\ B_{ft}^t \end{bmatrix} I_c = B_f^t I_c. \tag{11.7}$$

Исходя из выражения (11.4), аналогично можно показать, что

$$V_e = Q_f^t V_t. \tag{11.8}$$

В результате имеем следующую теорему:

Теорема 11.1. 1. Все токи, текущие через элементы электрической цепи N, можно выразить линейной комбинацией хордовых токов, т. е. токов, связанных с хордами остова N.

2. Все напряжения на элементах электрической цепи N можно выразить линейной комбинацией напряжений на ветвях, т. е. напряжений, связанных с ветвями остова цепи N.

Для иллюстрации выражений (11.7) и (11.8) рассмотрим цепь N, представленную на рис. 11.2.

Матрицами B_f и Q_f по отношению к остову T, состоящему из элементов 1, 4 и 5, будут матрицы

$$B_f = \begin{array}{c} \begin{matrix} 2 & 3 & 6 & 1 & 4 & 5 \end{matrix} \\ \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right] \end{array},$$

$$Q_f = \begin{array}{c} \begin{matrix} 2 & 3 & 6 & 1 & 4 & 5 \end{matrix} \\ \left[\begin{array}{ccc|ccc} -1 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & -1 & 0 & 1 & 0 \\ -1 & -1 & -1 & 0 & 0 & 1 \end{array}\right] \end{array}.$$

Тогда I_e и V_e можно выразить следующим образом:

$$\begin{bmatrix} i_2 \\ i_3 \\ i_6 \\ i_1 \\ i_4 \\ i_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} i_2 \\ i_3 \\ i_6 \end{bmatrix},$$

$$\begin{bmatrix} v_2 \\ v_3 \\ v_6 \\ v_1 \\ v_4 \\ v_5 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 0 & -1 \\ 0 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_4 \\ v_5 \end{bmatrix}.$$

Теорема 11.2. Пусть N — электрическая цепь ранга ρ с цикломатическим числом μ. Пусть B — матрица, полученная из μ любых независимых строк цикломатической матрицы цепи N, а Q — матрица, полученная из ρ любых независимых строк матрицы разрезов цепи N.

1. Контурное преобразование. Вектор-столбец I_e удовлетворяет уравнениям ЗКТ цепи N тогда и только тогда, когда существует такой вектор столбец I' из μ элементов, что

$$I_e = B^t I'. \tag{11.9}$$

2. Преобразование сечений. Вектор-столбец V_e удовлетворяет уравнениям ЗКН цепи N тогда и только тогда, когда существует такой вектор-столбец V' из ρ элементов, что

$$V_e = Q^t V'. \tag{11.10}$$

Доказательство. 1. Пусть B_f — базисная цикломатическая матрица цепи. Тогда существует такая невырожденная матрица D, что

$$B_f = DB. \tag{11.11}$$

Если I_e удовлетворяет уравнениям ЗКТ, то из выражения (11.7) имеем $I_e = B_f^t I_c$. Подставляя (11.11) в приведенное выше уравнение и полагая $I' = D^t I_c$, получим $I_e = B^t I'$. С другой стороны, если существует такой ток I', что $I_e = B^t I'$, то $QI_e = (QB^t)I' = 0$ по теореме 6.6 и ЗКТ выполнен.

Доказательство п. 2 следует из принципа двойственности. Уравнения (11.9) и (11.10) известны как контурное преобразование и преобразование сечения соответственно. Элементы I' и V' называются *переменными контуров* и *сечений* соответственно. В общем случае переменные контуров и сечений являются линейными комбинациями на ветвях хордовых токов и напряжений соответственно. Однако, если использовать B_f и (Q_f) в контурном преобразовании (сечения), то хордовые токи (напряжения на ветвях) становятся переменными контура (сечения). Преобразование

$$V_e = A^t V', \tag{11.12}$$

где A — усеченная матрица инциденции, называется *узловым преобразованием*. Очевидно, что это специальный случай преобразования сечений. Если v_r является узлом, по которому усечена матрица A, то элементы V' можно определить как напряжения на всех узлах (кроме v_r) по отношению к v_r. Об этих напряжениях говорят как об *опорных* или *базисных* на узлах напряжениях или просто как о *напряжениях на узлах* цепи.

Мы заканчиваем этот раздел интересным результатом, полученным Теллеженом [11.1], который просто следует из контурного преобразования и преобразования сечений.

Теорема 11.3 (Теллежен). Рассмотрим такие две электрические цепи N и $\hat{N}$, что соответствующие им графы являются идентичными. Пусть V_e и Ψ_e — векторы напряжений на элементах цепей N и $\hat{N}$ соответственно, а I_e и Λ_e — соответствующие векторы токов в элементах.

Тогда 1) $V_e^t \Lambda_e = 0$, 2) $I_e^t \Psi_e = 0$.

Доказательство. 1. Пусть B_f и Q_f обозначают базисную цикломатическую матрицу и матрицу сечений цепи N по отношению к остову T. Так как граф N такой же, как и $\hat{N}$, ясно, что $\hat{N}$ имеет такие же матрицы B_f и Q_f по отношению к T. Из контурного преобразования и преобразования сечений имеем $V_e = Q_f^t V_t$ и $\Lambda_e = B_f^t \Lambda_c$. Таким образом, $V_e^t \Lambda_e = V_t^t (Q_f B_f^t) \Lambda_c = 0$, согласно теореме (6.6).

Доказательство п. 2 следует из принципа двойственности. Теорема Теллежена является очень глубоким результатом в теории цепей с несколькими приложениями [11.2, 11.3]. В гл. 13 обсуждаются применения этой теоремы при вычислении чувствительности цепи с использованием понятия сопряжения сети [11.4—11.6].

11.2. Системы контурных уравнений и уравнений сечений

Как мы уже говорили раньше, задачей анализа цепи является определение токов и напряжений, связанных с элементами электрической цепи. Эти напряжения и токи можно определить из уравнений Кирхгофа и отношений ток — напряжение (кратко v—i) на элементах цепи. Однако эти уравнения содержат большее число переменных. Далее, как видно из теоремы 11.2, вместо уравнений ЗКТ можно использовать контурное преобразование, которое включает в качестве переменных только хордовые токи. Аналогично уравнения ЗКН можно заменить преобразованием сечения, которое включает только переменные напряжения на ветвях. Можно использовать преимущества этих преобразований для построения различных систем уравнений цепи, которые включают в качестве переменных только подмножество напряжений и (или) токов. Две такие системы уравнений, известные как контурная система и система сечения, мы исследуем в этом разделе.

При выводе контурной системы используем контурное преобразование вместо ЗКТ, и контурные переменные в этом случае будут служить в качестве независимых. При выводе системы сечений используем преобразование сечения вместо ЗКН, и переменные сечения будут служить в качестве независимых в данном случае.

Рассмотрим связную электрическую цепь N. Допустим, что цепь N состоит только из резисторов R, емкостей C и индуктивностей L, включая взаимные индуктивности, и независимые источники тока и напряжения. Допустим также, что начальные токи индуктивностей и напряжения емкостей заменены соответствующими источниками. Далее, переменные токов и напряжений являются преобразованием Лапласа комплексных частотных переменных.

В цепи N не может быть контура, состоящего только из независимых источников напряжения. Если бы такой контур имелся, то в соответствии с ЗКТ имелось бы и линейное соотношение между соответствующими напряжениями, нарушающее независимость источников напряжения. По этой же причине в цепи N не может быть сечения, состоящего только из независимых источников тока. Поэтому, следуя теореме 10.12, в цепи N существует остов, содержащий все источники напряжения, но не содержащий источников тока. Такой остов — отправная точка для построения систем как контурных уравнений, так и уравнений сечения.

Сначала выведем контурную систему. Пусть T — такой остов данной цепи, что он содержит все источники напряжения, но не

содержит источников тока. Разделим вектор V_e напряжений на элементах I_e токов в элементах следующим образом:

$$V_e=\begin{bmatrix}V_1\\V_2\\V_3\end{bmatrix} \text{ и } I_e=\begin{bmatrix}I_1\\I_2\\I_3\end{bmatrix},$$

где подстрочные индексы 1, 2 и 3 относятся к векторам, соответствующим источникам тока, RLC-элементам и источникам напряжения. Пусть B_f — фундаментальная цикломатическая матрица цепи N по отношению к T. Уравнение ЗКН для цепи N можно записать следующим образом:

$$B_fV_e=\begin{bmatrix}U & B_{12} & B_{13}\\0 & B_{22} & B_{23}\end{bmatrix}\begin{bmatrix}V_1\\V_2\\V_3\end{bmatrix}=[0],$$

т. е.

$$V_1=-B_{12}V_2-B_{13}V_3, \tag{11.13}$$

$$B_{22}V_2=-B_{23}V_3. \tag{11.14}$$

Вместо ЗКТ можно использовать контурное преобразование

$$\begin{bmatrix}I_1\\I_2\\I_3\end{bmatrix}=\begin{bmatrix}U & 0\\B_{12}^t & B_{22}^t\\B_{13}^t & B_{23}^t\end{bmatrix}\begin{bmatrix}I_1\\I_l\end{bmatrix},$$

где I_l обозначает вектор токов, связанный с хордами в остове T, не содержащем источников.

Из этих уравнений получаем

$$I_2=B_{12}^tI_1+B_{22}^tI_l, \tag{11.15}$$

$$I_3=B_{13}^tI_1+B_{23}^tI_l. \tag{11.16}$$

Заметим, что среди хордовых токов должны быть определены только те, что входят в I_l. Если Z_2 является матрицей полных сопротивлений RLC-элементов, то отношения v—i для этих элементов можно записать в виде

$$V_2=Z_2I_2. \tag{11.17}$$

Используя выражения (11.17) в (11.14), получим $B_{22}Z_2I_2=-B_{23}V_3$. Используя выражение (11.15) в вышеприведенном уравнении и преобразовывая его, получим

$$(B_{22}Z_2B_{22}^t)\,I_l=-B_{23}V_3-B_{22}Z_2B_{12}^tI_1. \tag{11.18}$$

Полученное соотношение является *контурной системой* уравнений, которая включает только μ—n_c-переменные, где n_c — число источников тока в цепи N. Отметим, что μ—n_c равно цикломатическому числу цепи, полученной после удаления из N всех источников тока.

Матрица $Z_l=B_{22}Z_2B_{22}^t$ в уравнении (11.18) называется *матрицей импедансов* контура цепи N. Если цепь N не имеет взаимных индуктивностей, то Z_2 будет диагональной матрицей без нулевых элементов на диагонали. Следовательно, в этом случае Z_l будет невырожденной, поскольку B_{22} имеет максимальный ранг, равный $\mu-n_c$. Если цепь N имеет взаимные индуктивности, то Z_l будет невырожденной, если только Z_2 является положительно определенной.

Найдя I_l, используя выражение (11.18), можно найти I_2, используя выражение (11.15), и V_2, используя выражение (11.17). Затем можно определить V_1 и I_3, используя выражения (11.13) и (11.16). (Отметим, что I_1 и V_3 имеют определенные значения.) Этим можно было бы закончить анализ цепи N с использованием контурной системы уравнений. Заметим, что контурная система выводится сначала подстановкой отношений v—i для элементов в уравнение ЗКН и затем использованием контурного преобразования. В результате можно вывести систему уравнений сечения в точном соответствии с принципом двойственности.

Сначала запишем уравнения ЗКТ в распределенной форме

$$\begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{21} & Q_{22} & U \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = [0],$$

где матрица коэффициентов является той же самой, что и фундаментальная матрица сечений цепи N по отношению к T. Из этих уравнений получим

$$Q_{12}I_2=-Q_{11}I_1, \tag{11.19}$$

$$Q_3=-Q_{21}I_1-Q_{22}I_2. \tag{11.20}$$

Вместо уравнений ЗКН можно использовать преобразование сечения

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} Q^t & Q_{21}^t \\ Q_{12}^t & Q_{22}^t \\ 0 & U \end{bmatrix} \begin{bmatrix} V_b \\ V_3 \end{bmatrix},$$

где V_b — вектор напряжений, связанный с ветвями остова T, не содержащими источников. Из этих уравнений получаем

$$V_1=Q_{11}^tV_b+Q_{21}^tV_3, \tag{11.21}$$

$$V_2=Q_{12}^tV_b+Q_{22}^tV_3. \tag{11.22}$$

Если Y_2 — матрица проводимости RLC-элементов в цепи N, то отношение v—i на элементах можно записать в виде

$$I_2=Y_2V_2. \tag{11.23}$$

Теперь совсем нетрудно вывести систему уравнений сечения, которая устанавливает взаимосвязь между V_b, I_1 и V_3. Используя выражение (11.23) в (11.19), получим $Q_{12}Y_2V_2=-Q_{11}I_1$. Подставляя фор-

мулу (11.22) в полученное выше выражение, имеем

$$(Q_{12}Y_2Q_{12}^t)V_b = -Q_{11}I_1 - Q_{12}Y_2Q_{22}^tV_3. \qquad (11.24)$$

Полученное соотношение является *системой уравнений сечения*, которая включает $\rho - n_v$-переменные, где n_v — число источников напряжения в цепи N. Отметим, что $\rho - n_v$ равно рангу цепи, полученной из N стягиванием всех источников напряжения. Матрица $Y_b = Q_{12}Y_2Q_{12}^t$ в выражении (11.24) называется *матрицей проводимости сечения*.

Эта матрица будет невырожденной, если N не имеет взаимных индуктивностей. Если N имеет взаимные индуктивности, то Y_b будет невырожденной только в том случае, если Y_2 положительно определена.

Если определено V_b с использованием выражения (11.24), то можно определить V_2, используя выражение (11.22), и затем определить I_2, используя выражение (11.23). Наконец, можно определить I_3 и V_2 из выражений (11.20) и (11.21) соответственно. Теперь проиллюстрируем определение контурной системы уравнений и системы уравнений сечения.

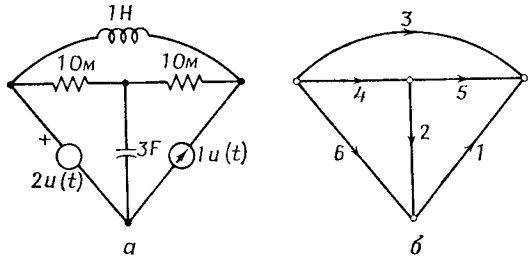


Рис. 11.3. Электрическая цепь и ее граф.

Рассмотрим цепь, представленную на рис. 11.3, *а*, где $u(t)$ — единичная ступенчатая функция. Граф этой цепи показан на рис. 11.3, *б*. Выбираем остов T, состоящий из ребер 4, 5 и 6. Заметим, что он содержит источник напряжения и не содержит источников тока. Фундаментальные цикломатическая матрица и матрица сечений по отношению к остову T даны ниже в требуемой распределенной форме:

$$B_f = \begin{array}{c} \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left[\begin{array}{c|cccc|c} 1 & 0 & 0 & -1 & -1 & 1 \\ \hline 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 & -1 & 0 \end{array}\right] \end{array},$$

$$Q_f = \begin{array}{c} \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left[\begin{array}{c|cccc|c} 1 & -1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ \hline -1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}.$$

Из этих матриц получаем

$$B_{12}=[0 \quad 0 \quad -1 \quad -1],$$
$$B_{13}=[1],$$
$$B_{22}=\begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 1 & -1 & -1\end{bmatrix},$$
$$B_{23}=\begin{bmatrix}-1\\ 0\end{bmatrix},$$
$$Q_{11}=\begin{bmatrix}1\\ 1\end{bmatrix},$$
$$Q_{12}=\begin{bmatrix}-1 & 1 & 1 & 0\\ 0 & 1 & 0 & 1\end{bmatrix},$$
$$Q_{21}=[-1],$$
$$Q_{22}=[1 \quad 0 \quad 0 \quad 0].$$

Также имеем

$$Z_2=\begin{bmatrix}1/3s & 0 & 0 & 0\\ 0 & s & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},$$
$$Y_2=\begin{bmatrix}3s & 0 & 0 & 0\\ 0 & 1/s & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix},$$

а также $v_6(s)=2/s$ и $i_1(s)=1/s$.

Используя эти соотношения в выражениях (11.18) и (11.24), получим контурную систему уравнений и систему уравнений сечения, как показано ниже:

Система контуров

$$\begin{bmatrix}1+\frac{1}{3s} & -1\\ -1 & s+2\end{bmatrix}\begin{bmatrix}i_2(s)\\ i_3(s)\end{bmatrix}=\begin{bmatrix}\frac{3}{s}\\ -\frac{2}{s}\end{bmatrix}.$$

Система разрезов

$$\begin{bmatrix}3s+1+\frac{1}{s} & \frac{1}{s}\\ \frac{1}{s} & 1+\frac{1}{s}\end{bmatrix}\begin{bmatrix}v_4(s)\\ v_5(s)\end{bmatrix}=\begin{bmatrix}6-\frac{1}{s}\\ -\frac{1}{s}\end{bmatrix}.$$

Предположим, что цепь N не имеет независимых источников напряжения. Тогда удобное описание N с помощью узловых напряжений как независимых переменных можно получить следующим образом:

Пусть A — матрица инциденций цепи N, усеченная по вершине v_r. Рассмотрим разбиение матрицы A в виде $A=[A_{11}A_{12}]$, где столбцы A_{11} и A_{12} соответствуют RLC-элементам и источникам тока. Если I_1 и I_2—вектор-столбцы токов в RLC-элементах и токи от источников тока, то уравнение ЗТК для N можно записать в виде $A_{11}I_1=-A_{12}I_2$. Имеем также $I_1=Y_1V_1$, где V_1 — вектор-столбец напряжений на RLC-элементах, а Y_1 — соответствующая матрица проводимостей. Далее, узловым преобразованием выражения (11.12) получаем

$$V_1=A_{11}^tV_n,$$

где V_n — вектор-столбец напряжений в узлах. Таким образом, из уравнений ЗКТ получаем $(A_{11}Y_1A_{11}^t)V_n=-A_{12}I_2$.

Полученные уравнения называются *узловыми уравнениями*. Матрица $A_{11}Y_1A_{11}^t$ называется *матрицей узловых проводимостей* N.

11.3. Метод смешанных переменных

В этом разделе обсуждается анализ цепей с помощью метода смешанных переменных. В этом методе, который является по сути комбинацией как метода контуров, так и метода сечений, некоторыми независимыми переменными являются напряжения, а другими независимыми переменными — токи. Ограничим наше рассмотрение RLC-цепями (без взаимных индуктивностей), содержащими независимые источники тока и напряжения. Его можно распространить непосредственно на цепи с взаимными индуктивностями.

Рассмотрим связную цепь N. Допустим, что элементы N разбиты на два таких подмножества E_1 и E_2, что E_1 содержит все источники напряжения, а E_2 — все источники тока. Пусть N_1' — цепь, полученная удалением E_2 из цепи N, а N_2^* получена стягиванием всех элементов E_1. Пусть T_1 — остовный лес цепи N_1', а T — остов N_2^*. Тогда $T=T_1\cup T_2$ является остовом N. Мы выбираем такие T_1 и T_2, что T содержит все источники напряжения, но не содержит источников тока.

Определим

T_v — подграф T_1, содержащий все элементы из T_1, исключая источники напряжения;

$\hat{T}_1$ — дополнение T_1 в N_1';

$\hat{T}_2$ — дополнение T_2 в N_2^*;

T_i — подграф $\hat{T}_2$, содержащий все элементы из $\hat{T}_2$, кроме источников тока.

Разобьем вектор напряжений на элементах V_e и вектор токов в элементах I_e следующим образом:

$$V_e=\begin{bmatrix} V_E \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_J \end{bmatrix}, \quad I_e=\begin{bmatrix} I_E \\ I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_J \end{bmatrix},$$

где индексы E, J, 1, 2, 3 и 4 относятся к источникам напряжения тока, а также к элементам T_v, T_2, $\hat{T}_1$ и T_i соответственно. Теперь мы ищем описание цепи N через переменные V_1 и I_4. Используя матрицы B_f и Q_f по отношению к T, можно записать уравнения ЗКН и ЗКТ следующим образом:

$$\begin{array}{c} \\ \\ 3 \\ 4 \\ J \end{array}\begin{array}{c} \begin{array}{cccccc} (T_v) & (T_2) & (\hat{T}_1) & & (T_i) & \\ E & 1 & 2 & 3 & 4 & J \end{array} \\ \begin{bmatrix} A & B & 0 & U & 0 & 0 \\ C & D & F & 0 & U & 0 \\ G & H & K & 0 & 0 & U \end{bmatrix} \end{array}\begin{bmatrix} V_E \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_J \end{bmatrix}=[0]. \tag{11.25}$$

Примечание. Объясните присутствие нулевой подматрицы в столбце, соответствующем T_2.

$$\begin{array}{c} \\ E \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cccccc} E & 1 & 2 & 3 & 4 & J \end{array} \\ \begin{bmatrix} U & 0 & 0 & -A^t & -C^t & -G^t \\ 0 & U & 0 & -B^t & -D^t & -H^t \\ 0 & 0 & U & 0 & -F^t & -K^t \end{bmatrix} \end{array}\begin{bmatrix} I_E \\ I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_J \end{bmatrix}=[0]. \tag{11.26}$$

Рассмотрим вторые системы уравнений в выражениях (11.25) и (11.26)

$$FV_2+V_4=-CV_E-DV_1, \tag{11.27}$$

$$I_1-B^tI_3=D^tI_4+H^tI_J. \tag{11.28}$$

Используя v—i-соотношения, получим $V_2=Z_2I_2$, $V_4=Z_4I_4$, $I_1=Y_1V_1$, $I_3=Y_3V_3$. Теперь можно записать выражения (11.27) и (11.28) в виде

$$FZ_2I_2+Z_4I_4=-CV_E-DV_1, \tag{11.29}$$

$$Y_1V_1-B^tY_3V_3=D^tI_4+H^tI_J. \tag{11.30}$$

Из контурного преобразования и преобразования сечения имеем

$$I_2 = F^t I_4 + K^t I_J, \tag{11.31}$$

$$V_3 = -AV_E - BV_1 \tag{11.32}$$

Теперь, подставляя I_2 и V_3 в выражения (11.29) и (11.30) и группируя члены, получим

$$\begin{bmatrix} Z_4 + FZ_2F^t & D \\ -D^t & Y_1 + B^tY_3B \end{bmatrix} \begin{bmatrix} I_4 \\ V_1 \end{bmatrix} = \begin{bmatrix} -C & -FZ_2K^t \\ -B^tY_3A & H^t \end{bmatrix} \begin{bmatrix} V_E \\ I_J \end{bmatrix}.$$

Полученное соотношение называется *гибридной системой* или *системой уравнений со смешанными переменными.*

Нетрудно убедиться, что если I_4 и V_1 определены с использованием уравнений, приведенных выше, то все другие переменные можно легко вычислить, используя выражения (11.25) и (11.26), а также контурное преобразование и преобразование сечения.

Заметим, что метод смешанных переменных сводится к контурному методу, если выбрать $E_1 = \varnothing$ и $E_2 = E$, и сводится к методу сечения, если выбрать $E_1 = E_2$ и $E_2 = \varnothing$, где E — множество элементов цепи N. Далее система уравнений со смешанными переменными включает $\rho(N_1') + \mu(N_2^*) - n_v - n_c$ переменных. Очевидно, что число $\rho(N_1') + \mu(N_2^*)$ зависит от выбора E_1 и E_2. Отсюда возникает проблема такого разбиения (E_1, E_2) системы элементов E цепи N, чтобы сумма $\rho(N_1') + \mu(N_2^*)$ была по возможности минимальной. Метод получения такого разбиения обсуждается в следующем разделе.

11.4. Главное разбиение графа

В этом разделе описывается главное разбиение графа, введенное в работе [11.7]. Как мы увидим, главное разбиение графа G определяет разбиение (E_1, E_2) системы ребер E графа G, использование которого в методе смешанных переменных приводит к минимальному числу независимых переменных. Наши рассуждения основаны на работах [11.7, 11.8].

Рассмотрим связный граф G. Все его подграфы, которые будут рассматриваться в этом разделе, являются реберно-порожденными подграфами. Поэтому подграф и его система ребер будут обозначаться одним и тем же символом.

Расстояние $d(T_1, T_2)$ между двумя остовами T_1 и T_2 графа G определяется как $d(T_1, T_2) = |T_1 - T_2| = |T_2 - T_1|$.

Таким образом, $d(T_1, T_2)$ равно числу ребер, которые имеются в $T_1(T_2)$ и которых нет в $T_2(T_1)$. Легко показать, что $d(T_1, T_2) = \rho(G)$— Число общих ветвей T_1 и T_2; $d(T_1, T_2) = \mu(G)$— Число общих хорд T_1 и T_2. Говорят, что остовы T_1 и T_2 являются *максимально удаленными*, если $d(T_1, T_2) \geqslant d(T_i, T_j)$ для любой пары остовов T_i и

T_j графа G. Например, $T_1=\{2, 3, 4, 7\}$ и $T_2=\{1, 3, 5, 6\}$ образуют пару максимально удаленных остовов графа, представленного на рис. 11.4.

Теорема 11.4. Пусть T_1 и T_2 образуют пару максимально удаленных остовов связного графа G.

1. Фундаментальный цикл графа G по отношению к T_1 или T_2, определенный общей хордой T_1 и T_2, не содержит общих ветвей этих остовов.

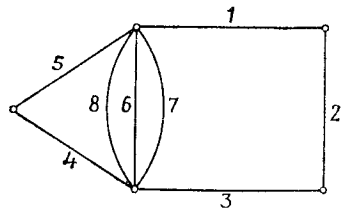


Рис. 11.4.

2. Фундаментальное сечение графа G по отношению к T_1 и T_2, определенное общей ветвью T_1 и T_2, не содержит общих хорд этих остовов.

Доказательство. 1. Предположим, что фундаментальный цикл графа G по отношению к T_1, определенный общей хордой c, содержит общую ветвь b. Тогда расстояние $d(T_1', T_2)$ между T_2 и остовом T_1' равно $(T_1—b) \cup c$ удовлетворяет соотношению $d(T_1', T_2)=d(T_1, T_2)+1$, противоречащему утверждению о максимальной удаленности T_1 и T_2.

Доказательство п. 2 следует из принципа двойственности.

Пусть c — общая хорда, а b — общая ветвь для любых двух остовов T_1 и T_2. Тогда последовательность P: $cT_1e_1T_2e_2T_1\ldots e_iT^*\ b$, где T^* означает либо T_1, либо T_2, называется *выводимой последовательностью* длиной i от общей хорды c до общей ветви b, если P имеет следующие свойства:

1. T_1 и T_2 входят в последовательность P попеременно.
2. e_1 находится в фундаментальном цикле по отношению к T_1, определяемом хордой c.
3. b находится в фундаментальном сечении по отношению к T^*, определяемом e_i.
4. Если e_j и e_{j+1} входят в P в виде $e_jT_ae_{j+1}$, где $T_a=T_1$ или T_2, то e_j — хорда T_a, а e_{j+1} — ветвь T_a и e_{j+1} входит в фундаментальный цикл по отношению к T_a, определенный e_j. Заменяя в этом определении цикл на разрез, а ветвь на хорду, можно, исходя из принципа двойственности, определить выводимую последовательность от общей ветви к общей хорде. Фактически если P является выводимой последовательностью от общей хорды c к общей ветви b, то последовательность P', записанная как и P, но в обратном порядке, будет выводимой последовательностью от b к c. В качестве примера рассмотрим остовы $T_1=\{1, 2, 3, 7, 10\}$ и $T_2=\{2, 6, 8, 11, 12\}$ графа, представлен-

ного на рис. 11.5. Ребро 5 является общей хордой T_1 и T_2, а ребро 2 — общей ветвью T_1 и T_2. Тогда P : $5T_17T_28T_12$ является выводимой последовательностью от 5 к 2, а P' : $2T_18T_27T_15$ — выводимой последовательностью от 2 к 5.

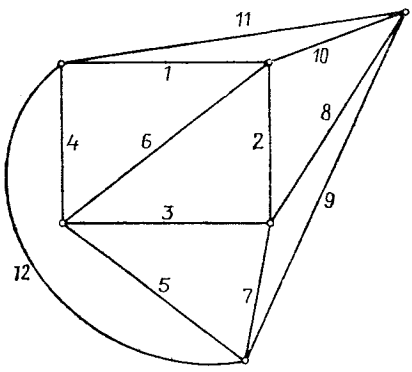


Рис. 11.5.

Пусть T_1 и T_2 образуют пару максимально удаленных остовов. Предположим, что существует выводимая последовательность P: $cT_1eT_2\ b$ длиной 1 от общей хорды c к общей ветви b. Тогда T_1 и T_2', равные $(T_2—b)\cup e$, образуют пару максимально удаленных остовов, для которых c — общая хорда, а b — общая ветвь. Далее, общая ветвь e_1 будет входить в фундаментальный цикл по отношению к T_1, определенный общей хордой c. Но это противоречит теореме 11.4. Таким образом, не существует выводимой последовательности длиной 1 от c к b. Из принципа двойственности можно показать, что не существует выводимой последовательности длиной 1 от b к c.

Теорема 11.5. Пусть c — общая хорда, а b — общая ветвь пары остовов T_1 и T_2 связного графа. Если T_1 и T_2 являются максимально удаленными, то не существует выводимых последовательностей от c к b и от b к c.

Доказательство. Доказательство проводится по индукции для длины выводимой последовательности по отношению к любой паре максимально удаленных остовов.

Как было показано выше, не существует выводимой последовательности длиной 1 от общей хорды к общей ветви для любой пары максимально удаленных остовов.

Допустим, что не существует такой последовательности длиной меньше чем $k\geqslant 2$. Предположим, что существует выводимая последовательность P: $cT_1e_1T_2e_2T_1\ldots e_kT^*b$ длины k от общей хорды c к общей ветви b для некоторой пары максимально удаленных остовов T_1 и T_2. Тогда она была бы кратчайшей такой последовательностью, так как (исходя из предположения) не существует такой последовательности длины, меньшей k.

Если $T^*=T_1$, то T_2 и $T_1'=(T_1—b)\cup e_k$ образуют пару максимально удаленных остовов, для которых c — общая хорда, а e_k — общая ветвь. Так как P — кратчайшая последовательность длины k, можно показать, что P' : $cT_1'e_1T_2e_2T_1'\ldots$ ... $e_{k-1}T_2e_k$ является выводимой последовательностью от c к e_k. Однако P' имеет длину $k—1$, и это ведет к противоречию, так как мы предполагаем, что не существует такой последовательности длины $k—1$.

Аналогично если $T^*=T_2$, то T_1 и $T_2'=(T_2—b)\cup e_k$ образуют пару максимально удаленных остовов, для которых c — общая хорда, а e_k — общая ветвь. Снова можно показать, что P'' : $cT_1e_1T_2e_2T_1\ldots e_{k-1}T_1e_k$ — выводимая последовательность от c к e_k. Так как P'' имеет длину $k—1$, приходим к противоречию.

Таким образом, не существует выводимой последовательности любой длины от общей хорды к общей ветви остовов T_1 и T_2. Аналогично не существует выводимой последовательности любой длины от общей ветви к общей хорде.

Пусть c — общая хорда T_1 и T_2 для заданной пары максимально удаленных деревьев T_1 и T_2.

K — подграф G_c графа G по отношению к c — строится следующим образом:

1. Пусть L_1 — множество ребер в фундаментальном цикле относительно T_1, определенного c. По теореме 11.4 множество L_1 не содержит *общих* ветвей.
2. Пусть L_2 — объединение всех фундаментальных циклов по отношению к T_2, определенных каждым ребром в L_1. По теореме 11.5 объединение L_2 не содержит общих ветвей.
3. Используя многократно приведенную выше схему, можно получать последовательность множеств ребер L_1, L_2, ... до тех пор, пока не достигнем множества $L_{k+1}=L_k$.

Тогда подграф, порожденный на множестве ребер L_k, называется *K-подграфом* G_c по отношению к c.

Заменяя в приведенной выше конструкции цикл сечением, а хорду — ветвью, можно из принципа двойственности определить *K-подграф* G_b по отношению к общей ветви b.

Главным подграфом G_1 по отношению к общим хордам является объединение K-подграфов относительно всех общих хорд. Главным подграфом G_2 по отношению к общим ветвям является объединение K-подграфов по отношению ко всем общим ветвям. Например, главными подграфами G_1 и G_2 графа, представленного на рис. 11.4 по отношению к паре остовов $T_1=\{2, 3, 4, 7\}$ и $T_2=\{1, 3, 5, 6\}$, являются $G_1=\{6, 7, 8\}$, $G_2=\{1, 2, 3\}$.

В работе [11.7] показано, что главные подграфы G_1 и G_2 не имеют общих ребер. Если бы они имели общее ребро, то можно было бы построить выводимую последовательность от общей хорды к общей ветви.

Таким образом, любой граф G состоит из трех подграфов: G_1 — главного подграфа по отношению к общим хордам, G_2 — главного подграфа по отношению к общим ветвям, G_0 — подграфа $G-(G_1 \cup G_2)$. Это разбиение (G_0, G_1, G_2) графа G называется *главным разбиением* графа G. Интересно отметить [11.7], что главное разбиение графа G единственно и не зависит от выбора максимально удаленных остовов.

Некоторые полезные свойства главных подграфов G_0, G_1, G_2 перечислены ниже. Они следуют из определений этих подграфов.

P1. G_1 содержит все общие хорды, но не содержит общих ветвей, G_2 содержит все общие ветви, но не содержит общих хорд.

P2. Любой фундаментальный цикл по отношению к T_1 или T_2, определенный ребром в G_1, состоит только из ребер G_1.

P3. Любое фундаментальное сечение по отношению к T_1 и T_2, определенное ребром в G_2, состоит только из ребер G_2.

P4. $T_1 \cap G_1$ и $T_2 \cap G_1$ являются остовными лесами G_1.

P5. $T_1 \cap G_2$ и $T_2 \cap G_2$ являются остовами графа G_2', полученного стягиванием всех ребер, не входящих в G_2.
P6. Любое ребро, концевые вершины которого являются компонентами G_1, также принадлежит G_1.
P7. Любое ребро, концевые вершины которого являются компонентами $G_0 \cup G_1$, также принадлежит $G_0 \cup G_1$.
P8. $T_1 \cap G_0$ и $T_2 \cap G_0$ являются остовными лесами графа G_0', полученного включением всех ребер G_1 и удалением всех ребер G_2 (это свойство можно доказать, используя свойство P3).

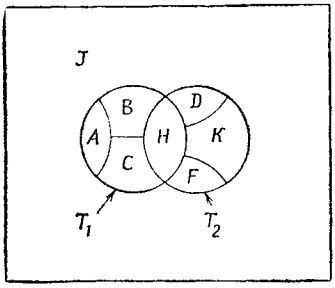


Рис. 11.6.

Заметим, что $G_1=\varnothing$, если G не содержит общих хорд, и $G_2=\varnothing$, если G не содержит общих ветвей.

Разбиение множества ребер E графа G, вытекающее из свойства P1, представлено на рис. 11.6. Оно следует из того, что $E_1=A \cup D \cup J$, $E_2=B \cup K \cup H$, $E_0=C \cup F$, где E_1, E_2 и E_0 означают множества ребер G_1, G_2, G_0 соответственно. Отметим, что, как следствие свойств P4 и P5, получаем $|A|=|D|$ и $|B|=|K|$. Результаты, представленные в предшествующих обсуждениях, приводят к следующей теореме:

Теорема 11.6. Для графа G, имеющего (G_0, G_1, G_2) в качестве главного разбиения, получаем
1) если $G_1 \neq \varnothing$, то $\rho(G_1)<\mu(G_1)$;
2) если $G_2 \neq \varnothing$, то $\rho(G_2')>\mu(G_2')$, где G_2' — граф, полученный стягиванием всех ребер в $(G_0 \cup G_1)$;
3) $\rho(G_0')=\mu(G_0')$, где G_0' получен стягиванием всех ребер в G_1 и удалением всех ребер в G_2;
4) максимальное расстояние между двумя любыми остовами равно $\rho(G_1)+\mu(G_2')+\rho(G_0')$.

Доказательство. 1. Из свойства P4 и из того, что $|A|=|D|$, получаем $\rho(G_1)=|A|<|D|+|J|$, если $G_1 \neq \varnothing=\mu(G_1)$.

2. Из свойства P5 и из того, что $|B|=|K|$, получаем $\rho(G_2')=|B|+|H|>K$, если $G_2 \neq \varnothing=\mu(G_2')$.

3. Из свойства P8 и из того, что $|C|=|F|$, получаем $\rho(G_0')=C=F=\mu(G_0')$.

4. Максимальное расстояние между любыми двумя остовами G равно $|A|+|B|+|C|=\rho(G_1)+\mu(G_2')+\rho(G_0')$.

Пусть d_m — максимальное расстояние между двумя остовами графа G, пусть (G_0, G_1, G_2) — главное разбиение G. Тогда из доказанной выше теоремы имеем $d_m=\rho(G_1)+\mu(G_2')+\rho(G_0')$. Но $\mu(G_2')+\rho(G_0')$ равно цикломатическому числу графа $(G_0 \cup G_2')$, полученного стягиванием всех ребер в G (рис. 11.6). Следовательно, $d_m=\rho(G_1)+\mu((G_0 \cup G_2)')$. Пусть E_a и E_b образуют произвольное разбиение множества ребер E графа G. Пусть G_a — подграф на множестве ребер

E_a, а G_b' — граф, полученный стягиванием всех ребер в G_a. В работе [11.8] показано, что $d_m \leqslant \rho(G_a)+\mu(G_b')$.

Заметим, что $\rho(G_a)+\mu(G_b')$ — число независимых переменных в анализе со смешанными переменными, если использовать разбиение (E_a, E_b) множества E. Следовательно, можно заключить, что d_m — минимальное число независимых переменных, требуемых в анализе со смешанными переменными. Это число может быть меньше ранга и цикломатического числа G. Таким образом, число независимых переменных в методе смешанных переменных может быть меньше требуемого для методов контуров и сечений. Например, граф G на рис. 11.4 имеет главное разбиение $G_0=\{4, 5\}$, $G_1=\{6, 7, 8\}$, $G=\{1, 2, 3\}$. Можно проверить, что $\rho=(G_1)=1$, $\mu(G_0 \cup G_2)'=2$, следовательно, $d_m=3$. Таким образом, только три независимые переменные требуются для анализа цепи по методу смешанных переменных, тогда как оба метода контуров и сечений требуют по четыре независимых переменных.

В работе [11.8] d_m называется *топологической степенью свободы* цепи. Некоторые интересные свойства таких разбиений (E_a, E_b), для которых $\rho(G_a)+\mu(G_b')=d_m$, обсуждаются в работе [11.8].

Авторы работ [11.7, 11.8] внесли несколько глубоких результатов в теорию графов. В работе [11.9] обсуждается алгоритм вычисления главного сечения графа. В работе [11.10] распространяется понятие «главное разбиение на матроиды».

11.5. Уравнения состояния

В разд. 11.2—11.4 для описания цепи рассматривались системы контурных уравнений, уравнений сечения и уравнений со смешанными переменными. Во временно́й области эти уравнения являются интегродифференциальными. В данном разделе мы описываем цепи в виде дифференциальных уравнений первого порядка без интегралов. Одной из причин такого описания является то, что в математической литературе имеется масса информации по решению подобных уравнений и свойствам таких решений, которые можно легко использовать в рассматриваемом случае. Далее, представление состояния является более общим в том случае, когда оно применимо к меняющимся во времени и нелинейным цепям. Вновь ограничиваем наше внимание специальным классом *RLC*-цепей с взаимными индуктивностями и независимыми источниками тока и напряжения.

Уравнения состояния электрической цепи N формулируются с использованием в качестве переменных производных емкостных напряжений и индуктивных токов. Очевидно, что не все емкостные напряжения можно выбрать в качестве независимых переменных, потому что в цепи могут существовать контуры, состоящие только из емкостей. Аналогично не все индуктивные токи можно выбрать в качестве независимых переменных, потому что могут существовать

сечения, состоящие только из индуктивностей. Как и в случае методов контура и сечения, исходным пунктом для вывода уравнений состояния является выбор соответствующего остова. В частности, выбираем такой остов, который содержит 1) все источники напряжения и ни одного источника тока, 2) наибольшее возможное число емкостей и 3) наименьшее возможное число индуктивностей. Остов, выбранный в соответствии с этими правилами, называется *нормальным деревом*.

Для заданной цепи N пусть N_1 — подграф цепи N, который содержит все источники напряжения и емкости, а N_2 — подграф цепи N, который содержит все источники напряжения, емкости и сопротивления. Нормальное дерево цепи N можно выбрать следующим образом:

1. Сначала выбрать такой остовный лес T_1 из N_1, чтобы он содержал все источники напряжения.
2. Затем выбрать такой остовный лес T_2 из N_2, что $T_1 \subseteq T_2$.
3. К T_2 добавить столько индуктивностей, сколько необходимо, чтобы получить остов T из N. Полученное таким образом дерево является нормальным деревом цепи N.

В качестве примера нормальное дерево цепи, изображенной на рис. 11.7, *а*, представлено на рис. 11.7, *б*.

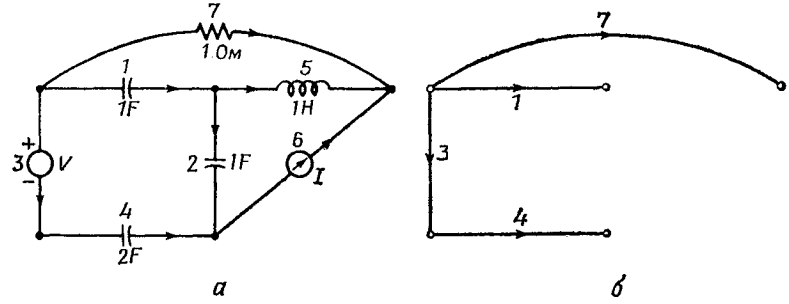


Рис. 11.7.
а — цепь; *б* — нормальное дерево цепи.

После выбора нормального дерева T дадим разбиение вектора V_e напряжений элемента и вектора I_e токов элемента следующим образом:

$$V_e = \begin{bmatrix} V_t \\ V_l \end{bmatrix}, \quad I_e = \begin{bmatrix} I_t \\ I_l \end{bmatrix},$$

$$V_t = \begin{bmatrix} V_E \\ V_{Ct} \\ V_{Rt} \\ V_{Lt} \end{bmatrix}, \quad V_l = \begin{bmatrix} V_{Cl} \\ V_{Rl} \\ V_{Ll} \\ V_J \end{bmatrix},$$

$$I_t=\begin{bmatrix} I_E \\ I_{Ct} \\ I_{Rt} \\ I_{Lt} \end{bmatrix}, \quad I_l=\begin{bmatrix} I_{Cl} \\ I_{Rl} \\ I_{Ll} \\ I_J \end{bmatrix},$$

где индексы E, J, C, R и L относятся к источникам напряжения, источникам тока, емкостям, резисторам и индуктивностям соответственно, а индексы t и l — к ветвям и хордам T соответственно.

Затем подматрицу Q_{fl} фундаментальной матрицы сечений $Q_f=[U\ Q_{fl}]$ можно разбить так:

$$Q_{fl}=\begin{matrix} \\ \text{Ветви} \downarrow \end{matrix}\begin{matrix} \\ E \\ C \\ R \\ L \end{matrix}\begin{matrix} \text{Связи} \rightarrow \\ C & R & L & J \end{matrix}\begin{bmatrix} Q_{EC} & Q_{ER} & Q_{EL} & Q_{EJ} \\ Q_{CC} & Q_{CR} & Q_{CL} & Q_{CJ} \\ 0 & Q_{RR} & Q_{RL} & Q_{RJ} \\ 0 & 0 & Q_{LL} & Q_{LJ} \end{bmatrix}. \quad (11.33)$$

Заметим, что $Q_{RC}=0$. Если бы существовала емкость в фундаментальном сечении по отношению к некоторому сопротивлению, то удаление этого сопротивления из T и добавление этой емкости привело бы к тому, что в остове было бы на одну емкость больше, чем в T. Это противоречило бы выбору нормального дерева T. Аналогично можно показать, что $Q_{LC}=0$ и $Q_{LR}=0$. Из уравнений ЗКТ $Q_f I_E=0$, и отсюда получаем следующее:

$$I_E=-Q_{EC}I_{Cl}-Q_{ER}I_{Rl}-Q_{EL}I_{Ll}-Q_{EJ}I_J, \quad (11.34a)$$
$$I_{Ct}=-Q_{CC}I_{Cl}-Q_{CR}I_{Rl}-Q_{CL}I_{Ll}-Q_{CJ}I_J, \quad (11.34b)$$
$$I_{Rt}=-Q_{RR}I_{Rl}-Q_{RL}I_{Ll}-Q_{RJ}I_J, \quad (11.34c)$$
$$I_{Lt}=-Q_{LL}I_{Ll}-Q_{LJ}I_J. \quad (11.34d)$$

Далее из преобразования сечений получаем

$$V_{Cl}=Q'_{EC}V_E+Q'_{CC}V_{Ct}, \quad (11.35a)$$
$$V_{Rl}=Q'_{ER}V_E+Q'_{CR}V_{Ct}+Q'_{RR}V_{Rt}, \quad (11.35b)$$
$$V_{Ll}=Q'_{EL}V_E+Q'_{CL}V_{Ct}+Q'_{RL}V_{Rt}+Q'_{LL}V_{Lt}, \quad (11.35c)$$
$$V_J=Q'_{EJ}V_E+Q'_{CJ}V_{Ct}+Q'_{RJ}V_{Rt}+Q'_{LJ}V_{Lt}. \quad (11.35d)$$

Используя элементные соотношения v—i и приведенные выше уравнения, мы должны получить уравнения состояния, исходя из производных V_{Ct} и I_{Lt}. Все другие переменные, не связанные с источниками, необходимо исключить.

Для емкостей нежелательными переменными являются I_{Ct}, V_{Cl} и I_{Cl}, а для индуктивностей — V_{Lt}, I_{Ll} и V_{Ll}. Исключим эти переменные. Сначала перепишем выражение (11.34, *б*) в виде

$$I_{Ct}+Q_{CC}I_{Cl}=[U \quad Q_{CC}]\begin{bmatrix}I_{Ct}\\ I_{Cl}\end{bmatrix}=-Q_{CR}I_{Rl}-Q_{CL}I_{Ll}-Q_{CJ}I_J.$$

Затем, используя соотношения v—i для емкостей в приведенных выше выражениях

$$\begin{bmatrix}I_{Ct}\\ I_{Cl}\end{bmatrix}=\begin{bmatrix}C_t & 0\\ 0 & C_l\end{bmatrix}\frac{d}{dt}\begin{bmatrix}V_{Ct}\\ V_{Cl}\end{bmatrix}$$

и подставляя V_{Cl} из уравнения (11.35а), получим

$$\mathscr{C}\frac{d}{dt}(V_{Ct})=-Q_{CR}I_{Rl}-Q_{CL}I_{Ll}-Q_{CJ}I_J+\mathscr{C}^*\frac{d}{dt}(V_E), \quad (11.36)$$

где

$$\mathscr{C}=[U \quad Q_{CC}]\begin{bmatrix}C_t & 0\\ 0 & C_l\end{bmatrix}\begin{bmatrix}U\\ Q^t_{EC}\end{bmatrix},$$

$$\mathscr{C}^*=-[U \quad Q_{CC}]\begin{bmatrix}C_t & 0\\ 0 & C_l\end{bmatrix}\begin{bmatrix}0\\ Q^t_{EC}\end{bmatrix}.$$

Можно показать, что $\mathscr{C}$ является матрицей проводимости сечений цепи, полученной из N удалением всех хордовых элементов, за исключением емкостей и стягиванием всех элементов дерева, кроме емкостей.

Аналогично, исходя из уравнения (11.35в) и используя v—i-соотношения

$$\begin{bmatrix}V_{Ll}\\ V_{Lt}\end{bmatrix}=\begin{bmatrix}L_{ll} & L_{lt}\\ L_{tl} & L_{tt}\end{bmatrix}\frac{d}{dt}\begin{bmatrix}I_{Ll}\\ I_{Lt}\end{bmatrix},$$

а затем заменяя I_{Lt} из уравнения (11.34г), можно получить следующее отношение для индуктивностей:

$$\mathscr{L}\frac{d}{dt}(I_{Ll})=Q^t_{CL}V_{Ct}+Q_{RL}V_{Rt}+Q^t_{EL}V_E+\mathscr{L}^*\frac{d}{dt}(I_J), \quad (11.37)$$

где

$$\mathscr{L}=[U \quad -Q^t_{LL}]\begin{bmatrix}L_{ll} & L_{lt}\\ L_{tl} & L_{tt}\end{bmatrix}\begin{bmatrix}U\\ -Q_{LL}\end{bmatrix},$$

$$\mathscr{L}^*=-[U \quad -Q_{LL}]\begin{bmatrix}L_{ll} & L_{lt}\\ L_{tl} & L_{tt}\end{bmatrix}\begin{bmatrix}0\\ -Q_{LJ}\end{bmatrix}.$$

Можно показать, что $\mathscr{L}$ является матрицей полных сопротивлений цепи, полученной удалением всех хордовых элементов, кроме ин-

дуктивностей, и стягиванием всех элементов дерева, кроме индуктивностей.

Теперь необходимо исключить I_{Rl} и V_{Rt} из уравнений (11.36) и (11.37). Для этого рассмотрим следующие v—i-соотношения: $I_{Rl}=G_l V_{Rl}$, $V_{Rt}=R_t I_{Rt}$. После подстановки V_{Rl} и I_{Rt} из уравнений (11.35б) и (11.34в) приведенные выше уравнения примут вид

$$I_{Rl}=G_l Q_{RR}^t V_{Rt}+G_l Q_{CR}^t V_{Ct}+G_l Q_{ER}^t V_E, \tag{11.38}$$

$$V_{Rt}=-R_t Q_{RR} I_{Rl}-R_t Q_{RL} I_{Ll}-R_t Q_{RJ} I_J. \tag{11.39}$$

Подставляя I_{Rl} из выражений (11.38), (11.39), приводим к виду

$$(U+R_t Q_{RR} G_l Q_{RR}^t) V_{Rt}=-R_t Q_{RR} G_l Q_{CR}^t V_{Ct}-R_t Q_{RR} G_l Q_{ER}^t V_E- \\ -R_t Q_{RL} I_{Ll}-R_t Q_{RJ} I_J. \tag{11.40}$$

Приведенное выше уравнение для V_{Rt} можно решить тогда и только тогда, когда существует обратная матрица для $(U+R_t Q_{RR} G_l Q_{RR}^t)$. Эту матрицу можно представить в виде $R_t G$, где $G=G_t+Q_{RR} G_l Q_{RR}^t$. Можно показать, что матрица $G_t+Q_{RR} G_l Q_{RR}^t$ является матрицей проводимости сечений цепи, полученной из N удалением всех хордовых элементов, кроме сопротивлений, и стягиванием всех элементов дерева, кроме сопротивлений. В таком случае матрица, обратная этой, существует, и уравнение (11.40) можно решить относительно V_{Rt}.

Далее подставим в уравнение (11.36) и (11.37) I_{Rl} и V_{Rt} в том виде, в каком они получены выше, и получим следующие уравнения состояния:

$$\frac{d}{dt}\begin{bmatrix}\mathscr{C} & 0\\ 0 & \mathscr{L}\end{bmatrix}\begin{bmatrix}V_{Ct}\\ I_{Ll}\end{bmatrix}=\begin{bmatrix}-\mathscr{Y} & \mathscr{H}\\ \mathscr{G} & -\mathscr{P}\end{bmatrix}\begin{bmatrix}V_{Ct}\\ I_{Ll}\end{bmatrix}+ \\ +\begin{bmatrix}-\mathscr{Y}^* & \mathscr{H}^*\\ \mathscr{S}^* & -\mathscr{G}^*\end{bmatrix}\begin{bmatrix}V_E\\ I_J\end{bmatrix}+\frac{d}{dt}\begin{bmatrix}\mathscr{C}^* & 0\\ 0 & \mathscr{L}^*\end{bmatrix}\begin{bmatrix}V_E\\ I_J\end{bmatrix}, \tag{11.41}$$

где

$$R=R_l+Q_{RR}^t R_t Q_{RR},$$

$$\mathscr{Y}=Q_{CR}R^{-1}Q_{CR}^t,\quad \mathscr{H}=-Q_{CL}+Q_{CR}R^{-1}Q_{RR}^t R_t Q_{RL},$$

$$\mathscr{P}=Q_{RL}^t G^{-1} Q_{RL},\quad \mathscr{G}=Q_{CL}^t-Q_{RL}^t G^{-1} Q_{RR} G_l Q_{CR}^t=-\mathscr{H}^t, \tag{11.42}$$

$$\mathscr{Y}^*=Q_{CR}R^{-1}Q_{ER},\quad \mathscr{H}^*=-Q_{CJ}+Q_{CR}R^{-1}Q_{RR}^t R_t Q_{RJ}, \\ \mathscr{P}^*=Q_{RL}^t G^{-1} Q_{RJ},\quad \mathscr{G}^*=Q_{EL}^t-Q_{RL}^t G^{-1} Q_{RR} G_l Q_{ER}^t. \tag{11.43}$$

Матрицы $\mathscr{C}$ и $\mathscr{L}$ являются невырожденными в случае RLC-цепи, и поэтому в таких случаях можно переписать уравнение (11.41) в виде

$$\frac{d}{dt}\begin{bmatrix}V_{Ct}\\ I_{Ll}\end{bmatrix}=\mathscr{A}\begin{bmatrix}V_{Ct}\\ I_{Ll}\end{bmatrix}+\mathscr{B}\begin{bmatrix}V_E\\ I_J\end{bmatrix}+\mathscr{D}\frac{d}{dt}\begin{bmatrix}V_E\\ I_J\end{bmatrix}. \tag{11.44}$$

Определив V_{Ct} и I_{Lt} из уравнения (11.44), можно получить все остальные напряжения и токи, исходя из V_{Ct}, I_{Jt}, V_E и I_J. Те, кого интересуют детали, могут обратиться к работам [11.11, 11.12]. Уравнение состояния (11.44) не приведено к нормальной форме, так как присутствуют производные от V_E и I_J. Определяя новое множество переменных, можно привести уравнение (11.44) к нормальному виду [11.11]. В качестве примера рассмотрим цепь, показанную на рис. 11.7, *а*. Нормальное дерево цепи представлено на рис. 11.7, *б*. Матрица Q_{fl} в форме разбиения дана ниже:

$$Q_{fl}=\begin{array}{c} \\ 3\\ 1\\ 4\\ 7 \end{array}\begin{array}{c} \begin{array}{ccc} 2 & 5 & 6 \end{array}\\ \left[\begin{array}{c|c|c} 1 & 0 & -1\\ \hline -1 & -1 & 0\\ 1 & 0 & -1\\ \hline 0 & 1 & 1 \end{array}\right] \end{array}.$$

Отсюда получаем

$$Q_{EC}=[1], \qquad Q_{RC}=[0],$$

$$Q_{EL}=[0], \qquad Q_{RL}=[1],$$

$$Q_{EJ}=[-1], \qquad Q_{RJ}=[1],$$

$$Q_{CC}=\begin{bmatrix} -1\\ 1 \end{bmatrix}, \qquad Q_{CL}=\begin{bmatrix} -1\\ 0 \end{bmatrix}, \qquad Q_{CJ}=\begin{bmatrix} 0\\ 1 \end{bmatrix}.$$

Используя приведенные выше матрицы, получаем

$$\mathcal{C}=\begin{bmatrix} 2 & -1\\ -1 & 3 \end{bmatrix}, \qquad \mathcal{C}^*=\begin{bmatrix} 1\\ -1 \end{bmatrix}, \qquad \mathcal{L}=[1], \qquad \mathcal{L}^*=[0],$$

$$G=[1], \qquad R=[0],$$

$$\mathcal{G}=\begin{bmatrix} 0 & 0\\ 0 & 0 \end{bmatrix}, \qquad \mathcal{G}^*=\begin{bmatrix} 0\\ 0 \end{bmatrix}, \qquad \mathcal{K}=\begin{bmatrix} 1\\ 0 \end{bmatrix}, \qquad \mathcal{K}^*=\begin{bmatrix} 0\\ 1 \end{bmatrix},$$

$$\mathcal{P}=[1], \qquad \mathcal{P}^*=[1], \qquad \mathcal{S}=[-1 \quad 0], \qquad \mathcal{S}^*=[0].$$

Уравнения состояния цепи, приведенной на рис. 11.7, получаются с использованием уравнения (11.41):

$$\frac{d}{dt}\left[\begin{array}{cc|c} 2 & -1 & 0\\ -1 & 3 & 0\\ \hline 0 & 0 & 1 \end{array}\right]\begin{bmatrix} v_1\\ v_4\\ \hline i_5 \end{bmatrix}=\left[\begin{array}{cc|c} 0 & 0 & 1\\ 0 & 0 & 0\\ \hline -1 & 0 & -1 \end{array}\right]\begin{bmatrix} v_1\\ v_4\\ \hline i_5 \end{bmatrix}+\left[\begin{array}{c|c} 0 & 0\\ 0 & 1\\ \hline 0 & -1 \end{array}\right]\begin{bmatrix} v_3\\ \hline i_6 \end{bmatrix}$$

$$+\left[\begin{array}{c|c} 1 & 0\\ -1 & 0\\ \hline 0 & 0 \end{array}\right]\begin{bmatrix} \dfrac{dv_3}{dt}\\ \dfrac{di_6}{dt} \end{bmatrix}.$$

11.6. Свойство неусиления в резистивных цепях

Эту главу мы завершаем интересным приложением теории графов к исследованию электрических цепей.

В теории цепей хорошо известно, что для заданной цепи, состоящей из резисторов и источников, величина напряжения на всех резисторах не выше суммы величин напряжений на источниках. Это свойство резистивных сетей известно как *свойство неусиления*. В этом разделе дается доказательство свойства неусиления. Это доказательство благодаря Волаверу [11.13] является чисто теоретико-графовым.

Напомним, что цикл, в котором все ребра ориентированы одинаково относительно ориентации этого цикла, называется *ориентированным циклом*. Сечение, в котором все ребра ориентированы одинаково по отношению к ориентации этого сечения, называется *ориентированным сечением*.

Доказательство неусиления, данное Волавером, основано на специальном случае леммы Минти о раскраске дуг (теорема 10.31), а именно на том, что в ориентированном графе каждое ребро находится либо в ориентированном цикле, либо в ориентированном сечении, либо отсутствует в обоих.

Теорема 11.7. Для заданной цепи источников и (линейных/нелинейных) положительных сопротивлений величина тока в любом сопротивлении с нулевым напряжением не больше суммы величин токов, текущих через источники.

Доказательство. Пусть все элементы с нулевым падением напряжения исключены (их можно рассматривать как короткие замыкания). Пусть направления рассмотрения элементов выбраны так, что все напряжения на элементах положительны. Тогда рассмотрим произвольное сопротивление с ненулевым напряжением. Не может существовать ориентированного цикла, который содержал бы такое сопротивление. Если бы такой направленный цикл существовал, то сумма всех напряжений по этому циклу цепи была бы ненулевой, что противоречит закону Кирхгофа для напряжений. Поэтому, как было показано раньше, существует ориентированное сечение, которое содержит рассматриваемое сопротивление. Пусть ток в рассматриваемом сопротивлении равен i_0. Выберем ориентированное сечение, которое содержит это сопротивление. Пусть R — множество всех других сопротивлений в этом сечении, а S — множество всех источников в сечении. Применяя закон Кирхгофа к сечению, получаем $i_0+\sum_{k\in R} i_k+\sum_{s\in S}\pm i_s=0$. Так как сопротивления положительны, то $v_i\geqslant 0$ для каждого сопротивления. Так как все напряжения положительны, ток в каждом из сопротивлений не отрицателен, и можно записать $|i_0|+\sum_{k\in R}|i_k|+\sum_{s\in S}\pm i_s=0$. Отсюда $|i_0|\leqslant\sum_{s\in S}\pm i_s\leqslant\sum_{s\in S}|i_s|$. Что и требовалось доказать.

Следующая теорема является двойственной к доказанной выше. Доказательство следует из принципа двойственности.

Теорема 11.8. Для заданной цепи, состоящей из источников и (линейных/нелинейных) положительных сопротивлений, величина напряжения на любом из сопротивлений не больше суммы величины напряжений на всех источниках.

Работы [11.14, 11.15] являются одними из первых, в которых обсуждается свойство неусиления.

11.7. Замечания, касающиеся литературы

Для дальнейшего изучения следует в первую очередь рекомендовать работу [11.16] по теоретико-графовому исследованию электрических цепей. Кроме того, рекомендуем работы [11.17 и 11.19]. В работе [11.11] можно найти более детальное обсуждение некоторых тем, представленных в данной главе, и в частности их распространение на более общие классы цепей.

Метод анализа цепей, названный *диакоптикой*, основанный на понятии разрыва, был введен в работах [11.20, 11.21]. Их автор применил это понятие для анализа определенного класса цепей. Его подход основывается на понятиях, заимствованных из тензорного анализа. В работах [11.22—11.25] разъясняются понятия диакоптики и делается очень многое для того, чтобы этот подход стал более известен инженерам-электротехникам. Другие описания диакоптики можно найти в работах [11.26—11.31].

В работе [11.32] рассматривается обобщенная форма анализа на основе смешанных переменных, которая включает все существующие формы (кроме диакоптики) в качестве специальных случаев. В работе [11.33] также исследуется вычислительная эффективность методов контуров и сечений. Авторы этих работ показали, что выбор оптимального метода анализа приведет к получению наиболее разреженных матриц сопротивлений контура и проводимости сечений. Авторы работы [11.34] использовали идеи теоретико-графового подхода, в особенности теорему Теллежена и специальный случай леммы о раскраске дуг для выявления некоторых свойств нелинейных цепей и нелинейных многополюсных резистивных цепей.

Уравнения состояния для *RLC*-схем были впервые выведены авторами работ [11.35, 11.36]. Ряд статей был посвящен обсуждению формирования уравнений состояния для обобщенных классов активных цепей [11.37—11.39]. В работе [11.16] устанавливаются необходимые и достаточные условия для разрешимости *RLC*-цепей, содержащих независимые источники. Используя этот подход, некоторые авторы обсуждали проблему разрешимости цепей для более общего класса цепей [11.37, 11.40].

Теоретико-матроидное изучение электрических цепей быстро развивается как перспективная область исследований. Определение электрической цепи основывается на графах. В качестве обобщения этого подхода в работах [11.41—11.43] вводится понятие «обобщенные цепи», т. е. цепи, основанные на матроидах, и устанавливаются некоторые свойства таких цепей. Ожидается, что более детальное изучение обобщенных цепей даст новый импульс некоторым классическим нерешенным задачам теории электрических цепей, в частности задаче синтеза многополюсной резистивной цепи. Авторы работы [11.44] недавно рассмотрели матрицы, использующие алгебру Ванга [11.45, 11.46], в качестве инструмента исследования. В работе

[11.47] исследуется задача разрешимости цепи с использованием понятия «объединение матроидов», а в работе [11.48] формулируется единый подход к трем фундаментальным проблемам в теории цепей и показывается, что все они сводятся к задаче определения базиса объединения двух соответствующим образом выбранных матроидов.

Теория электрических цепей продолжает оставаться богатым источником математических задач. В работе [11.49] обсуждается несколько таких задач.

Упражнения

11.1. Матрица путей $P=[p_{ij}]$ дерева T с соответствующей вершиной v_r определяется следующим образом: Если ветвь j является единственным путем в T от вершины v_i к v_j, то $p_{ij}=+1$ или $p_{ij}=-1$ в зависимости от того, соответствует или нет ориентация ветви направлению этого пути, в противном случае $p_{ij}=0$. Используя узловое преобразование, покажите, что $P=(A^{-1})^t$, где A является усеченной матрицей инциденций T по вершине v_r, и выведите теорему 6.12.

11.2. Получите контурную систему уравнений планарной цепи, используя ячейки как независимые циклы.

11.3. Пусть N — планарная цепь. Пусть планарная цепь $\hat{N}$ строится следующим образом:

а) граф $\hat{N}$ является двойственным графу N. Пусть e и e' являются соответствующими элементами из N и $\hat{N}$;

б) если e — резистор величиной R Ом, то e' — резистор величиной $1/R$ Ом;

в) если e — емкость (индуктивность) величиной K фарад (генри), то e' — индуктивность (емкость) величиной K генри (фарад);

г) если e — источник тока (напряжения) величиной $g(t)$, то e' — источник напряжения (тока) величиной $g(t)$.

Определите ориентации источников тока и напряжения в $\hat{N}$ так, чтобы контурные уравнения N (с ячейками, выбранными в качестве независимых циклов) стали узловыми уравнениями $\hat{N}$, где переменные контурных токов заменяются переменными узловых напряжений. Можно принять, что все ячейки из N ориентированы по часовой стрелке. (Цепи N и $\hat{N}$, определенные, как описано выше, называются *двойственными* цепями.)

11.4. Постройте двойственную цепь к цепи, показанной на рис. 11.8.

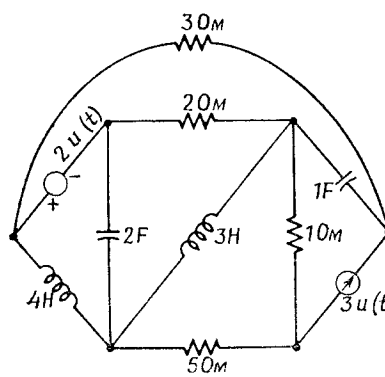


Рис. 11.8.

11.5. Определите все напряжения и токи в цепи, представленной на рис. 11.9, используя контурный метод анализа.
11.6. Повторите упражнение 11.5, используя методы сечения и смешанных переменных.

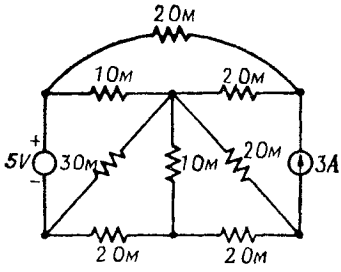


Рис. 11.9.

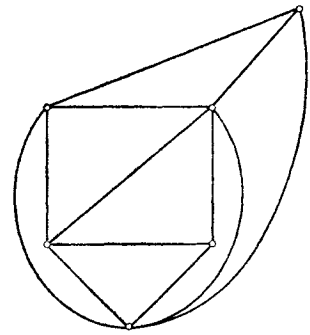


Рис. 11.10.

11.7. Найдите главное разбиение графа, представленного на рис. 11.10.
11.8. а) Получите уравнения состояния цепи, показанной на рис. 11.11. б) Выразите токи и напряжения через резисторы, исходя из переменных состояния и переменных источников.

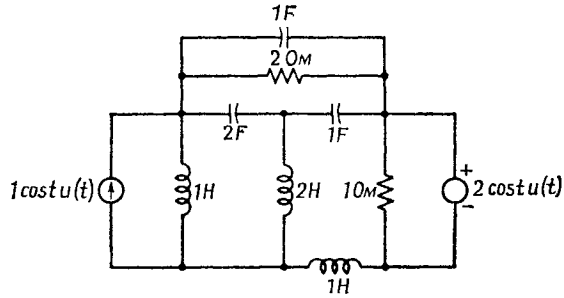


Рис. 11.11.

11.9. *Порядком сложности* электрической цепи N является максимальное число начальных условий, которые можно выбрать для цепи N. Оно совпадает с максимальным числом динамически независимых токов и напряжений на элементах, мгновенные значения которых достаточны для того, чтобы определить мгновенное состояние цепи. Более того, оно также равно числу собственных частот цепи N.

Докажите следующее для RLC-цепи:

а) Порядок сложности N равен числу реактивных элементов, меньшему суммы чисел линейно-независимых сечений, состоящих только из индуктивностей, и чисел линейно-независимых контуров, состоящих только из емкостей.

б) Число ненулевых собственных частот цепи N равно порядку сложности N, меньшему суммы чисел линейно-независимых сечений, состоящих только из емкостей, и чисел линейно-независимых цепей, состоящих только из индуктивностей.

12. Резистивные n-полюсные цепи

12.1. Введение

Цепь называется n-полюсной, если она имеет n пар доступных зажимов для подсоединения внешних устройств, например источников тока или напряжения. Каждая пара таких доступных зажимов называется полюсом. Очевидно, что n-полюсная цепь может иметь максимум $2n$ различных полюсных зажимов, в случае когда каждый полюс имеет свою собственную пару зажимов, и минимум $n+1$ различных полюсных зажимов. Последний случай возникает, например, когда один зажим является общим для всех полюсов. Вершины n-полюсной цепи, отличные от полюсных входов, называются *внутренними вершинами* цепи. Каждый полюс n-полюсной цепи связан с двумя переменными — напряжением между зажимами полюса и током, текущим через полюс. На рис. 12.1 представлена 3-полюсная цепь с 6 зажимами. Ссылки на полюсные напряжения и токи будут приниматься в соответствии с рис. 12.1.

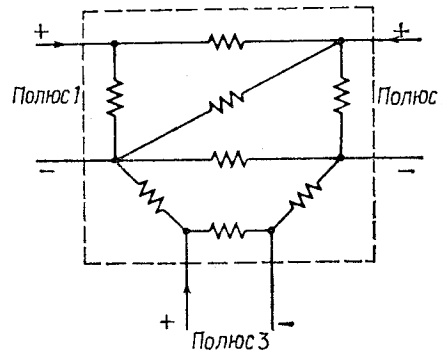


Рис. 12.1. 3-полюсная цепь с 6 зажимами.

При представлении n-полюсной цепи в виде ориентированного графа каждый ее полюс будет рассматриваться как ребро, называемое *полюсным ребром*, связывающим соответствующие зажимы полюса. Полюсное ребро можно рассматривать в качестве внешнего устройства, подключенного к полюсу. Подграф из полюсных ребер вместе с указанием положительных или отрицательных зажимов каждого полюса называется *полюсной конфигурацией* n-полюсной цепи.

Полюсное ребро может быть ориентированным как от положительного зажима к отрицательному, так и наоборот.

В соответствии с обозначениями полюсного напряжения и тока, из рис. 12.1 должно быть очевидным, что для представленного ранее выбора полюсной ориентации полюсное напряжение совпадает с напряжением, связанным с полюсным ребром, а полюсный ток

противоположен по направлению току, связанному с полюсным ребром. Соответствующие соотношения в последнем выборе ориентации полюса очевидны.

О n-полюсной цепи с внешними устройствами, подключенными к полюсам, будем говорить как о *нагруженной n-полюсной цепи.* Полагаем, что нагруженная n-полюсная цепь является связной и в ней не существует внутренних вершин. Если нагруженная полюсная цепь не является связной, то каждую компоненту нагруженной сети можно рассматривать отдельно как многополюсную цепь. Если в данной цепи существуют некоторые внутренние вершины, их можно исключить преобразованием типа звезда — треугольник и получить цепь, эквивалентную исходной в смысле отношения между полюсными переменными. Таким образом, мы сделали два допущения, не нарушая общности рассмотрения. Будем предполагать также, что в цепи отсутствуют параллельные проводимости. Если такие проводимости имеются в цепи, их можно свести к единственной проводимости.

Свойства n-полюсных цепей можно описать любой системой n независимых уравнений от $2n$ полюсных переменных. Особый интерес представляют матрица проводимостей короткого замыкания и матрица полных сопротивлений холостого хода. В описании с помощью матрицы проводимостей короткого замыкания полюсные напряжения выбираются в качестве независимых переменных. *Матрица проводимостей короткого замыкания* $Y=[y_{ij}]$ является матрицей порядка $n\times n$, которая преобразует вектор полюсных напряжений V_p в вектор полюсных токов I_p. Таким образом, имеем

$$YV_p=I_p. \tag{12.1}$$

Если $V_p=[v_1, v_2,\ldots,v_n]^t$ и $I_p=[i_1, i_2,\ldots, i_n]^t$, где v_j и i_j переменные напряжения и тока соответственно, связанные с j-м полюсом, тогда очевидно, что

$$y_{kj}=\frac{i_k}{v_j}\bigg|_{v_i=0 \text{ для всех } i\neq j}.$$

Физически это означает, что y_{kj} является током, текущим через k-й полюс, когда j-й полюс возбуждается источником единичного напряжения, причем все остальные полюса считаются короткозамкнутыми.

Необходимым условием существования матрицы проводимостей является отсутствие циклов в полюсной конфигурации. Иначе окажется невозможным выбор полюсных напряжений в качестве независимых переменных.

В описании с помощью матрицы полных сопротивлений холостого хода n-полюсной цепи в качестве независимых переменных выбираются полюсные токи. *Матрица полных сопротивлений холос-*

того хода цепи $Z=[z_{ij}]$ есть $n\times n$-матрица, которая преобразует вектор полюсных токов I_p в вектор полюсных напряжений V_p. Таким образом, имеем

$$ZI_p=V_p. \tag{12.2}$$

В этом случае можно записать

$$z_{kj}=\frac{v_k}{i_j}\Big|_{i_m=0 \text{ для всех } m\neq j}$$

Физически это означает, что z_{kj} — напряжение на k-м полюсе, когда j-й полюс возбуждается источником единичного тока, причем все остальные полюсы считаются разомкнутыми.

Чтобы существовало описание в виде матрицы полных сопротивлений, необходимо, чтобы в нагруженной n-полюсной цепи (которая предполагается связной) не было сечения, состоящего только из полюсных ребер, поскольку в таком описании полюсные токи выбираются в качестве независимых переменных. Другими словами, для существования матрицы полных сопротивлений требуется, чтобы сеть была связной.

Теперь мы приступаем к выводу уравнений для Y- и Z-матриц n-полюсной RLC-цепи N. Соответствующую нагруженную цепь будем обозначать через $\hat{N}$.

Рассмотрим сначала вывод Y-матрицы цепи N. Пусть T означает полюсную конфигурацию цепи N. Как и раньше, потребуем, чтобы T была ациклической, так что на каждое полюсное ребро можно ссылаться как на независимый источник напряжения. Мы предполагаем, что каждое полюсное ребро направлено от положительного зажима соответствующего полюса к отрицательному.

Пусть T_0 — такой остов цепи $\hat{N}$, что T является подграфом T_0. Ребра подграфа T назовем *полюсными ветвями*, а остальные ребра T_0 — *неполюсными ветвями*. Базисную матрицу сечений нагруженной цепи N относительно T_0 можно записать в виде

$$\left[\begin{array}{c|c} Q_1 & N \\ \hline Q_2 & 0 \end{array}\right], \text{ где подматрица } Q_f=\left[\begin{array}{c} Q_1 \\ \hline Q_2 \end{array}\right]$$

соответствует цепи N, причем строки Q_1 соответствуют полюсным ветвям, а строки Q — неполюсным ветвям.

Пусть вектор-столбцы V_p, V_{nb}, V_e определяются следующим образом: V_p — вектор полюсных напряжений, V_{nb} — вектор напряжений на неполюсных ветвях, V_e — вектор напряжений на ребрах в цепи N. Пусть векторы токов I_p и I_e определяются аналогично. Пусть Y_e обозначает такую диагональную матрицу проводимостей ребер в цепи N, что

$$I_e=Y_eV_e. \tag{12.3}$$

Теперь можно записать максимальную систему уравнений ЗКТ для нагруженной цепи $\hat{N}$ в следующем виде:

$$\left[\begin{array}{c|c} Q_1 & U \\ \hline Q_2 & 0 \end{array}\right]\left[\begin{array}{c} I_e \\ \hline -I_p \end{array}\right]=\left[\begin{array}{c} 0 \\ \hline 0 \end{array}\right], \text{ т. е. } \left[\begin{array}{c} Q_1 \\ \hline Q_2 \end{array}\right][I_e]=\left[\begin{array}{c} I_p \\ \hline 0 \end{array}\right]. \quad (12.4)$$

Также получаем соотношение (преобразование сечения, теорема 11.2)

$$V_e=\left[\begin{array}{c} Q_1 \\ \hline Q_2 \end{array}\right]^t\left[\begin{array}{c} V_p \\ \hline V_{nb} \end{array}\right]. \quad (12.5)$$

Используя формулы (12.3) и (12.5) в выражении (12.4), получим

$$\left[\begin{array}{c|c} Y_{11} & Y_{12} \\ \hline Y_{21} & Y_{22} \end{array}\right]\left[\begin{array}{c} V_p \\ \hline V_{nb} \end{array}\right]=\left[\begin{array}{c} I_p \\ \hline 0 \end{array}\right], \quad (12.6)$$

где $Y_{11}=Q_1Y_eQ_1^t$, $Y_{12}=Q_1Y_eQ_2^t=Y_{21}^t$, $Y_{22}=Q_2Y_eQ_2^t$.
Матрица

$$Y_0=Q_fY_eQ_f^t=\left[\begin{array}{c|c} Y_{11} & Y_{12} \\ \hline Y_{21} & Y_{22} \end{array}\right]$$

называется *матрицей проводимостей сечения* n-полюсной цепи N относительно T_0. Решая вторую часть системы уравнений (12.6), получим

$$V_{nb}=-Y_{22}^{-1}Y_{21}V_p. \quad (12.7)$$

Отметим, что Y_{22} является невырожденной, так как Q_2 имеет максимальный ранг, а Y_e является диагональной с ненулевыми элементами на диагонали. Используя формулу (12.7) в первой части системы уравнений (12.6), а именно $Y_{11}V_p+Y_{12}V_{nb}=I_p$, получаем

$$(Y_{11}-Y_{12}Y_{22}^{-1}Y_{21})V_p=I_p. \quad (12.8)$$

Таким образом, матрица проводимостей короткого замыкания V цепи N задается выражением

$$Y=Y_{11}-Y_{12}Y_{22}^{-1}Y_{21}. \quad (12.9)$$

Заметим, что если полюсная конфигурация T имеет $p>1$ компонент, то цепь N будет иметь $n+p$ полюсных входов. Если $p=1$, то T будет остовом цепи $\hat{N}$. В этом случае $T=T_0$, т. е. не будет существовать неполюсных ребер, и цепь N будет называться *n-полюсной цепью ранга n*. Можно убедиться, что матрица проводимостей Y короткого замыкания n-полюсной цепи ранга n является такой же, как и матрица проводимостей сечения Y_0, и поэтому для такой цепи

$$Y=Q_fY_eQ_f^t. \quad (12.10)$$

Заметим, что в общем случае Q_f не является базисной матрицей сечения цепи $\hat{N}$, хотя фактически она является подматрицей базис-

ной матрицы сечений цепи N по отношению к T_0. Однако на Q_f будет весьма удобно ссылаться как на базисную матрицу сечений цепи N по отношению к T_0. Если допустить присутствие ребер с нулевыми проводимостями, то матрица Q_f фактически становится базисной матрицей сечений цепи N.

Далее рассмотрим вывод Z-матрицы n-полюсной RLC-цепи N. Как и ранее, потребуем, чтобы цепь N была связной, так что полюсные ребра могут стать частью ко-остова дерева некоторого остова T_0 цепи N. Полюсные ребра будем называть *полюсными хордами*, а оставшиеся ребра ко-остова — *неполюсными хордами*. В этом случае можно говорить о каждом полюсном ребре как о представляющем независимый источник тока и ориентировать его от отрицательного зажима полюса к положительному. Базисную цикломатическую матрицу цепи N по отношению к T_0 можно записать в виде

$$\left[\begin{array}{c|c} B_1 & U \\ \hline B_2 & 0 \end{array}\right], \text{ где подматрица } B_f = \left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right]$$

соответствует n-полюсной цепи N, причем строки B_1 соответствуют полюсным, а строки B_2 — неполюсным хордам.

Пусть V_p, V_e, I_p и I_e определяются так, как и ранее. Кроме того, пусть I_{nc} — вектор токов в неполюсных хордах. Теперь можно записать максимальную систему независимых уравнений ЗКН в следующем виде:

$$\left[\begin{array}{c|c} B_1 & U \\ \hline B_2 & 0 \end{array}\right] \left[\begin{array}{c} V_e \\ \hline -V_p \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 0 \end{array}\right]. \tag{12.11}$$

Если Z_e — диагональная матрица полных реберных сопротивлений в цепи N, то

$$V_e = Z_e I_e. \tag{12.12}$$

Также существует соотношение (контурное преобразование, теорема 11.2)

$$I_e = \left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right]^t \left[\begin{array}{c} I_p \\ \hline I_{nc} \end{array}\right]. \tag{12.13}$$

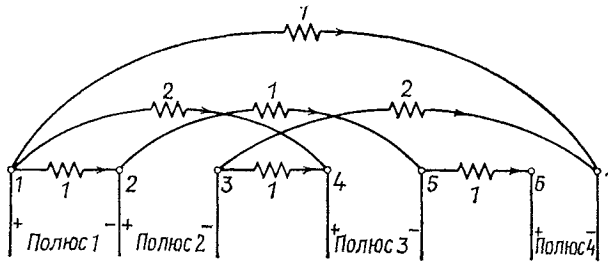
Используя формулы (12.11) — (12.13), получим

$$\left[\begin{array}{c|c} Z_{11} & Z_{12} \\ \hline Z_{21} & Z_{22} \end{array}\right] \left[\begin{array}{c} I_p \\ \hline I_{nc} \end{array}\right] = \left[\begin{array}{c} V_p \\ \hline 0 \end{array}\right], \tag{12.14}$$

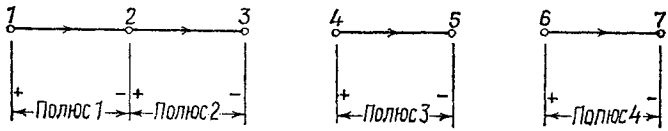
где $Z_{11} = B_1 Z_e B_1^t$, $Z_{12} = B_1 Z_e B_2^t = Z_{21}^t$, $Z_{22} = B_2 Z_e B_2^t$.
Решая вторую часть системы уравнений (12.14), получим

$$I_{nc} = -Z_{22}^{-1} Z_{21} I_p. \tag{12.15}$$

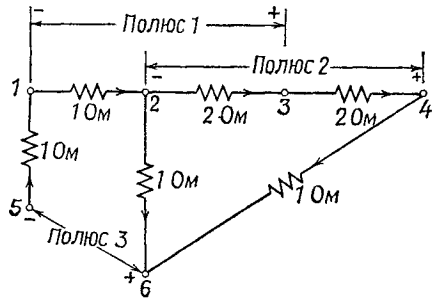
Можно заметить, что Z_{22} является невырожденной, поскольку B_2 имеет максимальный ранг, а Z_e является диагональной матрицей с ненулевым элементом на диагонали.



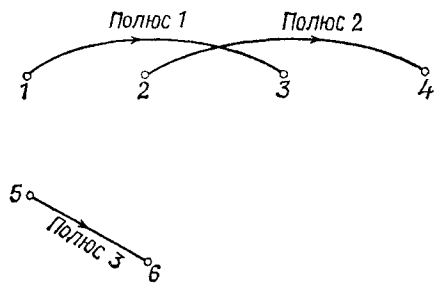
а



б



в



г

Рис. 12.2.

а — резистивная 4-полюсная цепь (все проводимости даны в сименсах); *б* — полюсная конфигурация цепи; *в* — резистивная 3-полюсная цепь; *г* — полюсная конфигурация цепи.

Наконец, используя формулу (12.15) в первой части системы уравнений (12.14), получим

$$(Z_{11} - Z_{12}Z_{22}^{-1}Z_{21})\, I_p = V_p. \tag{12.16}$$

Таким образом, матрица Z полных сопротивлений холостого хода цепи N задается выражением

$$Z = Z_{11} - Z_{12}Z_{22}^{-1}Z_{21}. \tag{12.17}$$

Если цепь N не имеет контуров, то она будет остовом цепи $\hat{N}$ и, следовательно, полюсные ребра будут образовывать соответствующий ко-остов. В этом случае неполюсные хорды будут отсутствовать и о цепи N можно говорить как о *n-полюсной цепи с цикломатическим числом n*. Таким образом, матрица Z полных сопротивлений холостого хода n-полюсной цепи с цикломатическим числом n задается выражением

$$Z = B_f Z_e B_f^t. \tag{12.18}$$

Как и в случае матрицы Q_f будем называть B_f базисной цикломатической матрицей цепи N. Ниже на примерах показан вывод Y- и Z-матриц n-полюсной цепи.

Сначала вычислим Y-матрицу сопротивлений 4-полюсной цепи, представленной на рис. 12.2, *а*. Полюсная конфигурация этой цепи показана на рис. 12.2, *б*. Выбирая ребра (3.4) и (5.6) в качестве неполюсных ветвей, получим матрицу

$$Q_f = \left[\begin{array}{c} Q_1 \\ \hline Q_2 \end{array}\right].$$

$$= \begin{array}{l} \\ \textit{Полюс 1} \\ \textit{Полюс 2} \\ \textit{Полюс 3} \\ \textit{Полюс 4} \\ \hline \textit{Неполюсная ветвь } (3,4) \\ \textit{Неполюсная ветвь } (5,6) \end{array} \begin{array}{c} (1,2)(3,4)(5,6)(1,4)(2,5)(3,7)(1,7) \\ \left[\begin{array}{ccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right] \end{array}.$$

Для этой цепи

$$Y_e = \operatorname{diag}[g_{12}g_{34}g_{56}g_{14}g_{25}g_{37}g_{17}] = \\ = \operatorname{diag}[1 \;\; 1 \;\; 1 \;\; 2 \;\; 1 \;\; 2 \;\; 1],$$

где g_{ij} — проводимость ребра (i, j), связывающего вершины i и j. Теперь получим

$$Q_f Y_e Q_f' = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$$

$$= \left[\begin{array}{cccc|cc} 4 & 3 & 1 & 1 & 3 & 1 \\ 3 & 4 & 2 & 1 & 4 & 1 \\ 1 & 2 & 4 & 3 & 4 & 3 \\ 1 & 1 & 3 & 3 & 3 & 3 \\ \hline 3 & 4 & 4 & 3 & 7 & 3 \\ 1 & 1 & 3 & 3 & 3 & 4 \end{array}\right].$$

Матрица проводимостей короткого замыкания 4-полюсной цепи, показанной на рис. 12.2, *а*, дается выражением

$$Y = Y_{11} - Y_{12} Y_{22}^{-1} Y_{21} =$$

$$= \frac{1}{19} \begin{bmatrix} 51 & 23 & -11 & -2 \\ 23 & 29 & 1 & -5 \\ -11 & 1 & 21 & 9 \\ -2 & -5 & 9 & 12 \end{bmatrix}.$$

Теперь рассмотрим резистивную 3-полюсную цепь, показанную на рис. 12.2, *в*. Полюсная конфигурация этой цепи показана на рис. 12.2, *г*. Выбирая ребро (4.6) в качестве неполюсной хорды, получим матрицу B_f в виде

$$B_f = \left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right] =$$

$$= \begin{array}{l} \\ \textit{Полюс 1} \\ \textit{Полюс 2} \\ \textit{Полюс 3} \\ \textit{Неполюсная хорда } (4,6) \end{array} \begin{array}{c} \begin{array}{cccccc} (1,2) & (2,3) & (3,4) & (1,5) & (2,6) & (4,6) \end{array} \\ \left[\begin{array}{cccccc} -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & -1 & 0 \\ \hline 0 & 1 & 1 & 0 & -1 & 1 \end{array}\right] \end{array}.$$

Матрица Z_e для этой цепи имеет вид

$$Z_e = \operatorname{diag} [r_{12} \quad r_{23} \quad r_{34} \quad r_{15} \quad r_{26} \quad r_{46}]$$
$$= \operatorname{diag} [1 \quad 2 \quad 2 \quad 1 \quad 1 \quad 1].$$

Теперь получаем

$$B_f Z_e B_f^t = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$

$$= \left[\begin{array}{ccc|c} 3 & 2 & 1 & -2 \\ 2 & 4 & 0 & -4 \\ 1 & 0 & 3 & 1 \\ \hline -2 & -4 & 1 & 6 \end{array}\right].$$

Матрица полных сопротивлений холостого хода этой цепи

$$Z = Z_{11} - Z_{12} Z_{22}^{-1} Z_{21} = \frac{1}{6} \begin{bmatrix} 14 & 4 & 8 \\ 4 & 8 & 4 \\ 8 & 4 & 17 \end{bmatrix}.$$

Как можно видеть из выражений (12.10) и (12.18), свойства Y-матрицы n-полюсной цепи ранга n и Z-матрицы той же цепи с цикломатическим числом n тесно связаны со свойствами соответствующих базисной матрицы сечений и цикломатической матрицы.

В следующем разделе этой главы рассматриваются несколько свойств этих матриц и обсуждаются процедуры их реализации.

12.2. *Y*-матрицы резистивной (*n*-полюсной цепи ранга *n*

В этом разделе обсуждаются некоторые важные свойства Y-матриц резистивных n-полюсных цепей ранга n. Очевидно, что эти n-полюсные цепи имеют $n+1$ узлов. Поэтому мы их называем $(n+1)$-узловыми n-полюсными цепями. Во всех рассуждениях принимаем, что в рассматриваемой цепи не существует отрицательных проводимостей.

12.2.1. Основные свойства

Рассмотрим $(n+1)$-узловую резистивную n-полюсную цепь с полюсной конфигурацией T. Из выражения (12.10) известно, что матрица проводимостей Y короткого замыкания цепи N задается выражением

$$Y = Q_f G_e Q_f^t, \tag{12.19}$$

где Q_f — базисная матрица сечений цепи N по отношению к T, а G_e — диагональная матрица реберных проводимостей. Например,

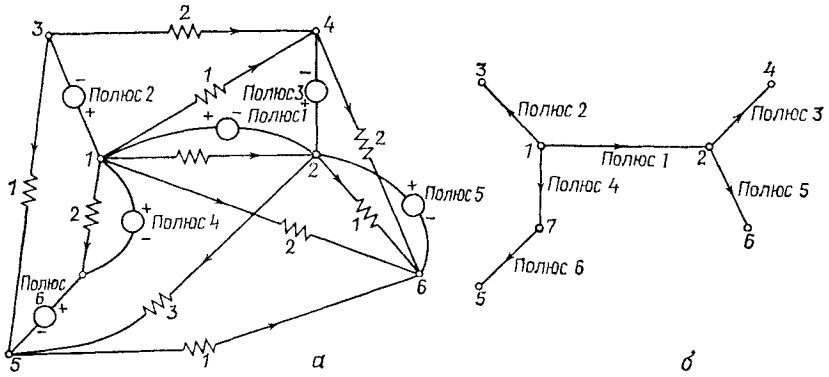


Рис. 12.3. 6-полюсная цепь и ее полюсная конфигурация (все проводимости даны в сименсах).

для 6-полюсной цепи, показанной на рис. 12.3, имеем

$$Q_f = \begin{array}{c} \begin{matrix} (1,2) & (1,4) & (1,6) & (1,7) & (2,5) & (2,6) & (3,4) & (3,5) & (4,6) & (5,6) \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \end{array}$$

и

$$G_e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Следовательно, для этой цепи

$$Y = Q_f G_e Q_f'$$

$$= \begin{bmatrix} 10 & -2 & 3 & -4 & 3 & -4 \\ -2 & 3 & -2 & -1 & 0 & -1 \\ 3 & -2 & 5 & 0 & -2 & 0 \\ -4 & -1 & 0 & 7 & -1 & 5 \\ 3 & 0 & -2 & -1 & 6 & -1 \\ & 1 & 0 & 5 & -1 & 5 \end{bmatrix}. \qquad (12.20)$$

Ниже будем обозначать через g_{ij} проводимость ребра (i, j), связывающего вершины i и j. Полагаем, что строка i в матрице Q_f соответствует полюсу i. Для дальнейшего обсуждения требуется ввести несколько определений.

Говорят, что проводимость g_{ij} *стягивает* полюс цепи N, если полюс лежит на единственном пути в T между вершинами i и j. Например, в цепи на рис. 12.3 проводимость g_{35} стягивает все те полюса, которые входят в образуемый ею базисный цикл.

В литературе по теории цепей висячая вершина в T обычно называется *концевой вершиной*. Аналогично висячий полюс в T называется *концевым полюсом*. Поэтому во всех обсуждениях данной главы используются термины «концевая вершина» и «концевой полюс» вместо «висячая вершина» и «висячий полюс». Далее используется также термин *коротко-замкнутая* вместо *стянутая*.

Дерево называется *линейным*, если оно имеет точно две концевые вершины. Другими словами, линейное дерево есть путь. Например, полюса 1, 2 и 5 на рис. 12.3 образуют линейное дерево.

Дерево называется *звездой*, если все его ребра имеют общую вершину, называемую *вершиной звезды*. Другими словами, звезда имеет только одну неконцевую вершину. Например, полюса 1, 2 и 4 на рис. 12.3 образуют звезду.

Говорят, что два полюса i и j *подобно ориентированы*, если они ориентированы в одном направлении в любом пути в T, содержащем эти полюса. В противном случае они называются *противоположно ориентированными*. Рассмотрим произвольный диагональный элемент y_{ii} матрицы Y. По теореме 2.15 ненулевые элементы в i-й строке Q_f соответствуют всем тем проводимостям, которые стягивают полюс i. Поэтому из (12.19) получаем

y_{ii}=Сумма всех проводимостей, которые стягивают полюс i. (12.21)

Рассмотрим далее произвольный недиагональный элемент y_{ij} матрицы Y. Пусть q_{ir} обозначает элемент (i, r) матрицы Q. Тогда известно (упражнение 6.1), что если для любых i и j и любых r и s произведения $q_{ir}q_{jr}$ и $q_{is}q_{js}$ являются ненулевыми, то они имеют одинаковый знак.

Поэтому из выражения (12.19) получим

$$y_{ij} = \pm \begin{cases} \text{Сумма всех проводимостей, которые} \\ \text{стягивают оба полюса } i \text{ и } j. \end{cases} \qquad (12.22)$$

Теперь необходимо определить знак y_{ij}. Предположим, что для некоторого r произведения $q_{ir}q_{jr} \neq 0$. Тогда знак y_{ij} будет тем же самым, что и знак $q_{ir}\, q_{jr}$. Поскольку сомножители произведения $q_{ir}\, q_{jr}$ не равны нулю, очевидно, что оба полюса i и j принадлежат базисному циклу C, образованному проводимостью, которая соответствует столбцу r в Q_f. Имеем два случая. Сначала допустим, что полюса i и j подобно ориентированы. Тогда ориентации обоих полюсов будут совпадать или не совпадать с ориентацией цикла C. Другими словами, оба элемента q_{ir} и q_{jr} будут иметь одинаковый знак. Следовательно, в этом случае y_{ij} — положительно.

С другой стороны, если полюса i и j противоположно ориентированы, то они будут иметь и обратную относительную ориентацию в цикле C, так что q_{ir} и q_{ijr} будут иметь обратные знаки. В этом случае y_{ij} — отрицательно. Отсюда имеем следующую теорему:

Теорема 12.1. Если полюса i и j подобно ориентированы, то y_{ij} положительно, в противном случае — отрицательно. Следующий результат легко получается из теоремы 12.1.

Теорема 12.2. 1. Если полюса i, j и k образуют линейное дерево, когда все другие полюса являются короткозамкнутыми, то $y_{ij}y_{ik}y_{jk} > 0$. 2. Если полюса i, j и k образуют звезду, когда все другие полюса короткозамкнуты, то $y_{ij}y_{ik}y_{jk} \leqslant 0$. Можно проиллюстрировать справедливость теорем 12.1 и 12.2 с помощью Y-матрицы 6-полюсной цепи, изображенной на рис. 12.3.

Теорема 12.3. Предположим, что полюсы i, j и k образуют линейное дерево, когда все остальные полюса короткозамкнуты. Далее, допустим, что в этом линейном дереве полюса имеют порядок i, j и k. Тогда $|y_{ij}| \geqslant |y_{ik}|$. Приведенный выше результат следует из того факта, что каждая проводимость, которая стягивает полюса i и k, также неизбежно стягивает полюс j. Важное следствие из теоремы 12.3 формируется следующим образом:

Следствие 12.3.1. Пусть y_{ij} — ненулевой элемент матрицы Y, имеющий минимальное значение. Тогда существует точно одна проводимость, которая стягивает полюса i и j. Величина такой проводимости равна $|y_{ij}|$. Другими словами, строки i и j матрицы Q_f содержат ненулевые элементы одновременно только в одном столбце. (Предполагается, что параллельные проводимости объединены в одну.)

Например, в матрице Y из выражения (12.20) y_{24} — ненулевой элемент с минимальным значением, а g_{35} (рис. 12.3) — единственная проводимость, стягивающая оба полюса *2* и *4*.

Теорема 12.4. Пусть максимальное значение элементов в i-строке матрицы Y равно M_i, и пусть элементы, имеющие значение M_i, встречаются одновременно в этой строке в позициях столбцов j, k, l, . . ., и только в них. Тогда подграф, образованный полюсами i, j, k, l, . . ., — связный.

Доказательство. Если подграф, образованный множеством полюсов $\{i, j, k, l, \ldots\}$, не является связным, то в этом множестве существует некоторый полюс d, которого нет в компоненте, содержащей полюс i. Так как T является связным, то в нем существует такой полюс p, не принадлежащий множеству $\{i, j, k, l, \ldots\}$, что полюса $\{d, p, i\}$ образуют линейное дерево с тремя полюсами, появляющимися в этом порядке, когда все другие полюса в T короткозамкнуты.

Отметим, что по теореме 12.3 $|y_{ip}| \geqslant |y_{id}|$. Так как y_{id} — элемент, имеющий максимальное значение в i-й строке матрицы Y, то следует, что $|y_{ip}| = |y_{id}|$. По-

этому p — также элемент множества $\{i, j, k, l\}$, что приводит к противоречию. Отсюда следует, что подграф, состоящий из полюсов $\{i, j, k, l, \ldots\}$, связный.

Теорема 12.5. Пусть наименьший элемент в строке i матрицы Y есть m_i, где m_i, возможно, равен нулю. Пусть в этой строке элемент с меньшим значением появляется одновременно в позиции столбцов $p, q, r, \ldots$, и только в них. Тогда подграф, образованный всеми полюсами T, отличными от $p, q, r, \ldots$, связен.

Доказательство. Если граф, который определен в теореме, не является связным, то существует полюс j из множества $\{p, q, r, \ldots\}$, а полюс k, не принадлежащий $\{p, q, r, \ldots\}$, такой, что полюсы i, j и k образуют линейное дерево, когда все другие полюсы T короткозамкнуты. По теореме 12.3 имеем $|y_{ij}| \geqslant |y_{ik}|$. Поскольку $|y_{ij}| = m_i$ (минимальному по величине элементу в i-й строке из Y), то следует, что $|y_{ik}| = m_i$ и k принадлежит множеству $\{p, q, r, \ldots\}$, что приводит к противоречию.

Обе приведенные выше теоремы иллюстрируются с помощью матрицы Y из выражения (12 20).

Из формул (12.21) и (12.22) видно, что $|y_{ii}| > |y_{ij}|$ для всех i и j. Это является только частным случаем более общего свойства, называемого *схемностью*, которое обнаруживает Y-матрица. Вещественная симметрическая матрица называется *схемной*, если каждый главный минор матрицы не меньше значения любого другого минора, построенного из тех же самых строк (или столбцов).

В работе [12.1] показано, что любая матрица $P = KDK^t$, где K — унимодулярная матрица, а D — диагональная матрица с вещественными положительными элементами, является схемной. Так как матрица Q_f унимодулярна, то из (12.19) можно заключить, что матрица проводимостей короткого замыкания $(n+1)$-узловой резистивной n-полюсной цепи является схемной. Фактически также можно показать, что матрицы проводимостей короткого замыкания и сопротивлений холостого хода любой n-полюсной цепи являются схемными.

Например, рассмотрим миноры

$$M_1 = \begin{vmatrix} 10 & -2 & 3 \\ -2 & 3 & -2 \\ 3 & -2 & 5 \end{vmatrix} \quad \text{и} \quad M_2 = \begin{vmatrix} 10 & 3 & -4 \\ -2 & -2 & -1 \\ 3 & 5 & 0 \end{vmatrix}$$

матрицы Y из выражения (12.20). Оба этих минора построены из первых трех строк матрицы, причем первый является главным минором. Можно проверить, что $M_1 = 87$ и $M_2 = 57$, убеждаясь, что $M_1 \geqslant |M_2|$. Для детального обсуждения свойств схемных матриц можно обратиться к работе [12.2]. Далее охарактеризуем Y-матрицы $(n+1)$-узловой резистивной n-полюсной цепи, имеющей конфигурацию в виде линейного дерева или звезды. В последующем обсуждении говорим, что матрица Y представима в особой форме, если ее можно привести к этой форме одним или более применением следующих операций:

1) переставить две любые строки и соответствующие столбцы; 2) изменить знаки всех элементов в любой строке и соответствующем столбце.

Заметим, что эти две операции соответствуют перенумерации и изменению ориентации некоторых полюсов.

12.2.2. Полюсная конфигурация в виде звезды

Рассмотрим Y-матрицу $(n+1)$-узловой n-полюсной цепи N, имеющей полюсную конфигурацию в виде звезды. Положим, что все полюса в T ориентированы к вершине звезды (рис. 12.4). Обозначим центральную вершину звезды через 0. Пусть другие вершины звезды обозначаются через 1, 2, . . ., n, так что вершины i и 0 образуют входы полюса i. Тогда из выражения (12.21) имеем

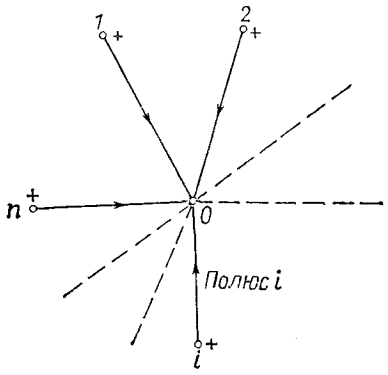


Рис. 12.4. Звезда.

$$y_{ii} = \sum_{\substack{j=0 \\ j\neq i}}^{n} g_{ij}. \qquad (12.23)$$

Далее, по теореме 12.1

$$y_{ij} \leqslant 0. \qquad (12.24)$$

Поэтому из выражения (12.22) получим

$$y_{ij} = -g_{ij}. \qquad (12.25)$$

Теперь можно видеть из формул (12.23) и (12.25), что для всех i

$$y_{ii} \geqslant \sum_{\substack{j=1 \\ j\neq i}}^{n} |y_{ij}|. \qquad (12.26)$$

Вещественная симметричная матрица $Y=[y_{ij}]$ является *гипердоминантной*, если она удовлетворяет неравенствам (12.24) и (12.26), т. е. 1) y_{ij} является неположительным для всех i и $j\neq i$ и 2) $y_{ii} \geqslant \sum_{\substack{j=1 \\ j\neq i}}^{n} |y_{ij}|$.

Можно легко получить из формул (12.23) и (12.25) следующие выражения для проводимостей цепи N: $g_{ij}=-y_{ij}$, $j\neq i$, $i\neq 0$, $j\neq 0$, $g_{i0}=y_{ii}-\sum_{j=1}^{n}|y_{ij}|$, $i\neq 0$.

Проводимости g_{ij}, рассчитываемые, как показано выше, будут неотрицательными, если матрица Y удовлетворяет неравенствам (12.24) и (12.26). Таким образом, мы доказали следующее:

Теорема 12.6. Вещественная симметричная матрица Y реализуема как матрица проводимостей короткого замыкания $(n+1)$-узловой резистивной n-полюсной цепи, имеющей полюсную конфигурацию в виде звезды и не содержащей отрицательных проводимостей тогда и только тогда, когда она представима в гипердоминантной форме.

12.2.3. Полюсная конфигурация в виде линейного дерева

Рассмотрим Y-матрицу $(n+1)$-узловой резистивной n-полюсной цепи, имеющей полюсную конфигурацию в виде линейного дерева T. Пусть вершины T помечены последовательно, начиная от концевой

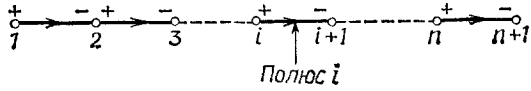


Рис. 12.5. Линейное дерево.

вершины, и пусть вершины i и $i+1$ образуют положительные и отрицательные входы полюса i (рис. 12.5). Так как все полюса дерева T подобно ориентированы, по теореме 12.1 имеем

$$y_{ij} \geqslant 0 \text{ для всех } i \text{ и } j. \tag{12.27}$$

Далее по теореме 12.3 имеем для всех i

$$y_{ii} \geqslant y_{i,\,i+1} \geqslant \ldots \geqslant y_{in}, \tag{12.28}$$

$$y_{ii} \geqslant y_{i-1,\,i} \geqslant \ldots \geqslant y_{in}. \tag{12.29}$$

Выведем простое выражение для проводимости g_{ij}, исходя из элементов Y-матрицы.

Предположим, что все полюса будут короткозамкнутыми, кроме полюсов $i-1$, i, $j-1$ и j, где $i>1$ и $i+1<j\leqslant n$. Тогда в результате получится цепь, представленная на рис. 12.6, в которой отмечены только интересующие нас проводимости. Теперь легко показать, что $y_{i-1,\,j-1}=e+c$, $y_{i-1,\,j}=e$, $y_{i,\,j-1}=g_{ij}+c+d+e$, $y_{i,\,j}=e+d$. Из приведенного выше получим

Рис. 12.6.

$$g_{ij}=(y_{i,\,j-1}+y_{i-1,\,j})-(y_{ij}+y_{i-1,\,j-1}), \quad i>1,\ i+1<j\leqslant n. \tag{12.30}$$

Можно также доказать следующее:

$$\begin{aligned} g_{i,\,i+1} &= (y_{ii}+y_{i-1,\,i+1})-(y_{i,\,i+1}+y_{i-1,\,i}), \quad 1<i<n,\\ g_{1j} &= y_{1,\,j-1}-y_{1j}, \quad 2\leqslant j\leqslant n,\\ g_{1,\,n+1} &= y_{1n},\\ g_{i,\,n+1} &= y_{in}-y_{i-1,\,n}, \quad 1<i\leqslant n. \end{aligned} \tag{12.31}$$

Вещественная симметричная матрица $Y=[y_{ij}]$ называется *однородно сужающейся*, если 1) $y_{ij}\geqslant 0$ для всех i и j; 2) $y_{ij}\geqslant y_{i,j+1}$ для $j\geqslant i$; 3) $y_{ij}\geqslant y_{i-1,\,j}$ для $j\geqslant i$; 4) $y_{i,\,j-1}+y_{i-1,\,j}\geqslant y_{ij}+y_{i-1,\,j-1}$ для $i>1$, $i+1\leqslant j\leqslant n$. Из выражения (12.27) с помощью (12.31) получаем следующую теорему:

Теорема 12.7. Вещественная симметричная матрица $Y=[y_{ij}]$ является реализуемой как матрица проводимостей короткого замыкания $(n+1)$-узловой резистивной n-полюсной цепи, имеющей полюсную конфигурацию в виде линейного дерева и не содержащей отрицательных проводимостей, тогда и только тогда, когда матрица Y представима в однородно сужающейся форме.

Для заданной однородно сужающейся матрицы автор работы [12.3] разработал следующую простую процедуру получения проводимостей n-полюсной цепи, реализующей матрицу Y.

1. Сначала построить из заданной однородно сужающейся $n\times n$-матрицы Y такую новую $n\times n$-матрицу Y', что а) первая строка матрицы Y' является такой же, как и первая строка матрицы Y; б) для всех $i>1$ i-я строка матрицы Y' отличается от i-й и $(i-1)$-й строк матрицы Y.

2. Построить такую $n\times n$-матрицу Y'', что а) последний столбец матрицы Y'' является таким же, как и последний столбец матрицы Y'; б) для всех $i<n$ i-й столбец матрицы Y'' отличается от i-го и $(i+1)$-го столбцов матрицы Y'. Тогда можно видеть, что $g_{ij}=y''_{i,\,j-1}$ для всех $j>i$. Например, для однородно сужающейся матрицы соответствующими Y' и Y''-матрицами будут

$$Y=\begin{bmatrix} 5 & 4 & 3 & 1 & 1\\ 4 & 5 & 3 & 1 & 1\\ 3 & 3 & 11 & 8 & 5\\ 1 & 1 & 8 & 9 & 5\\ 1 & 1 & 5 & 5 & 6 \end{bmatrix}$$

$$Y'=\begin{bmatrix} 5 & 4 & 3 & 1 & 1\\ & 1 & 0 & 0 & 0\\ & & 8 & 7 & 4\\ & & & 1 & 0\\ & & & & 1 \end{bmatrix},$$

$$Y''=\begin{bmatrix} 1 & 1 & 2 & 0 & 1\\ & 1 & 0 & 0 & 0\\ & & 1 & 3 & 4\\ & & & 1 & 0\\ & & & & 1 \end{bmatrix}$$

$$=\begin{bmatrix} g_{12} & g_{13} & g_{14} & g_{15} & g_{16}\\ & g_{23} & g_{24} & g_{25} & g_{26}\\ & & g_{34} & g_{35} & g_{36}\\ & & & g_{45} & g_{46}\\ & & & & g_{56} \end{bmatrix}.$$

В приведенных выше матрицах Y' и Y'' не показаны элементы ниже диагоналей, так как эти элементы не представляют интереса при вычислении проводимостей g_{ij}.

12.2.4. Полюсное преобразование

Рассмотрим две различные $(n+1)$-узловые n-полюсные цепи N и N^*, построенные на основе одной и той же резистивной цепи. Пусть T и T^* являются полюсными конфигурациями цепей N и N^* соответственно. Выведем выражение, связывающее матрицы проводимостей короткого замыкания Y и Y^* этих n-полюсных цепей. Пусть Q_f и Q_f^* являются базисными матрицами сечений цепей N и N^* по отношению к T и T^* соответственно. Тогда $Y=Q_f G_e Q_f^t$ и $Y^*=Q_f^* G_e (Q_f^*)^t$, где G_e — диагональная матрица проводимости ребер рассматриваемой резистивной цепи.

Рассмотрим граф $T'=T \cup T^*$. Базисную матрицу сечений T' по отношению к T можно записать следующим образом:

$$\begin{matrix} T-T^* & & T^* \\ [P & | & M]. \end{matrix}$$

Затем можно выразить Q_f^* в виде

$$Q_f^* = M^{-1} Q_f. \tag{12.32}$$

Следовательно,

$$Y^* = Q_f^* G_e (Q_f^*)^t = M^{-1} Q_f G_e Q_f^t (M^{-1})^t = M^{-1} Y (M^{-1})^t. \tag{12.33}$$

Теперь из формулы (11.8) имеем $V_p^* = M^t V_p$, где V_p и V_p^* — векторы полюсных напряжений цепей N и N^* соответственно. Таким образом,

$$V_p = (M^{-1})^t V_p^*. \tag{12.34}$$

Объединяя выражения (12.33) и (12.34), получаем следующую теорему:

Теорема 12.8. Пусть N и N^* являются n-полюсными цепями, построенными на основе одной и той же резистивной цепи. Если $V_p = K V_p^*$, то $Y^* = K^t Y K$.

Заметим, что матрицу K в этой теореме легко получить, рассматривая T и T^*. Например, рассмотрим две n-полюсные цепи N и N^*, представленные на рис. 12.7. Эти n-полюсные цепи строятся из одной и той же исходной цепи. Матрица проводимостей короткого замыкания цепи N имеет вид

$$Y = \begin{bmatrix} 3 & 2 & 0 & 1 & -1 & 2 \\ 2 & 5 & -2 & 0 & 2 & 2 \\ 0 & -2 & 9 & 0 & -3 & 0 \\ 1 & 0 & 0 & 9 & -1 & -1 \\ -1 & 2 & -3 & -1 & 8 & 0 \\ 2 & 2 & 0 & -1 & 0 & 8 \end{bmatrix}.$$

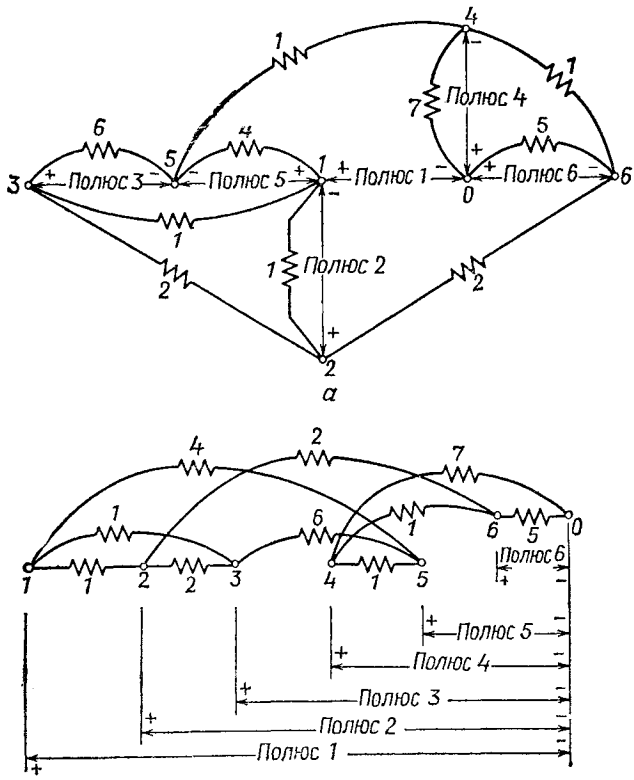


Рис. 12.7.

а — цепь N; *б* — цепь N^* (все проводимости приведены в сименсах).

Векторы V_p и V_p^* взаимосвязаны следующим образом:

$$V_p = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ v_3^* \\ v_4^* \\ v_5^* \\ v_6^* \end{bmatrix} = KV_p^*.$$

Теперь можно убедиться, что

$$Y^* = K'YK = \begin{bmatrix} 6 & -1 & -1 & 0 & -4 & 0 \\ -1 & 5 & -2 & 0 & 0 & -2 \\ -1 & -2 & 9 & 0 & -6 & 0 \\ 0 & 0 & 0 & 9 & -1 & -1 \\ -4 & 0 & -6 & -1 & 11 & 0 \\ 0 & -2 & 0 & -1 & 0 & 8 \end{bmatrix}.$$

12.3. Реализация $(n+1)$-узловых резистивных n-полюсных цепей (подход Седербаума)

В этом разделе рассматривается задача реализации вещественной симметричной $n \times n$-матрицы Y как матрицы проводимостей короткого замыкания $(n+1)$-узловой резистивной n-полюсной цепи, не содержащей отрицательных проводимостей. Для решения этой задачи имеется два основных подхода, предложенных Седербаумом и Гуиллемином. Подход Седербаума [12.1] требует разложения матрицы Y в произведение $Y = Q_f G_e Q_f^t$, где Q_f — матрица, элементы которой равны — 1, 1 или 0, а G_e — диагональная матрица с неотрицательными элементами, представляющими проводимости ребер искомой n-полюсной цепи N. Реализация матрицы Y достигается в том случае, если $[U\ Q_f]$ можно реализовать как базисную матрицу сечений связного графа. Тогда этот связный граф будет графом нагруженной n-полюсной цепи $\hat{N}$, причем столбцы Q_f соответствуют ребрам, представляющим проводимости искомой цепи N. Фактически, как будет показано ниже, подход Седербаума также применим и к реализации матриц холостого хода n-полюсных цепей с цикломатическим числом n.

Подход Гуиллемина [12.3] требует определения полюсной конфигурации (если она существует) и преобразования матрицы Y в матрицу Y', которая соответствует полюсной конфигурации в виде звезды. Если матрица Y' является гипердоминантной, то реализации (как было показано в предыдущем разделе) можно достичь просмотром. Посредством предполагаемого переопределения полюсов эта же цепь может также реализовать данную матрицу Y.

Ниже описывается алгоритм Седербуама, позволяющий разложить матрицу Y в произведение $Y = Q_f G_e Q_f^t$. В следующем разделе рассматривается реализация $[U\ Q_f]$ как базисной матрицы сечений. Алгоритм, основанный на подходе Гуиллемина для реализации Y-матрицы $(n+1)$-узловой n-полюсной цепи, обсуждается в разд. 12.5. Пусть $\hat{q}_1, \hat{q}_2, \ldots$ — столбцы матрицы Q_f. Пусть, далее, $g_1, g_2, \ldots$ — диагональные элементы G_e. Тогда $\hat{q}_i$ является i-м столбцом Q_f, а g_i является i-м диагональным элементом G_e. Заметим, что $\hat{q}_i$ соответствует g_i. Далее, элементы $\hat{q}_i$ будут обозна-

чаться как q_{1i}, q_{2i}, . . ., q_{ni}. Начиная с матрицы $Y^{(0)}=Y$, алгоритм Седербаума позволяет строить последовательность матриц $Y^{(0)}$, $Y^{(1)}$, . . ., соответствующих удалению на каждом шаге из цепи соответственно обозначенной проводимости. Этот процесс продолжается до тех пор, пока не получится диагональная матрица или не будет выяснено, что декомпозиция невозможна.

На каждом шаге алгоритма необходимо рассматривать два случая:

Случай 1. Матрица Y — диагональная с k нулевыми диагональными элементами.

Пусть Y' — матрица, полученная удалением из матрицы Y всех строк и столбцов, которые содержат нули на диагонали. Легко видеть, что декомпозиция в этом случае будет иметь вид

$$Y=\left[\frac{U}{0}\right]Y'\left[\frac{U}{0}\right]^t,$$

где считается, что нулевые диагональные элементы матрицы Y находятся на последних k позициях. Например, матрицу

$$\begin{bmatrix}3&0&0&0&0\\0&1&0&0&0\\0&0&2&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}$$

можно представить в виде декомпозиции

$$Y=\left[\begin{array}{ccc}1&0&0\\0&1&0\\0&0&1\\\hline 0&0&0\\0&0&0\end{array}\right]\begin{bmatrix}3&0&0\\0&1&0\\0&0&2\end{bmatrix}\left[\begin{array}{ccc|cc}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\end{array}\right].$$

Случай 2. Матрица Y имеет некоторые ненулевые недиагональные элементы.

В этом случае сначала выбираем ненулевой недиагональный элемент с наименьшим значением. Допустим, что y_{12} — такой элемент матрицы Y. Если Y является реализуемой, то из следствия 12.3.1 известно, что в искомой цепи N должно существовать ребро со значением проводимости, равным $|y_{12}|$. Другими словами, G_e должно иметь диагональный элемент, равный $|y_{12}|$. Не нарушая общности, допустим, что $g_1=|y_{12}|$. Столбец q_1, соответствующий столбцу g_1, можно определить следующим образом. Очевидно, что ребро с величиной проводимости $|y_{12}|$ стягивает оба полюса 1 и 2 цепи N. Поэтому два первых элемента q_1 являются ненулевыми. Согласно следствию 12.3.1, первые две строки Q_f не могут иметь

ненулевые элементы одновременно во всех столбцах, кроме первого. Поэтому произвольно приписываем $+1$ к первому элементу $\hat{q}_1$, фиксируя таким образом ориентацию ребра, имеющего проводимость q_1 относительно полюса 1. Тогда, согласно теореме 12.1, знак второго элемента в $\hat{q}_1$ является таким же, как и знак y_{12}. Таким образом,

$$\hat{q}_1 = \begin{bmatrix} 1 \\ \pm 1 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

Заметим, что если y_{rs}— ненулевой недиагональный элемент с минимальным значением, то r-й и s-й элементы $\hat{q}_1$ будут определяться аналогично. Теперь остается получить остальные элементы $\hat{q}_1$. Этого можно достичь, используя следующее правило, основанное на теореме 12.2:

$$q_{j1} = \begin{cases} 1, \text{ если } y_{12}y_{1j}y_{2j} > 0 \text{ и } y_{1j} > 0; \\ -1, \text{ если } y_{12}y_{1j}y_{2j} > 0 \text{ и } y_{1j} < 0; \\ 0 \text{ в противном случае.} \end{cases}$$

Этим заканчивается обсуждение двух случаев.

На следующем шаге необходимо найти второй столбец матрицы Q_f и соответствующий диагональный элемент g_2 из матрицы G_e. Это можно сделать рассмотрением матрицы проводимостей короткого замыкания $Y^{(1)}$ цепи $N^{(1)}$, которая получается удалением из цепи N проводимости g_1, т. е. при предположении, что $g_1=0$ в цепи N.

Удалим из матрицы Q_f ее первый столбец, а из матрицы G_e— ее первые строку и столбец. Пусть результирующие матрицы обозначаются через $Q_f^{(1)}$ и $G_e^{(1)}$. Тогда можно заметить, что

$$Y^{(1)} = Q_f^{(1)}\, G_e^{(1)}\, (Q_f^{(1)})^t = Y - \hat{q}_1 [g_1] (\hat{q}_1)^t. \tag{12.35}$$

Следовательно,

$$y_{ij}^{(1)} = y_{ij} - q_{i1} q_{j1} g_1. \tag{12.36}$$

Заметим, что если матрица реализуема, то все нулевые элементы матрицы Y остаются неизменными в матрице $Y^{(1)}$ и никакой ненулевой элемент y_{ij} не может изменить знак или увеличить свое значение. Следовательно, в таком случае матрица $Y^{(1)}$ будет иметь больше нулевых недиагональных элементов, чем матрица Y.

Предположим, что матрица $Y^{(1)}$ — диагональная. Тогда декомпозиция, описанная в случае 1, даст все оставшиеся столбцы матрицы Q_f и оставшиеся диагональные элементы матрицы G_e.

Если матрица $Y^{(1)}$ — диагональная, то повторяем процедуру, описанную в случае 2, чтобы получить проводимость g_2 и второй столбец $\hat{q}_2$ из матрицы Q_f. Тогда матрица $Y^{(2)}$ задается следующим выражением $Y^{(2)}=Y^{(1)}-\hat{q}_2[g_2](\hat{q}_2)^t$.

Если матрица Y реализуема, то, очевидно, что каждая матрица в последовательности $Y^{(0)}$, $Y^{(1)}$, $Y^{(2)}$, ... подпадает под один из случаев, рассмотренных ранее, и поэтому ее можно декомпозировать. Если на каком-либо шаге недиагональные элементы соответствующей матрицы увеличиваются по величине или меняют знак, то декомпозиция невозможна и поэтому можно закончить алгоритм на этом шаге. В некоторых случаях декомпозиция оказывается возможной, но матрица Q_f может не быть унимодулярной, или матрица G_e может иметь отрицательные диагональные элементы, или то и другое вместе. И в этом случае матрица Y также не является реализуемой.

Проиллюстрируем алгоритм Седербаума с помощью матрицы

$$Y=Y^{(0)}=\begin{bmatrix} 3 & 2 & 1^* & -2 & 0 \\ 2 & 5 & -2 & -2 & 1 \\ 1 & -2 & 3 & 0 & 0 \\ -2 & -2 & 0 & 5 & 3 \\ 0 & 1 & 0 & 3 & 4 \end{bmatrix}.$$

Так как матрица Y — не диагональная, случай 1 отпадает. Поэтому действуем, как описано в случае 2.

Элемент y_{13} является минимальным по значению ненулевым элементом матрицы Y. Устанавливаем $g_1=|y_{13}|=1$, $q_{11}=1$, $q_{31}=1$ (так как y_{13} положителен). Затем определяем оставшиеся элементы первого столбца $\hat{q}_1$ из матрицы Q_f, используя следующие правила:

$$q_{i1}=\begin{cases} 1, \text{ если } y_{if}y_{3f}y_{13}>0 \text{ и } y_{1f}>0; \\ -1, \text{ если } y_{if}y_{3f}y_{13}>0 \text{ и } y_{1f}<0; \\ 0 \text{ в противном случае.} \end{cases}$$

Например, $y_{12}y_{32}y_{13}<0$, поэтому $q_{21}=0$, откуда имеем

$$\hat{q}_1=\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Затем получаем матрицу $Y^{(1)}$ в виде

$$Y^{(1)} = Y^{(0)} - \hat{q}_1[g_1](\hat{q}_1)^t$$

$$= \begin{bmatrix} 2 & 2 & 0 & -2 & 0 \\ 2 & 5 & -2 & -2 & 1^* \\ 0 & -2 & 2 & 0 & 0 \\ -2 & -2 & 0 & 5 & 3 \\ 0 & 1 & 0 & 3 & 4 \end{bmatrix}.$$

В матрице $Y^{(1)}$ элемент (2, 5) имеет минимальное ненулевое значение, поэтому получаем $g_2=1$, $q_{22}=1$, $q_{52}=1$. Затем получаем остальные элементы $\hat{q}_2$, как и раньше, и поэтому

$$\hat{q}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Матрица $Y^{(2)}$ получается в виде

$$Y^{(2)} = Y^{(1)} - \hat{q}_2[g_2](\hat{q}_2)^t$$

$$= \begin{bmatrix} 2 & 2 & 0 & -2^* & 0 \\ 2 & 4 & -2 & -2 & 0 \\ 0 & -2 & 2 & 0 & 0 \\ -2 & -2 & 0 & 5 & 3 \\ 0 & 0 & 0 & 3 & 3 \end{bmatrix}.$$

Продолжая таким же образом, далее получим

$$g_3=2, \quad \hat{q}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \\ 0 \end{bmatrix};$$

$$Y^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -2^* & 0 & 0 \\ 0 & -2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 3 & 3 \end{bmatrix}, \quad g_4=2, \quad \hat{q}_4 = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix};$$

$$Y^{(4)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 3^* \\ 0 & 0 & 0 & 3 & 3 \end{bmatrix} \quad g_5=3, \quad \hat{q}_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

В приведенных выше матрицах элементы, выбранные для рассмотрения на каждом шаге, отмечены звездочкой. Ниже приводится декомпозиция

$$Y^{(0)}=\begin{bmatrix} 3 & 2 & 1 & -2 & 0 \\ 2 & 5 & -2 & -2 & 1 \\ 1 & -2 & 3 & 0 & 0 \\ -2 & -2 & 0 & 5 & 3 \\ 0 & 1 & 0 & 3 & 4 \end{bmatrix}=\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\times\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Хотелось бы отметить, что декомпозиционный алгоритм Седербаума, который мы только что обсудили, применим к любым схемным матрицам: это означает, что если схемную матрицу K можно декомпозировать в виде PDP^t, где P — унимодулярная матрица, а D — диагональная с положительными элементами, то, используя этот алгоритм, можно найти такую декомпозицию. Поэтому ясно, что этот алгоритм можно использовать для декомпозиции матрицы холостого хода резистивной n-полюсной цепи, имеющей цикломатическое число n в виде произведения $B_f R B_f^t$. Таким образом, подход Седербаума для реализации Y-матрицы $(n+1)$-узловой n-полюсной цепи также применим для реализации Z-матрицы n-полюсной цепи с цикломатическим числом n. Следующим шагом подхода Седербаума после получения B_f или Q_f является определение ориентированного графа, соответствующего этим матрицам. Предположим, что определен неориентированный граф, соответствующий B_f или Q_f. Тогда можно легко определить соответствующий ориентированный граф (если он существует) анализом B_f или Q_f, так как такой случай возможен (упражнение 6.1).

Таким образом, возникает необходимость того, чтобы алгоритм реализовывал данную матрицу как цикломатическую матрицу или матрицу сечений неориентированного графа. Такой алгоритм обсуждается в следующем разделе.

12.4. Реализация цикломатической матрицы и матрицы сечений

В этом разделе обсуждается задача построения неориентированного графа, имеющего данную систему циклов и сечений. Не нарушая общности, можно допустить, что нам задаются базисные матрица сечений Q_f или базисная цикломатическая матрица графа, который необходимо построить.

Предположим, что B_f— базисная цикломатическая матрица по отношению к остову T. Тогда ее можно записать в виде $B_f=[U\ F]$, где F и U относятся к остову T и соответствующему ко-остову. Из определения B_f следует, что ненулевые элементы в каждой строке F соответствуют ветвям, которые образуют в остове T путь между концевыми вершинами соответствующей хорды. Другими словами, каждая строка F соответствует пути в остове T. По этой причине F называется *матрицей путей дерева* по отношению к T. При заданной матрице F из выражения (6.13) известно, что $Q_f=[F^tU]$. Таким образом, можно видеть, что реализации B_f и Q_f не являются независимыми задачами. Обе они эквивалентны реализации F как матрицы путей дерева.

Существенной частью процесса реализации F является определение структуры дерева, имеющего ветви, соответствующие таким столбцам F, что путь в дереве, описываемый каждой строкой из F, присутствует в этом дереве. Ясно, что задача построения искомого графа по определенному таким образом дереву обязательно влечет за собой введение хорд между соответствующими вершинами для каждой строки матрицы F. Теперь обсудим алгоритм Рао [12.4], касающийся реализации матрицы F как матрицы путей в дереве неориентированного графа G.

12.4.1. Подготовка

Сначала рассмотрим правила, которые можно использовать для упрощения матрицы F, перед тем как приступить к ее реализации. Если матрица F является блочной, т. е. представимой в виде

$$\begin{bmatrix} F_1 & 0 & \dots & 0 \\ 0 & F_2 & \dots & 0 \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ 0 & 0 & \dots & F_n \end{bmatrix},$$

то каждую из подматриц $F_1, F_2, \dots, F_n$ можно реализовать отдельно и, чтобы получить искомое дерево T, можно соединить произвольно реализованные подграфы (деревья). Если в матрице F имеется много идентичных строк, это означает наличие параллельных хорд в графе G. При реализации соответствующего дерева T все строки, за исключением одной, можно удалить из матрицы F. Строку с единственным ненулевым элементом можно также исключить, поскольку она не накладывает ограничений на взаимосвязь ветвей дерева T.

В случае когда присутствует особая ветвь в единственном из заданных путей, реализации дерева можно достичь после удаления

из матрицы F соответствующего столбца. Ветвь можно добавить позже к одному из концов соответствующего пути. Далее, если некоторые k столбцов матрицы F являются одинаковыми, то достаточно реализовать матрицу, полученную удалением k—1 из этих столбцов. Ветвь, соответствующая оставшемуся столбцу, позже заменяется множеством последовательно соединенных ветвей, упорядоченных произвольным способом. Используя перечисленные правила, сложность данной матрицы F можно снизить последовательным удалением избыточных строк и столбцов. Эта процедура называется *подготовкой* к реализации матрицы F.

12.4.2. Выделение концевых ветвей

Первым основным шагом в алгоритме Рао для реализации матрицы F является идентификация концевой ветви дерева, реализующего матрицу F. Обсудим процедуру, позволяющую идентифицировать такую концевую ветвь. Эта процедура применима к любой матрице общего вида F, и если она допускает различные реализации, то существует по крайней мере одна реализация с выделенной ветвью в качестве концевой. Пусть $W=[w_{ij}]$ — симметричная матрица, определяемая в виде

$$W = F^t F, \tag{12.37}$$

где умножение и сложение выполняется в поле действительных чисел. Ясно, что элемент w_{ij} равен числу путей, задаваемых матрицей F, в которых присутствует как ветвь i, так и ветвь j. Предполагалось, что i-я строка или столбец матрицы W соответствуют i-й ветви дерева T.

Допустим, что G — граф, реализующий данную матрицу F, а T — соответствующий остов. Если рассматривать каждое ребро графа G как проводимость в 1 См и если $Y=[y_{ij}]$ является матрицей короткого замыкания результирующей n-полюсной цепи по отношению к полюсной конфигурации T, то можно видеть, что для $i\neq j$ $w_{ij}=|y_{ij}|$. Отсюда имеем следующие свойства, которые являются очевидными повторениями теорем 12.3—12.5:

Свойство 12.1. Если ветви i, j и k образуют линейное дерево в указанном порядке, когда все другие ветви дерева T короткозамкнуты, то $w_{ij}\geqslant w_{ik}$.

Свойство 12.2. Пусть максимальное значение элементов в i-й строке матрицы W равняется M_i, и пусть этот элемент M_i встречается одновременно в этой строке в позициях столбцов j, k, l, ..., и только в них. Тогда ветви, соответствующие i, j, k, l, . . ., образуют поддерево дерева T.

Свойство 12.3. Пусть наименьший элемент i-й строки матрицы W равен m_i, где m_i, возможно, равно нулю. Пусть в этой строке наименьший элемент встречается одновременно в позициях столбцов

p, q, r, . . ., и только в них. Тогда ветви дерева T, отличные от p, q, r, . . ., образуют поддерево дерева T.

Для дальнейшего обсуждения введем следующие определения:

L-множество является множеством ветвей, которые образуют поддерево дерева T, имеющее только одну вершину, общую со своим дополнением в дереве T, и только одну ветвь из этого множества, называемую ведущей ветвью L-множества, инцидентную этой вершине.

Для иллюстрации приведенного выше определения отметим, что на рис. 12.8 ветви l, k, m и n образуют L-множество с ветвью l в качестве ведущей ветви, тогда как ветви k, m и n не образуют L-мно-

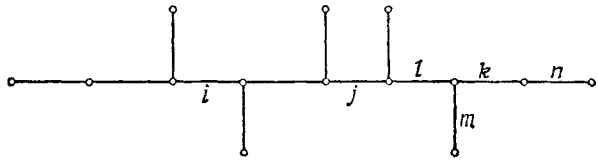


Рис. 12.8. Иллюстрация L-множества.

жества. В простейшем случае в качестве L-множества можно рассматривать концевую ветвь. Очевидно, что дополнением L-множества по отношению к дереву T является поддерево этого дерева T. Для любой ветви r, отличной от концевой ветви дерева T, существуют два L-множества, причем каждое имеет ветвь r в качестве ведущей.

Проверим условия, при которых концевая ветвь поддерева дерева T может также быть концевой ветвью дерева T. Пусть ветви дерева T разделены на два таких множества S и R, что ветви множества T образуют поддерево дерева T. Далее, пусть выполняются соотношения

$$w_{r_i s_i} = w_{r_i s_j} \text{ для всех } r_i \in R \text{ и } s_i,\ s_j \in S. \qquad (12.38)$$

Покажем, что концевая ветвь дерева, которое было бы получено замыканием накоротко всех ветвей множества R, является концевой ветвью дерева T или T', образованного ребрами, соответствующими ветвям дерева T в графе, который 2-изоморфен данному графу. Если равенство

$$w_{rs}=0 \text{ для всех } r \in R \text{ и } s \in S \qquad (12.39)$$

выполняется, то матрица F является блочной матрицей, а подматрицы матрицы F, соответствующие ветвям множеств S и R, можно реализовать раздельно. В таком случае концевая ветвь поддерева, построенного из ветвей множества S, может, очевидно, стать концевой ветвью T в общей реализации.

С другой стороны, пусть существует некоторое $r_i \in R$, для которого $w_{r_i s}$, $s \in S$, не равен нулю. Множество S можно разбить

на непересекающиеся L-множества, поскольку ветви множества R образуют поддерево дерева T. Сначала докажем, что каждое из этих L-множеств является линейным поддеревом дерева T.

Пусть в одном из L-множеств, имеющих s_i в качестве ведущей ветви, существуют такие ветви s_j и s_k, что s_j, s_i и s_k образуют звезду, тогда как другие ветви дерева T являются короткозамкнутыми. Поскольку пути, в которых присутствуют ветви как s_k, так и r_i, отличны от путей, в которых присутствуют ветви s_j и r_i, и поскольку ветвь s_i присутствует в обоих типах путей, получаем

$$w_{r_i s_i} \geqslant w_{r_i s_j} + w_{r_i s_k}. \tag{12.40}$$

Однако в соответствии с формулой (12.38) $w_{r_i s_i} = w_{r_i s_j} = w_{r_i s_k}$. Далее, все они являются ненулевыми. Поэтому неравенство (12.40) удовлетворить нельзя. Отсюда ветви s_i, s_j и s_k должны входить в линейное поддерево дерева T. Рассуждая далее, можно показать, что все ветви в каждом из L-множеств, на которые разбивается множество S, образуют линейное поддерево дерева T.

Таким образом, когда ветви множества R являются короткозамкнутыми, из графа G получается новый граф G_1, который в общем случае является разделимым, и каждая его неразделимая компонента содержит одно или более L-множеств, на которые разбито множество S.

Пусть некоторая неразделимая компонента графа G_1 содержит только одно L-множество, а его ветви образуют линейное поддерево L_1 с ветвями x и y в качестве концевых.

Рассмотрим задание F, определяемое путями, которые включают ветви в поддерево L_1. Некоторые из этих путей могут содержать ветви только поддерева L_1. Остальные должны содержать как ветви из L_1, так и некоторые ветви, не принадлежащие L_1, но каждый из последних путей должен включать все ветви L_1, потому что все рассматриваемые элементы матрицы W равны. Поэтому подграф G_1, состоящий из ветвей поддерева L_1 и хорд, которым соответствуют пути, содержащие только ветви поддерева L_1, можно «перевернуть», чтобы получить граф, 2-изоморфный графу G_1. Если в одном графе концевой ветвью является x, то в другом — y.

Например, для графа G, представленного на рис. 12.9, *а*, соответствующий граф G_1 содержит три неразделимые компоненты, как показано на рис. 12.9, *б*. Заметим, что s_6 является концевой ветвью остова этого графа. Граф G_1', показанный на рис. 12.9, *в*, является 2-изоморфным к графу G_1, а s_4 является концевой ветвью соответствующего дерева G_1'. Таким образом, другая реализация G_1 (рис. 12.9, *г*) матрицы путей дерева F для графа G существует с s_4 в качестве концевой ветви. Используя алгоритм, который будет приведен ниже, можно определить концевую ветвь некоторого дерева, реализующего данную матрицу F. Используя этот алгоритм,

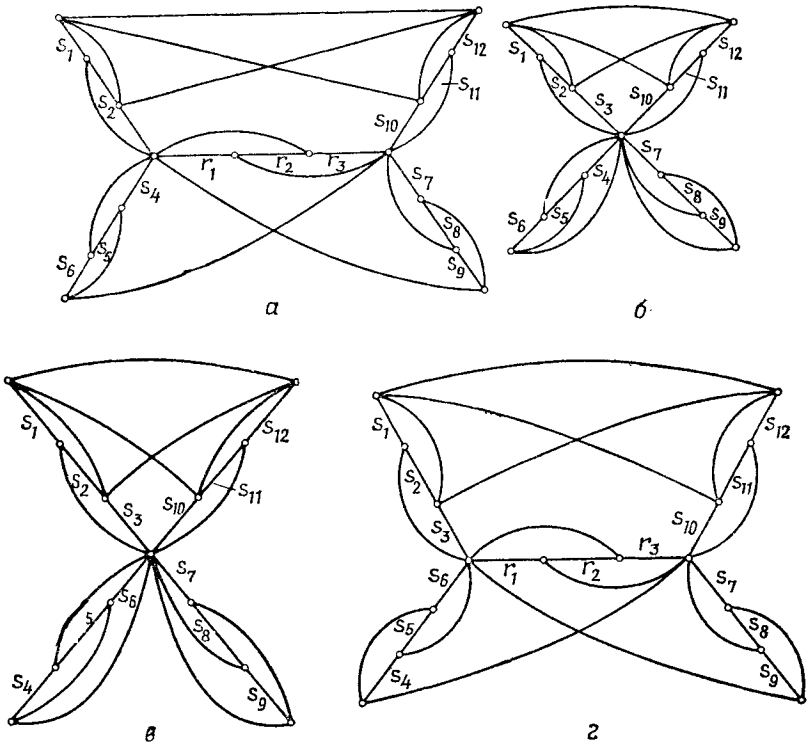


Рис. 12.9. Пример для концевой и дентификации из полученного графа.
а — граф G; *б* — граф G_1; *в* — граф G_1'; *г* — граф G'.

можно выделить любую из ветвей s_1, s_4, s_6, s_7, s_9 и s_{12} как концевую ветвь дерева, реализующего F_1-матрицу. Для каждой из этих возможностей существуют три реализации для матрицы F, имеющие выбранную ветвь в качестве концевой. Результат, который получается из приведенных выше рассуждений, можно сформулировать следующим образом. Когда ветви дерева T можно разделить на два множества R и S, имеющих свойство, определяемое выражением (12.38), то для нахождения концевой ветви T, которая по определению существует в множестве S, можно рассмотреть граф, полученный замыканием накоротко ветвей в множестве R, и найти концевую ветвь дерева, построенного из ветвей множества S. Определенную таким образом концевую ветвь можно считать концевой ветвью дерева искомого графа.

Ниже приведен алгоритм нахождения концевой ветви дерева T, реализующего данную матрицу F.

Шаг 1. Получить матрицу $W^- = F^t F$.

Шаг 2. Рассмотреть произвольную строку i матрицы W.

Пусть m_i (возможно, равный нулю) — минимальный элемент в этой строке. Если существует только один элемент в i-й строке, например элемент w_{ij}, имеющий это минимальное значение, то j — концевая ветвь. В противном случае пусть столбцы, в которых элементы в строке i равны m_i, образуют множество $S=\{s_1, \ldots, s_n\}$. Очевидно, что множество S содержит концевую ветвь дерева T. Пусть подмножество ветвей дерева T, которые не входят в множество S, обозначается как множество R.

Шаг 3. Рассмотреть произвольную строку $r_i \in R$.

Если $w_{r_i s_i} > w_{r_i s_j}$, где $s_i, s_j \in S$, исключаем элемент s_i из множества S. Применяя это правило ко всем подходящим парам элементов в строке r_i, исключим все возможные элементы из множества S и обозначим оставшееся множество через S_1, а дополнительное множество — через R_i.

Шаг 4. Повторить шаг 3 для других строк матрицы W, выбирая каждый раз такую строку r_j, что r_j— элемент множества R, например R_i, применимого на данной стадии. Эта процедура повторяется до тех пор, пока либо

а) множество S_k содержит только один элемент, который в таком случае соответствует концевой ветви, либо

б) множество S_k более не сократимо, т. е. $w_{r_i s_i} = w_{r_i s_j}$ для всех $r_i \in R_k$ и каждой пары элементов s_i и s_j из множества S_k.

Шаг 5. Если множество S_k содержит более одного элемента (случай *б* шага 4), выбрать любой элемент из множества S_k, чтобы заменить им элемент r_i в шаге 3 и далее выполнить 3, 4 и 5, шаги 3—5, т. е. удалить произвольный элемент из множества S_k, добавив его к множеству R_k, и продолжить процесс дальнейшего сокращения множества S_k. Повторное применение этих шагов неизбежно ведет к множеству S_p, содержащему только один элемент, который можно считать концевой ветвью дерева T.

Приведем доказательство корректности этого алгоритма. Сначала отметим, что, согласно свойству 12.3, элементы множества в шаге 2 образуют поддерево дерева T. Поэтому дополнительное множество обязательно содержит по меньшей мере одну концевую ветвь. Заметим далее, что если s_k— элемент множества S, то из свойства 12.1 следует, что все элементы выделенного L-множества, которое не содержит i и для которого s_k является ведущим ребром, присутствуют в множестве S. В шаге 3 получаем сокращенное множество S_1 (удаляя такие элементы, подобные s_k, что $w_{r_i s_k} > w_{r_i s_j}$, $r_i \in R$ и $s_k, s_j \in S$). Необходимо показать, что множество S_1 содержит по крайней мере одну концевую ветвь дерева T. Этот факт очевиден 1) когда s_k и s_j находятся в одном и том же L-множестве, а ветви r_i, s_k и s_j содержатся в линейном поддереве T. С другой стороны, если s_k и s_j принадлежат одному L-множеству, а ветви r_i, s_k и s_j образуют звезду, когда все другие ветви T короткозамкнуты,

это L-множество содержит по крайней мере две концевые ветви. Отсюда в этом случае, даже если s_k является концевой ветвью, сокращенное множество S_1 (полученное удалением такого s_k) содержит концевую ветвь T. Как оказывается, когда удаляется ребро s_k (свойство 12.1), то также следует удалить каждую ветвь, присутствовавшую в дереве, содержащем r_i и s_k в качестве концевых. Таким образом, снова после завершения шага 3 множество ветвей, содержащееся в R, образует поддерево дерева T.

Приведенная выше рекурсивная процедура повторяется в шаге 4, при этом выбирают другую соответствующую строку матрицы W. На каждой стадии сокращенное множество S_i по определению содержит концевую ветвь, и ветви в R_i продолжают образовывать поддерево дерева T. Во многих случаях можно определить концевую ветвь, выполняя эту процедуру (применение шага 4) до тех пор, пока не получится S_k, содержащее только один элемент. Однако, если этого не получается и оказывается невозможным далее сократить число элементов в S_k применением шага 4, это означает, что $w_{r_i s_i}=w_{r_i s_j}$ для всех $r_i \in R_k$ и для всех s_i, $s_j \in S$. Таким образом, на этом этапе элементы R_k образуют поддерево дерева T, а элементы множеств R_k и S_k удовлетворяют формуле (12.38).

Если G_1— граф, полученный из графа G замыканием ветвей R_k, то, как уже упоминалось выше, граф G_1 является разделимым графом, имеющим, например, G_{11}, G_{12}, . . ., G_{1q} в качестве неразделимых компонент. Множество S_k включает ветви остова T_1 графа G.

В шаге 5 элемент из множества S_k выбираем произвольно, например ветвь из G_{1i}, переносим этот элемент в множество R_k и выполняем шаги 3 и 4, используя строку, соответствующую этому элементу. Как можно было видеть, оставшиеся элементы множества S_k в конце этих шагов являются ветвями во всех неразделимых компонентах G_1, кроме G_{1i}. Повторение этих шагов неизбежно приводит к множеству, содержащему ветви, присутствующие только в одной компоненте, и, следовательно, к концевой ветви искомого дерева T.

Приведенные выше рассуждения доказывают корректность алгоритма определения концевой ветви дерева, реализующего данную матрицу F.

12.4.3. Нагруженные пути

Пусть b — концевая ветвь, определенная, как и ранее. Пусть F_1— матрица, полученная удалением из F столбца, соответствующего ветви b. Если дерево T, реализующее F_1, получено, то дерево T для F можно построить добавлением оконечной ветви b к T_1, только если пути в T, содержащие ветвь b, имеют общую концевую. Однако это выполняется не всегда. Например, рассмотрим матрицу F:

$$F=\begin{array}{c} \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{ccccc} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right] \end{array}.$$

Легко определить, что ветвь 1 является концевой. Тогда матрица F_1, полученная удалением столбца 1 из матрицы F, имеет вид

$$F_1=\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Можно показать, что матрица F_1 является реализуемой деревом T_1, представленным на рис. 12.10, *а*. Но это дерево нельзя увеличить, чтобы получить дерево, реализующее матрицу F, поскольку не

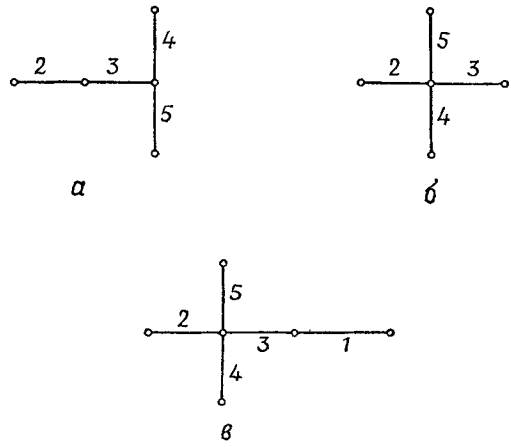


Рис. 12.10.

а — дерево T_1; *б* — дерево T_1'; *в* — дерево T.

будут удовлетворять условию пути, соответствующие хордам 6 и 7. Однако это не означает, что матрица F нереализуема, поскольку одна из других реализаций F_1 может привести к реализации матрицы F. Фактически дерево T_1', показанное на рис. 12.10, *б*, также реализует F_1 и может быть расширено до дерева T (рис. 12.10, *в*), реализующего матрицу F. Чтобы преодолеть выше упомянутую трудность, введем дополнительные пути в T_1, т. е. нагрузим матрицу F_1 такими дополнительными строками, что дерево, реализующее

нагруженную матрицу, по определению окажется дополнением концевой ветви в искомую реализацию дерева T.

Пусть пути, содержащие концевую ветвь, задаваемую матрицей F, обозначаются через $p'_1, p'_2, \ldots, p'_r$. Считается, что все эти пути различны и что каждый содержит более одной ветви. В целях дальнейшего обсуждения определим путь как тривиальный, если существует другой путь, идентичный этому, или когда он содержит только одну ветвь. Таким образом, при данных допущениях среди $p'_1, p'_2, \ldots, p'_r$ нет тривиального пути. Добавлением концевой ветви к каждому из этих путей получаем также пути в T. Пусть этими путями будут $p_1, p_2, \ldots, p_r$. Заметим, что все эти пути имеют общую концевую вершину в T.

Далее, любое дерево, реализующее матрицу F_1, также содержит эти пути. Теперь опишем процедуру такого нагружения F_1, что в каждом дереве, реализующем F_1, пути $p_1, p_2, \ldots, p_r$ имеют общую концевую вершину. Из такого дерева легко построить искомое дерево T добавлением ветви b к общей концевой вершине этих путей.

Рассмотрим три произвольных пути p_i, p_j и p_k из множества $p_1, p_2, \ldots, p_r$. Если $p_i \oplus p_j = p_k$, то не существует реализации, в которой пути p_i, p_j и p_k имели бы общую концевую вершину. Это следует из того факта, что все p_i, p_j и $p_i \oplus p_j$ могут быть путями в дереве только двумя различными способами, показанными на рис. 12.11. Таким образом, матрица F является нереализуемой,

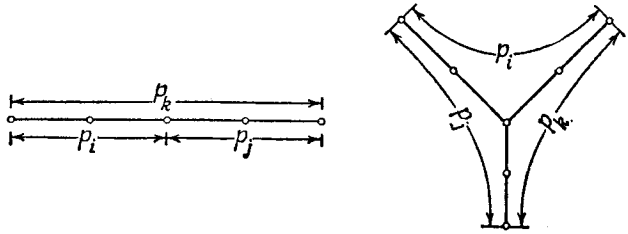


Рис. 12.11.

если сумма по mod 2 любых трех путей в $p_1, p_2, \ldots, p_r$ является пустым множеством. Чтобы проверить наличие такой ситуации, необходимо отдельно просмотреть C_p^3 множеств путей. Однако поскольку нашей целью является реализация F, то проверим только $r-2$ из этих множеств на согласованность и затем дополним матрицу F_1 $2r-3$ дополнительными строками, чтобы убедиться, что нагруженная матрица F_1 оказывается нереализуемой, если F нереализуема. Сначала убедимся, что $p_1 \oplus p_2 \neq p_k$, $k=3, 4, \ldots, r$. Если $p_1 \oplus p_2 = p_k$ для некоторого k, то матрица F является, конечно, нереализуемой. После этой проверки матрица F_1 дополняется строками, соответствующими следующим путям: 1) $p_1 \oplus p_2$, $p_1 \oplus p_3$, ...

. . ., $p_1 \oplus p_r$; 2) $p_2 \oplus p_3$, $p_2 \oplus p_4$, . . ., $p_2 \oplus p_r$. Дополнительные пути, определенные в п. 1, накладывают условие, что каждый из путей p_2, p_3, . . ., p_r, имеет общую с p_1 концевую вершину в дереве, реализующем нагруженную матрицу F_1, так как $p_1 \oplus p_k$ являются путем, только если p_1 и p_k имеют общую концевую вершину. Аналогично условие существования второго множества путей заставляет каждый из путей p_3, p_4, . . ., p_r, иметь также общую концевую вершину с p_2. Поскольку в T_1 не может существовать цикла и предыдущая проверка, гарантируя, что $p_1 \oplus p_2 \neq p_k$, $k=3, 4, \ldots, r$, обеспечивает существование в T_1 путей, подобных показанным на рис. 12.11, то общая концевая вершина p_1 и p_2 является также концевой вершиной каждого пути p_k, $k=3, 4, \ldots, r$. Таким образом, если нагруженная матрица F_1 реализуема, то в любой реализации T_1 этой матрицы пути T_1, p_2, . . ., p_r имеют общую концевую вершину. В таком случае T_1 можно дополнить до дерева, реализующего матрицу F. С другой стороны, если нагруженная матрица F_1 нереализуема, то это означает, что справедлив один из следующих случаев:
1) F_1 является нереализуемой или 2) не существует реализации матрицы F_1, в которой пути p_1, p_2, . . ., p_r имеют общую концевую вершину.

В обоих случаях матрица F является также нереализуемой. Теперь можно сформулировать процедуру для определения дерева T, реализующего данную матрицу F.

12.4.4. Построение дерева

Сначала многократно применим к матрице F операции подготовки до тех пор, пока не получим матрицу F', которая дальше будет несократимой. Когда получено дерево T', реализующее F', реализацию T в виде дерева можно построить на его основе. На следующем шаге определяем концевую ветвь T'. Далее выполняется предварительная проверка путей на согласованность. Столбец, соответствующий выделенной концевой ветви, удаляется из F', а результирующая матрица затем заполняется добавочными строками, как описано ранее. Теперь выполняются операции подготовки над строками этой нагруженной матрицы и результирующая матрица обозначается как F_1'. Выделим теперь концевую ветвь дерева T_1, которое могло бы реализовать F_1', и продолжим аналогично, чтобы найти F_2'. Эта процедура повторяется до тех пор, пока не будет получена матрица F_k', содержащая только два столбца. В случае если бы F' и, следовательно, F были нереализуемыми, этот факт был бы обнаружен на одном из этапов процесса сокращения.

Если матрица F' является реализуемой, то, начиная с дерева T_k, т. е. дерева, реализующего F_k', концевые ветви присоединяются

в обратном порядке относительно той последовательности, в которой они удалялись.

Применение процедуры подготовки гарантирует существование двух путей, содержащих концевую ветвь, и, следовательно, положение этой концевой ветви можно однозначно определить на каждом этапе. Таким образом, можно построить T' и, следовательно, T. Добавляя к последнему дереву хорды, соответствующие строкам матрицы F, получим реализацию графа G.

Можно отметить, что не обязательно сокращать данную матрицу F до матрицы F' применением процедуры подготовки. Процесс сокращения можно применить непосредственно к матрице F, так как пути $p'_1, p'_2, \ldots, p'_r$, рассматриваемые на каждом этапе, различны и каждый из них содержит более одной ветви. Если в процессе построения дерева T из T_k ни на каком этапе не образуются два различных пути, содержащих концевую ветвь, то концевую ветвь можно присоединить к любому концу подходящего пути. Проиллюстрируем двумя примерами материал данного раздела. Пусть необходимо реализовать матрицу

$$F = \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right] \end{array}.$$

Можно видеть, что данную матрицу нельзя сократить применением процедуры подготовки. Тогда получим

$$F'F = \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}\left[\begin{array}{cccc} 2 & 0 & 1 & 2 \\ 0 & 2 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 2 & 1 & 2 & 3 \end{array}\right] \end{array}.$$

Рассматривая первую строку, заключаем, что ветвь 2 является концевой ветвью дерева T. Так как существует только два пути, включающих эту ветвь, нет необходимости проверять согласованность путей, как описано в разд. 12.4.3. Удаляя столбец 2 и дополняя матрицу строкой, соответствующей $p_2 \oplus p_3$, получим

$$F_1\ (\textit{возрастает}) = \begin{array}{c} \begin{array}{ccc} 1 & 3 & 4 \end{array} \\ \left[\begin{array}{ccc} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right] \end{array}.$$

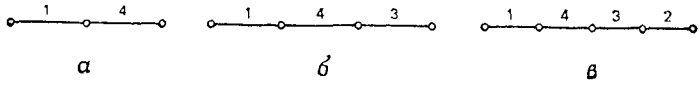
После удаления строк, соответствующих тривиальным путям, имеем

$$F_1' = \begin{array}{c} \begin{matrix} 1 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \end{array},$$

для которой матрица W

$$\begin{array}{c} \begin{matrix} 1 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 3 \\ 4 \end{matrix}\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 3 \end{bmatrix} \end{array}.$$

Анализируя первую строку, заключаем, что ветвь 3 является концевой для T_1. Дерево T_2, реализующее матрицу F_2, которая получается удалением из матрицы F_1' столбца, соответствующего ветви 3, имеет только две ветви и представлено на рис. 12.12, *а*. К этому



а *б* *в*

Рис. 12.12.
а — дерево T_2 *б* — дерево T_1 *в* — дерево T.

дереву добавляем такую ветвь 3, что {1, 3, 4} и {3, 4} являются путями, и получаем T_1, как показано на рис. 12.12, *б*. Далее ветвь 2 добавляется к дереву T_1, так что {2, 3, 4} и {2, 3} являются путями. Таким образом, получается дерево T, показанное на рис. 12.12, *в*. В качестве второго примера рассмотрим следующую матрицу:

$$F = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{array}.$$

Соответствующая матрица W теперь определяется в виде

$$W = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}\begin{bmatrix} 5 & 3 & 3 & 3 \\ 3 & 3 & 1 & 2 \\ 3 & 1 & 3 & 2 \\ 3 & 2 & 2 & 3 \end{bmatrix} \end{array}$$

Строке 1 соответствует множество $S=\{2, 3, 4\}$. Так как это множество S далее не сократимо, то удаляем из него произвольный элемент, например 2. Теперь находим, что $w_{2,4}>w_{2,3}$ и, следовательно, ветвь 3 является концевой. Таким образом, $p_1=\{1\}$, $p_2=\{1, 4\}$ и $p_3=\{1, 2, 4\}$. Далее имеем $p_1 \oplus p_2 \neq p_3$. Теперь удаляем столбец 3 из матрицы F и дополняем ее строками $p_1 \oplus p_2$, $p_1 \oplus p_3$, $p_2 \oplus p_3$. После удаления строк, соответствующих тривиальным путям, получим матрицу F_1':

$$F_1'=\begin{array}{c} \\ 1 \\ 1 \\ 1 \\ 0 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}.$$

Соответствующая матрица W имеет вид

$$\begin{array}{c} \\ 1 \\ 2 \\ 4 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 4 \end{array} \\ \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{bmatrix} \end{array}.$$

Множество S, соответствующее первой строке матрицы W, состоит из элементов 2 и 4. Так как это множество далее не сократимо, то удаляем произвольный элемент, например 2. Теперь элемент 4 является концевой ветвью, так как в множестве S на этом шаге существует только один элемент. Тогда имеем $p_1=\{1\}$, $p_2=\{1, 2\}$, $p_3=\{2\}$. Сумма по mod 2 p_1 и p_2 равна p_3, и, следовательно, матрица F нереализуема.

12.5. Реализация $(n+1)$-узловых резистивных n-полюсных цепей (подход Гуиллемина)

В разд. 12.3 упоминалось, что существуют два основных подхода (один по Седербауму и другой по Гуиллемину) к задаче реализации данной вещественной симметричной матрицы $Y=[y_{ij}]$ как матрицы проводимостей короткого замыкания $(\iota+1)$-узловой резистивной и n-полюсной цепи, не содержащей отрицательных проводимостей. В разд. 12.3 и 12.4 были рассмотрены два главных шага в подходе Седербаума.

Теперь обсудим алгоритм, предложенный Рао [12.4], который использует подход Гуиллемина. Последний, как отмечалось выше, состоит из следующих основных этапов:

1. Получить полюсную конфигурацию дерева T n-полюсной цепи N, реализующей матрицу Y.

2. Получить из дерева T и матрицы Y матрицу проводимостей короткого замыкания Y' n-полюсной цепи, построенной на цепи N, которая имеет полюсную конфигурацию в виде звезды, вершина которой является отрицательно обозначенным зажимом для всех полюсов (разд. 12.2.4).

3. Если матрица Y' является гипердоминантной, то можно легко получить искомую цепь (разд. 12.2.2), в противном случае матрица Y является нереализуемой.

12.5.1. Выделение концевого полюса

Первым важным шагом алгоритма Рао для реализации матриц Y $(n+1)$-узловой резистивной n-полюсной цепи является выделение концевого полюса полюсной конфигурации искомой цепи. Легко видеть, что, применяя к матрице $||y_{ij}||$, алгоритм, описанный в разд. 12.4.2, можно определить такой концевой полюс. В целях полноты приведем далее шаги этого алгоритма.

Шаг 1. Рассмотреть произвольную строку i матрицы Y. Пусть m_i— минимум абсолютных величин — элементов этой строки, и пусть S является множеством столбцов, в котором элемент в строке i равен $\pm m_i$. Если S содержит только один элемент, то этот элемент соответствует концевому полюсу. В противном случае пусть $S=\{s_1, s_2, \ldots, s_k\}$.

Шаг 2. Рассмотреть произвольную другую строку j матрицы Y, где $j \notin S$. Если $|y_{js_p}| \geqslant |y_{js_q}|$ для некоторых s_p и s_q, содержащихся в S, то удалить s_p из S. Удалить все такие s_p для строки j.

Шаг 3. Повторить шаг 2 с некоторой другой строкой k, где k не содержится в сокращенном множестве S. Если повторное применение этого шага не приводит к множеству S, состоящему только из одного элемента, то удалить произвольно любой элемент из множества S на данном этапе и повторить шаги 2 и 3. Окончательно множество S будет содержать только один элемент, который можно идентифицировать как концевой полюс.

12.5.2. Реализация Y-матрицы с ненулевыми элементами

В этом разделе мы рассмотрим реализацию Y-матрицы, которые не содержат нулевых элементов. Общий случай, где допускаются и нулевые элементы, рассматривается в работе [12.4]. Цепь N, реализующая Y-матрицу с ненулевыми элементами, характеризуется тем, что существует хорда, стягивающая каждую пару полюсов цепи N. Если определить путь в T от одной конечной вершины к другой, как *максимальный*, то для каждого такого максимального пути существует хорда в цепи N. Далее, проводимость этой хорды равна величине передаточной проводимости между двумя концевыми

полюсами максимального пути. В дальнейшем каждый максимальный путь будет идентифицироваться и обозначаться последовательностью составляющих его полюсов в порядке, в котором они встречаются, и отмечаться подчеркиванием горизонтальной линией. Если точный порядок всех или некоторых полюсов, содержащихся в пути, неизвестен, то обозначения соответствующих полюсов заключаем в скобки. Так, например, i, (j, k), p, q, (r, s, t) является максимальным путем, содержащим i, j, k, p, q, r, s, и t, в котором точное расположение j и k во второй и третьей позициях, а также относительные позиции r, s и t в последних трех позициях еще должны быть определены.

Пусть полюс i идентифицирован как концевой полюс с использованием процедуры, приведенной выше. Нашей целью является определение множества максимальных путей из концевой вершины дерева T. Очевидно, этих упорядоченных максимальных путей достаточно для установления полюсной конфигурации дерева.

Пусть m_i является минимальной из абсолютных величин элементов в i-й строке матрицы Y, и пусть передаточная проводимость y_{ij}, $j \neq i$, между другими элементами (если они есть) будет равна $\pm m_i$. Полюс j может быть, а может и не быть концевым полюсом. В случае когда он не является концевым, полюса L-множества (разд. 12.4), которое не содержит полюс i и которое имеет j в качестве ведущего полюса, образуют линейное поддерево дерева T и существует только одна хорда в цепи N, которая стягивает i и полюса L-множества. Следовательно, независимо от того, является или нет j концевым полюсом, в дереве T существует только один максимальный путь, который содержит как i, так и j. Пусть y_{ij} положительно (отрицательно) для любого k; если знаки y_{ik} и y_{jk} совпадают (различаются), то k находится на максимальном пути, соединяющем i и j (теорема 12.2). Аналогично могут быть определены все полюса, которые содержатся в этом максимальном пути. Пусть этот путь будет i, (j, k, l, . . .), где порядок полюсов будет определен позже. Очевидно, что ни один из полюсов в скобках в данном представлении не может быть концевым полюсом в другом максимальном пути, содержащем i. С тем чтобы оставить эти полюса вне рассмотрения, в непосредственно следующих шагах элементы в i-й строке матрицы Y в столбцах j, k, l, . . . обводим кружками. Выберем другой элемент y_{ip} ($p \neq i$), имеющий минимальное по абсолютной величине значение среди не отмеченных кружками элементов в i-й строке, и получим другой максимальный путь, т. е. определим все элементы x, для которых y_{ip}, y_{xc}, y_{xp} — положительны. Пусть этим путем является i, (p, q, . . .). Теперь обводим кружками элементы y_{ip}, y_{iq}, ... Процедуру повторяем до тех пор, пока все элементы матрицы Y в i-й строке, за исключением y_{ij}, не будут обведены кружками. Отметим, что каждый из полюсов j, p, . . . на-

ходится только на одном максимальном пути, тогда как другие могут находиться более чем на одном максимальном пути. Данные, необходимые для построения дерева, будут полные, если мы переставим полюса каждого из этих максимальных путей в том порядке, в котором они находятся в них.

Для этого упорядочим полюса в каждом максимальном пути в убывающем порядке их передаточных проводимостей с полюсом i. Когда более чем один полюс имеет одно и то же значение передаточной проводимости по отношению к полюсу i, заключаем эти полюса в скобки. Таким образом, когда первая фаза закончена, имеем множество частично-упорядоченных последовательностей вида $i, m, (h, k, n, o), s, (t, u, v), (w, x, j)$.

Теперь необходимо упорядочить полюса внутри каждой пары скобок. Для этого рассмотрим строку, соответствующую полюсу m, непосредственно предшествующему скобкам, содержащим h, k, n и o, и расположим полюса в скобках в убывающем порядке по величине их передаточных проводимостей относительно полюса m. Если найдены некоторые передаточные проводимости с равными величинами, заключим соответствующие полюсы в скобки и повторим процедуру. Упорядочение полюсов можно также выполнить, рассматривая полюс s, непосредственно следующий за скобками, и располагая полюсы в скобках в возрастающем порядке величин их передаточных проводимостей относительно полюса s. Таким образом, можно определить последовательность полюсов в каждом максимальном пути.

Однако в некоторых случаях возможна ситуация, когда передаточные проводимости всех полюсов в скобках по отношению к любому внешнему полюсу имеют равное значение. Например, для последовательности вида $\ldots, s, (p, q, r), l, \ldots$ мы можем иметь $|y_{sp}| = |y_{sp}| = |y_{sq}| = |y_{sr}|$ и $|y_{pt}| = |y_{qt}| = |y_{rt}|$.

В таком случае выделяем квадратную подматрицу матрицы Y, соответствующую полюсам в скобках, и определяем ее полюсную структуру в виде линейного дерева отдельно. Пусть q, r, p занимают определенный порядок. Тогда либо $\ldots, s, q, r, p, t, \ldots$, либо $s, p, r, q, t, \ldots$ можно взять в качестве упорядоченной последовательности, а два соответствующих графа являются 2-изоморфными. После того как закончено упорядочение полюсов в максимальном пути, упорядочение общих ветвей в других максимальных путях является таким же, как и раньше для реализуемых матриц. Это дает преимущество при упорядочении ветвей в новом пути.

Проиллюстрируем приведенную выше процедуру определением конфигурации полюсного дерева, заданного матрицей проводимостей Y, приведенной ниже:

$$Y=\begin{array}{c|cccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline 1 & 67 & 20 & 10 & 15 & 45 & 17 & 23 & 33 & 15 & 26 \\ 2 & 20 & 30 & 3 & -6 & 23 & 5 & 12 & 8 & -3 & 29 \\ 3 & 10 & 3 & 30 & 2 & 5 & -4 & -1 & 11 & 5 & 3 \\ 4 & 15 & -6 & 2 & 45 & 30 & 9 & 4 & 11 & -15 & -12 \\ 5 & 45 & 23 & 5 & 30 & 79 & 14 & 16 & 25 & -18 & 29 \\ 6 & 17 & 5 & -4 & 9 & 14 & 30 & -7 & 24 & 3 & 5 \\ 7 & 23 & 12 & -1 & 4 & 16 & -7 & 49 & -8 & 7 & 12 \\ 8 & 33 & 8 & 11 & 11 & 25 & 24 & -8 & 42 & 8 & 14 \\ 9 & 15 & -3 & 5 & -15 & -18 & 3 & 7 & 8 & 36 & -3 \\ 10 & 26 & 29 & 3 & -12 & 29 & 5 & 12 & 14 & -3 & 41 \end{array}$$

1. Рассматривая первую строку, определяем, что полюс 3 является концевым.

2. В строке 3 минимальным по величине является элемент в 7 столбце, причем y_{37}— отрицательно. Просматривая столбцы 3 и 7 замечаем, что для $k=3$, 7 и 8 только y_{k3} и y_{k7} имеют противоположные знаки.

Таким образом 3, (7, 8) является максимальным путем. Отметим кружками элементы в столбцах 7 и 8 третьей строки, которая выглядит теперь следующим образом:

$$\left[10 \quad 3 \quad 30 \quad 2 \quad 5 \quad -4 \quad \textcircled{-1} \quad \textcircled{11} \quad 5 \quad 3\right].$$

3. Среди неотмеченных элементов в строке 3 элемент y_{34} имеет минимальное значение и положителен. Просматривая столбцы 3 и 4, замечаем, что для $k=1$, 3, 4, 5 и 8 элементы y_{k3} и y_{k4} имеют одинаковый знак, а 3, (1, 4, 5, 8) является другим максимальным путем. Обводим теперь элементы y_{31}, y_{34} и y_{35}. Аналогично получаем другие максимальные пути 3, (2, 1, 5, 8, 10); (3, 6); и 3, (9, 1, 8).

4. Рассмотрим путь 3, (2, 1, 5, 8, 10). Располагая полюса в скобках в убывающем порядке значений их передаточных проводимостей по отношению к полюсу 3, получим 3, 8, 1, 5, (2, 10). Рассматривая 5-ю строку матрицы Y, находим, что $|y_{5,10}|>|y_{5,2}|$. Таким образом, 3, 8, 1, 5, 10, 2 является упорядоченным максимальным путем.

5. Поскольку другие пути имеют общие ветви с этим путем, то их также легко упорядочить. Другими упорядоченными максимальными путями являются 3, 8, 7; 3, 8, 1, 5, 4; и 3, 8, 1, 9.

6. Теперь легко определить полюсное дерево T (рис. 12.13, *а*).

7. Ориентацию любого из полюсов, например полюса 1, можно выбрать произвольно. Ориентации остальных полюсов получают,

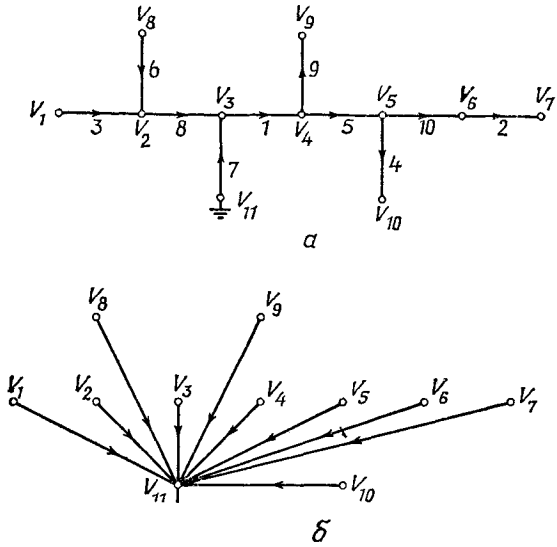


Рис. 12.13.
а — дерево T; *б* — дерево T^*.

используя теорему 12.1. Результирующая ориентация полюсов показана на рис. 12.13, *а*.

8. Пусть звезда T^*, имеющая то же множество вершин, что и дерево T, выбрана такой, как показано на рис. 12.13, *б*.

Матрица K, связывающая векторы полюсных напряжений V_p и V_p^*, приведена ниже. (Отметим, что $V_p = KV_p^*$.)

$$K = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Матрица проводимостей короткого замыкания Y^* по отношению к T^* определяется выражением

$$Y^* = K^t Y K$$

$$= \begin{bmatrix} 30 & -15 & 0 & 0 & 0 & 0 & -3 & -4 & -5 & -2 \\ -15 & 24 & -1 & 0 & 0 & -6 & 0 & -2 & 0 & 0 \\ 0 & -1 & 30 & -7 & -4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -7 & 26 & -16 & 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & -4 & -16 & 23 & 0 & 0 & 0 & 0 & -3 \\ 0 & -6 & 0 & 0 & 0 & 13 & -1 & 0 & 0 & -6 \\ -3 & 0 & 0 & 0 & 0 & -1 & 30 & -5 & -3 & -6 \\ -4 & -2 & 0 & 0 & 0 & 0 & -5 & 30 & -3 & -9 \\ -5 & 0 & 0 & -3 & 0 & 0 & -3 & -3 & 36 & -15 \\ -2 & 0 & 0 & 0 & -3 & -6 & -6 & -9 & -15 & 45 \end{bmatrix}.$$

Матрица Y^* является гипердоминантной. Проводимости 10-полюсной цепи, реализующей матрицу Y, можно вычислить из Y^*, и они определяются следующими соотношениями:

$$g_{ij} = -y_{ij}, \quad 1 \leqslant i, \ j \leqslant 10,$$

$$g_{i,\,11} = y_{ii} - \sum_{\substack{j=1 \\ j \neq i}}^{10} |y_{ij}|,$$

$$1 \leqslant i \leqslant 10.$$

12.6. Замечания, касающиеся литературы

Исследования в области резистивных n-полюсных цепей в основном касались следующих тем:

1. Y-матрицы n-полюсных цепей с числом узлов $(n+1)$, т. е. n-полюсные цепи ранга n.
2. Y-матрицы n-полюсных цепей с числом полюсов, большим $n+1$.
3. Z-матрицы n-полюсных цепей с цикломатическим числом n.

Декомпозиционный алгоритм Седербаума, приведенный в разд. 12.3, и алгоритм Рао, приведенный в разд. 12.4, при реализации матрицы сечений или цикломатической матрицы полностью решают задачу построения n-полюсной цепи, имеющей $n+1$ узлов и заданную Y-матрицу. Рао [12.4] также построил другой алгоритм для решения данной задачи. Этот алгоритм, как упоминалось в разд. 12.5, основан на подходе Гуиллемина, который включает построение полюсной конфигурации по данной Y-матрице. Если необходим алгоритм, который реализует Y-матрицу, не имеющую ненулевых элементов, то следует посмотреть работу [12.5]. Некоторые из ранних сообщений по $(n+1)$-узловым резистивным n-полюсным цепям можно найти в работах [12.6, 12.7].

Гуиллемин [12.8] дал общий подход к исследованию задачи реализации Y-матрицы n-полюсной цепью, имеющей более $n+1$ узлов. Используя этот подход, в разное время было сделано много попыток

раскрыть свойства Y-матриц для таких цепей. Значительных результатов было немного, за исключением специального случая $(n+2)$-узловых цепей. Можно также посмотреть работы [12.9—12.11].

Как показано в разд. 12.3, алгоритм Седербаума можно применить к декомпозиции Z-матрицы n-полюсной цепи с цикломатическим числом n. Этот алгоритм и алгоритм из разд. 12.4 полностью решают задачу реализации резистивной матрицы холостого хода n-полюсной цепи с цикломатическим числом n. Альтернативные подходы к решению этой задачи обсуждаются в работах [12.12, 12.13]. Можно обратиться также к работам [12.14, 12.15], в которых обсуждаются взаимоотношения между резистивными цепями и сетями связи.

Седербаум [12.16] ввел понятия «модифицированная матрица сечений» и «модифицированная цикломатическая матрица». Некоторые приложения этих матриц к исследованию n-полюсной задачи рассматриваются в работах [12.17, 12.18]. Обсуждались также многие другие интересные вопросы, касающиеся задачи о n-полюсной резистивной цепи, например в работах [12.19—12.21]. Задача реализации матрицы сечений и цикломатической матрицы рассматривается в работах [12.22—12.25].

В работе [12.26] обсуждаются некоторые результаты в области резистивных n-полюсных цепей, а также применение задачи реализации цикломатической матрицы при синтезе переключательных схем. Понятия из области резистивных n-полюсных цепей можно успешно использовать для реализации матриц состояния RLC-цепей [12.27].

Упражнения

12.1. Рассмотрим n-полюсную цепь N. Допустим, что не существует контура, состоящего только из полюсов. Покажите, что если I_p является вектором полюсных токов, а I — вектором токов, связанных с неполюсными элементами цепи N, то существует такая матрица M, состоящая из элементов -1, $+1$ или 0, что $I_p = MI$.

Примечание. Используйте теорему 10.31 и [12.28].

12.2. Выполните упражнение 12.1, заменив контуры на сечения: I_p на V_p и I на V, где V_p и V — векторы, связанные с полюсными и неполюсными элементами цепи N соответственно.

12.3. Используя подход Седербаума, получите резистивную 6-полюсную цепь, имеющую следующую матрицу проводимостей короткого замыкания:

$$\begin{bmatrix} 3 & 2 & 0 & 1 & -1 & 2 \\ 2 & 5 & -2 & 0 & 2 & 2 \\ 0 & -2 & 9 & 0 & -3 & 0 \\ 1 & 0 & 0 & 9 & -1 & -1 \\ -1 & 2 & -3 & -1 & 8 & 0 \\ 2 & 2 & 0 & -1 & 0 & 8 \end{bmatrix}.$$

12.4. Реализуйте следующую матрицу как матрицу путей дерева:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

12.5. Пусть $Z=[z_{ij}]$ является матрицей сопротивлений холостого хода n-полюсной цепи N с цикломатическим числом n. Покажите, что если полюса цепи N образуют остов в виде звезды нагруженной цепи $\hat{N}$, то Z удовлетворяет следующему соотношению:
для каждых трех индексов i, j и k равны все три элемента z_{ij}, z_{jk} и z_{kj} или равны два меньших элемента [2.14].

12.6. Пусть матрицы Q_1, Q_2, Y_{12}, Y_{22} и т. д., связанные с n-полюсной цепью N, определены, как в разд. 12.1. Матрица $Q_1-Y_{12}Y_{22}^{-1}Q_2$ называется *модифицированной матрицей сечений* цепи N. Докажите следующее:

а) $Y=Q_mY_eQ_1^t$, где Q_m является модифицированной матрицей сечений цепи N.

б) Матрица Q_m является модифицированной матрицей сечений n-полюсной цепи N тогда и только тогда, когда $Q_mY_eQ_2^t=0$.

12.7. Пусть N является параллельной комбинацией двух n-полюсных цепей N_1 и N_2, имеющих Y_1 и Y_2 в качестве матриц проводимостей короткого замыкания. Покажите, что матрица проводимости короткого замыкания цепи N равна Y_1+Y_2 тогда и только тогда, когда цепи N_1 и N_2 имеют одинаковую матрицу сечений [12.18].

12.8. Используя результаты упражнений 12.6 и 12.7, постройте 4-полюсную цепь с матрицей проводимости короткого замыкания, равной матрице проводимостей короткого замыкания 4-полюсной цепи, представленной на рис. 12.14, в котором номиналы всех проводимостей приведены в сименсах.

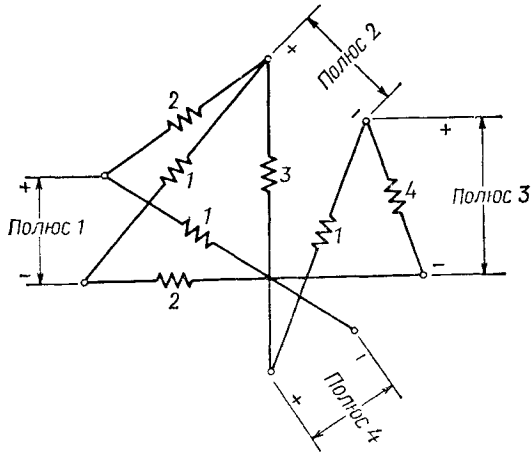


Рис. 12.14.

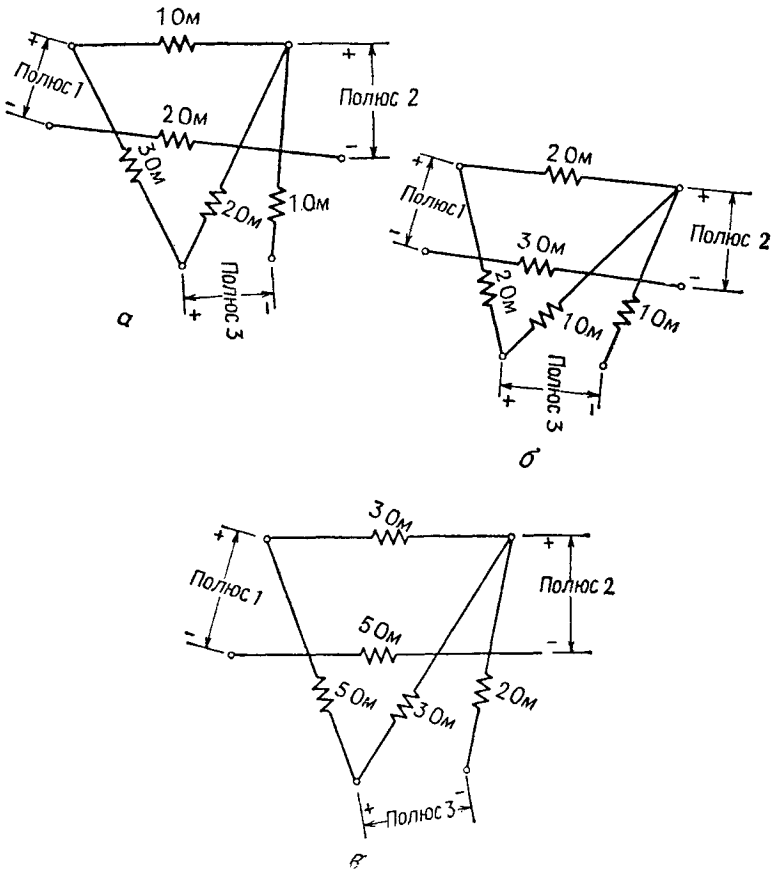


Рис. 12.15.

а — цепь N_1; *б* — цепь N_2; *в* — псевдопоследовательное соединение цепей N_1 и N_2.

12.9. Пусть матрицы B_1, B_2, Z_{12}, Z_{22} и т. д., связанные с n-полюсной цепью N, определены в соответствии с разд. 12.1. Тогда матрица $(B_1 Z_{12} Z_{22}^{-1} B_2)$ называется *модифицированной цикломатической матрицей* цепи N.

Докажите следующее:

а) $Z = B_m Z_e B_1^t$, где B_m — модифицированная цикломатическая матрица цепи N.

б) Матрица B_m является модифицированной цикломатической матрицей n-полюсной цепи тогда и только тогда, когда $B_m Z_e B_2^t = 0$.

12.10. Рассмотрите две n-полюсные цепи N_1 и N_2, имеющие идентичные полюсные и реберные конфигурации, а также ориентации. Пусть Z_1 и Z_2 — матрицы полных сопротивлений холостого хода цепей N_1 и N_2. Постройте третью n-полюсную цепь N_3, имеющую такие же реберные и полюсные конфигурации, а также ориентации, как и цепи N_1 и N_2, но имеющую полное сопротивление каждого ребра, равное сумме полных сопротивлений соответствующих ребер цепей N_1 и N_2. Цепь N_3 будем называть *псевдопоследовательным соединением* цепей N_1 и N_2 (две цепи N_1 и

N_2 и их псевдопоследовательное соединение N_3 представлены на рис. 12.15). Покажите, что матрица полных сопротивлений холостого хода цепи N_3 равна Z_1+Z_2 тогда и только тогда, когда цепи N_1 и N_2 имеют идентичные модифицированные цикломатические матрицы [12.29].

12.11. Используя результаты упражнения 12.9, *б*, постройте 4-полюсную цепь, модифицированная цикломатическая матрица которой совпадает с соответствующей матрицей цепи, представленной на рис. 12.14.

12.12. Для данной *n*-полюсной цепи N, используя результаты упражнений 12.9 и 12.10, попытайтесь построить новую 4-полюсную цепь, матрица полных сопротивлений короткого замыкания которой совпадает с соответствующей матрицей, представленной на рис. 12.14. Обсудите все трудности, которые могут встретиться.

13. Функции цепи и чувствительность цепи

В этой главе сначала выведем формулы для функций цепей в терминах проводимостей, связанных с некоторыми подграфами данной цепи. Такие формулы, называемые топологическими формулами, впервые были выведены Кирхгофом с использованием сопротивлений и позднее, в 1892 г., Максвеллом с использованием проводимостей. Исследование в этой главе основывается на матрице неопределенных проводимостей. Многие результаты, которые будут здесь представлены, просто следуют из результатов гл. 6.

В заключительной части главы описывается метод вычисления чувствительности функций цепи. Этот метод основывается на понятии «сопряженная цепь» и теореме Теллежена 11.3.

12.1. Топологические формулы для *RLC*-цепей без взаимных индуктивностей

В этом разделе выводятся топологические формулы для *RLC*-цепей без взаимных индуктивностей. Матрица узловых проводимостей является отправной точкой для вывода этих формул.

Сначала рассмотрим 1-полюсную *RLC*-цепь N без взаимных индуктивностей. Пусть цепь N имеет $n+1$ узлов, обозначаемых 0, 1, 2, . . ., n, и пусть узлы 1 и 0 являются соответственно положительно и отрицательно обозначенными зажимами полюса (рис. 13.1). Допустим наличие нулевых начальных условий (емкостных напряжений и индуктивных токов) в цепи N. Далее, все переменные токов и напряжений являются преобразованиями Лапласа от комплексной частотной переменной. Возбудим цепь, подсоединяя к полюсу источник тока величиной I_1. Если V_1, V_2, . . ., V_n — напряжения в узлах 1, 2, . . ., n относительно узла 0, а Y — матрица узловых проводимостей цепи N с отмеченным нулевым

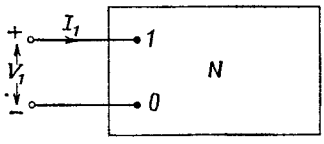


Рис. 13.1.1-полюсная цепь.

узлом, то узловыми уравнениями для N будут

$$YV=I, \text{ где} \tag{13.1}$$

$$V=\begin{bmatrix} V_1 \\ V_2 \\ \cdot \\ \cdot \\ \cdot \\ V_n \end{bmatrix} \quad \text{и} \quad I=\begin{bmatrix} I_1 \\ I_2 \\ \cdot \\ \cdot \\ \cdot \\ I_n \end{bmatrix}.$$

Решая уравнение (13.1) относительно V_i, получим $V_i=(\Delta_{ii}/\Delta)I_i$, где $\Delta=\det Y$ и $\Delta_{ii}=(1,\ 1)$ — алгебраическое дополнение Y. Поэтому полное сопротивление z цепи N в точке возбуждения задается в виде

$$z=V_i/I_i=\Delta_{ii}/\Delta, \tag{13.2}$$

а проводимость в точке возбуждения — в виде

$$y=1/z=\Delta/\Delta_{ii}. \tag{13.3}$$

Чтобы вывести топологические формулы для z и y, необходимо выразить Δ_{ii} и Δ через соответствующие величины, связанные с некоторыми подграфами цепи N. Как показано ниже, это нетрудно сделать. В дальнейшем *произведением проводимостей* подграфа цепи N будем называть произведение проводимостей, связанных с ребрами подграфа. Если подграф не имеет ребер, то произведение его проводимостей принимается равным 1. Аналогично определяется *произведение полных сопротивлений* подграфа цепи N.
В соответствии с этими определениями примем следующие обозначения:

W — сумма произведений проводимостей остовов цепи N;
W_{ij} — сумма произведений проводимостей всех остовных 2-деревьев цепи N.
$C(W)$ — сумма произведений полных сопротивлений всех коостовов цепи N.
$C(W_{ij})$ — сумма произведений полных сопротивлений дополнений всех остовных 2-деревьев T_{ij} цепи N. (13.4)

Если $z_1,\ z_2,\ \ldots,\ z_m$ — полные сопротивления ребер цепи N, то, очевидно,

$$C(W)=W\prod_{i=1}^{m} z_i, \tag{13.5}$$

$$C(W_{i,j})=W_{i,j}\prod_{i=1}^{m} z_i. \tag{13.6}$$

Рассмотрим теперь матрицу проводимостей узлов цепи N. Если обозначить символом N взвешенный граф, причем веса представляют проводимости соответствующих ребер, то можно видеть (упражнение 6.16), что

$$\Delta = \det Y = W, \tag{13.7}$$

$$\Delta_{ij} = W_{ij,\,0}. \tag{13.8}$$

Используя выражения (13.5) — (13.8) в формулах (13.2) и (13.3), получим следующую теорему

Теорема 13.1. Пусть z и y — полное сопротивление и проводимость соответственно в точке возбуждения однополюсной RLC-цепи без взаимных индуктивностей. Если 1 и 0 являются зажимами полюса цепи, то

1) $z = W_{1,0}/W = C(W_{1,0})/C(W)$,
2) $y = W/W_{1,0} = C(W)/C(W_{1,0})$.

Выведем топологические формулы для функций полного сопротивления холостого хода и проводимости короткого замыкания 2-полюсной RLC-цепи N (рис. 13.2) без взаимных индуктивностей. Снова предполагаем в цепи N наличие нулевых начальных условий. Если полюса цепи N возбуждаются источниками тока величиной I_1 и I_2, то узловые уравнения цепи N можно записать в виде $YV = I$, где $I = [I_1,\ I_2\ -I_2\ 0\ \ldots\ 0]$.

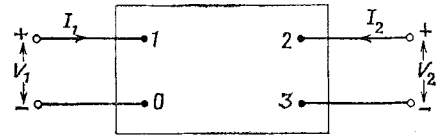


Рис. 13.2. Простая 2-полюсная RLC-цепь.

Решая их относительно узловых напряжений V_1, V_2 и V_3, получим

$$V_1 = \frac{1}{\Delta}(\Delta_{11}I_1 + \Delta_{21}I_2 - \Delta_{31}I_2),$$

$$V_2 = \frac{1}{\Delta}(\Delta_{12}I_1 + \Delta_{22}I_2 - \Delta_{32}I_2),$$

$$V_3 = \frac{1}{\Delta}(\Delta_{13}I_1 + \Delta_{23}I_2 - \Delta_{33}I_2).$$

Из приведенных выше соотношений получаем

$$\begin{bmatrix} V_1 \\ V_2 - V_3 \end{bmatrix} = \frac{1}{\Delta}\begin{bmatrix} \Delta_{11} & \Delta_{21} - \Delta_{31} \\ \Delta_{12} - \Delta_{13} & \Delta_{22} + \Delta_{33} - \Delta_{32} - \Delta_{23} \end{bmatrix}\begin{bmatrix} I_1 \\ I_2 \end{bmatrix}. \tag{13.9}$$

Отметим, что, поскольку матрица Y — симметричная, имеем $\Delta_{ij} = \Delta_{ji}$. Так как V_1 — напряжение на полюсе 1, а $V_2 - V_3$ — напряжение на полюсе 2, матрица коэффициентов в выражении (13.9) равна матрице полных сопротивлений холостого хода Z_{oc} цепи N. Таким образом,

$$Z_{oc} = \frac{1}{\Delta}\begin{bmatrix} \Delta_{11} & \Delta_{12} - \Delta_{13} \\ \Delta_{12} - \Delta_{13} & \Delta_{22} + \Delta_{33} - 2\Delta_{23} \end{bmatrix}. \tag{13.10}$$

Чтобы выразить элементы Z_{oc} через соответствующие произведения проводимостей цепи N, сначала заметим, что

$$\Delta_{11}=W_{1,\,0}. \tag{13.11}$$

Из формулы (13.8) также имеем

$$\Delta_{12}-\Delta_{13}=W_{12,\,0}-W_{13,\,0}. \tag{13.12}$$

Поскольку каждое остовное 2-дерево $T_{12,\,0}$ является или остовным 2-деревом $T_{12,\,30}$, или остовным 2-деревом $T_{123,\,0}$, получаем

$$W_{12,\,0}=W_{12,\,30}+W_{123,\,0}. \tag{13.13}$$

Аналогично

$$W_{13,\,0}=W_{13,\,20}+W_{123,\,0}. \tag{13.14}$$

Используя выражения (13.13) и (13.14) в формуле (13.12), получим

$$\Delta_{12}-\Delta_{13}=W_{12,\,30}-W_{13,\,20}. \tag{13.15}$$

Рассуждая аналогично, получаем

$$\begin{aligned}\Delta_{22}+\Delta_{33}-2\Delta_{23}&=W_{2,\,0}+W_{3,\,0}-2W_{23,\,0}=W_{23,\,0}+W_{2,\,30}+W_{23,0}+\\&+W_{3,\,20}-2W_{23,\,0}=W_{2,\,30}+W_{3,\,20}=W_{2,\,3}.\end{aligned} \tag{13.16}$$

Используя выражения (13.15) и (13.16) в формуле (13.10), получим следующую теорему:

Теорема 13.2. Пусть N — 2-полюсная RLC-цепь без взаимных индуктивностей. Пусть положительные и отрицательные зажимы полюсов цепи N будут такими, как показано на рис. 13.2. Тогда матрица полных сопротивлений холостого хода Z_{oc} цепи N определяется выражением

$$Z_{oc}=\frac{1}{W}\begin{bmatrix} W_{1,\,0} & W_{12,\,30}-W_{13,\,20} \\ W_{12,\,30}-W_{13,\,20} & W_{2,\,3}\end{bmatrix}.$$

Из выражений (13.5) и (13.6) ясно, что элементы Z_{oc} можно также выразить с использованием произведений полных сопротивлений соответствующих коостовов и дополнений остовных 2-деревьев.

Матрицу проводимостей короткого замыкания Y_{sc} можно получить, инвертируя матрицу холостого хода Z_{oc}. Таким образом,

$$Y_{sc}=\frac{1}{\Delta_{11,\,22}+\Delta_{11,\,23}-2\Delta_{11,\,23}}\begin{bmatrix}\Delta_{22}+\Delta_{33}-2\Delta_{23} & \Delta_{12}-\Delta_{12} \\ \Delta_{13}-\Delta_{12} & \Delta_{11}\end{bmatrix}, \tag{13.17}$$

Здесь мы использовали тождество

$$\Delta_{ii,\,jk}=\frac{1}{\Delta}(\Delta_{ii}\Delta_{jk}-\Delta_{ji}\Delta_{ik}). \tag{13.18}$$

Заметим, что $\Delta_{ii,\,jk}$ — алгебраическое дополнение 2-го порядка Y по отношению к его $(i,\,i)$- и $(j,\,k)$-элементам и задается выражением

$$\Delta_{ii,\,jk} = \operatorname{sgn}(i-j)\operatorname{sgn}(i-k)(-1)^{j+k}\det(Y_{ii,\,jk}), \qquad (13.19)$$

где 1) $\operatorname{sgn}(r-s) = \begin{cases} 1, & \text{если } (r-s)>0, \\ -1, & \text{если } (r-s)<0; \end{cases}$

2) $Y_{ii,\,jk}$ — матрица, полученная удалением из матрицы Y строк i и j и столбцов i и k.

Фактически $\Delta_{ii,\,jk}$ — алгебраическое дополнение матрицы Y_{ii} относительно $(j,\,k)$-элемента матрицы Y, а выражение (13.18) — тождество Якоби [13.1]. Выразим $\Delta_{ii,\,jk}$ через соответствующие произведения проводимостей.

Рассмотрим цепь N', которая получается из цепи N замыканием узлов i и 0. Тогда матрица проводимостей узлов Y' цепи N' с отмеченным узлом 0 равна Y_{ii}. Если Δ'_{jk} — алгебраическое дополнение Y' относительно $(j,\,k)$-элемента из матрицы Y, то, как уже отмечалось, $\Delta'_{jk}=\Delta_{ii,\,jk}$.

Если $T'_{jk,\,0}$ — остовное 2-дерево цепи N' с узлами j и k в одной компоненте и узлом 0 в другой компоненте, а $W'_{jk,\,0}$ — сумма произведений проводимостей всех остовных 2-деревьев $T'_{jk,\,0}$ цепи N', то из выражения (13.8) получим $\Delta'_{jk}=W'_{jk,\,0}$. Но остовное 2-дерево $T'_{jk,\,0}$ цепи N' является остовным 3-деревом цепи N типа $T_{i,\,jk,\,0}$, потому что в цепи N' узлы 0 и i представлены единственным узлом. Таким образом,

$$\Delta_{ii,\,jk} = \Delta'_{jk} = W'_{jk,\,0} = W_{i,\,jk,\,0}, \qquad (13.20)$$

где $W_{i,\,jk,\,0}$ — сумма произведений проводимостей всех остовных 3-деревьев типа $T_{i,\,jk,\,0}$.

Рассмотрим член $\Delta_{11,\,22}+\Delta_{11,\,33}-2\Delta_{11,\,23}$ в выражении (13.7). Из выражения (13.20) ясно, что $\Delta_{11,\,22}+\Delta_{11,\,33}-2\Delta_{11,\,23}=W_{1,\,2,\,0}+W_{1,\,3,\,0}-2W_{1,\,2,\,3,\,0}$. Поскольку $W_{1,\,2,\,0}=W_{13,\,2,\,0}+W_{1,\,23,\,0}+W_{1,\,2,\,30}$ и $W_{1,\,3,\,0}=W_{12,\,3,\,0}+W_{1,\,23,\,0}+W_{1,\,3,\,20}$, получим

$$\Delta_{11,\,22}+\Delta_{11,\,33}-2\Delta_{11,\,23}=W_{13,\,2,\,0}+W_{1,\,2,\,30}+W_{12,\,3,\,0}+W_{1,\,3,\,20}. \qquad (13.21)$$

Используя выражения (13.8), (13.15), (13.16) и (13.21), получим топологические формулы для всех элементов Y_{sc}.

Теорема 13.3. Пусть N — 2-полюсная RLC-цепь без взаимных индуктивностей. Пусть положительные и отрицательные зажимы полюсов цепи N будут такими, как показано на рис. 13.2. Тогда матрица проводимостей короткого замыкания Y_{sc} цепи N задается выражением

$$Y_{sc} = \frac{1}{W_{13,\,2,\,0}+W_{1,\,2,\,30}+W_{12,\,3,\,0}+W_{1,\,3,\,20}} \begin{bmatrix} W_{2,\,3} & W_{13,20}-W_{12,30} \\ W_{13,\,20}-W_{12,\,30} & W_{1,\,0} \end{bmatrix}.$$

Проиллюстрируем топологические оценки матриц полных сопротивлений холостого хода и матриц проводимостей короткого замыкания 2-полюсной *RLC*-цепи. Следующий пример взят из работы [13.2].

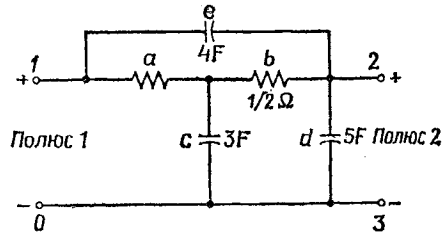


Рис. 13.3.

Рассмотрим цепь, представленную на рис. 13.3. Элементы этой цепи обозначены символами a, b, c, d и e. Номиналы ее элементов показаны на рисунке. Отметим, что 3 и 0 обозначают одну и ту же вершину. Поэтому цепь N не содержит остовных 2-деревьев типа $T_{13,\,20}$. По этой же причине она не содержит остовных 3-деревьев типа $T_{13,\,2,\,0}$, $T_{12,\,3,\,0}$ и $T_{1,\,3,\,20}$.

Остовы, остовные 2-деревья и 3-деревья, необходимые для определения Z_{oc} и Y_{sc}, получаются следующим образом:

Остовы:

(a, c, e), (c, b, e), (b, d, e), (a, b, c),
(a, b, d), (a, c, d), (a, d, e) и (c, d, e).

$T_{1,\,0}$: (b, c), (b, d), (c, d), (a, d),
(a, b), (b, e), (c, e) и (a, e).

$T_{2,\,3}$: (a, b), (a, e), (a, c), (c, e) и (b, e).

$T_{12,\,30}$: (a, b), (a, e), (b, e) и (c, e).

$T_{1,\,2,\,30}$: (a), (b) и (c).

Из приведенных выше деревьев можно получить

$$W = 60s^3 + 111s^2 + 16s,$$
$$W_{1,\,0} = 27s^2 + 33s + 2,$$
$$W_{2,\,3} = 12s^2 + 15s + 2,$$
$$W_{12,\,30} = 12s^2 + 12s + 2,$$
$$W_{1,\,2,\,30} = 3s + 3.$$

Используя эти выражения и теоремы 13.2 и 13.3, можно получить Z_{oc} и Y_{sc}:

$$Z_{oc} = \frac{1}{60s^3 + 111s^2 + 16s} \begin{bmatrix} 27s^2 + 33s + 2 & 12s^2 + 12s + 2 \\ 12s^2 + 12s + 2 & 12s^2 + 15s + 2 \end{bmatrix},$$

$$Y_{sc} = \frac{1}{3s + 3} \begin{bmatrix} 12s^2 + 15s + 2 & -(12s^2 + 12s + 2) \\ -(12s^2 + 12s + 2) & 27s^2 + 33s + 2 \end{bmatrix}.$$

13.2. Топологические формулы для общих линейных цепей

В этом разделе выводятся топологические формулы для общих линейных инвариантных по времени цепей с сосредоточенными параметрами. Эти формулы выражаются в виде произведений проводимостей ориентированных деревьев и 2-деревьев соответствующего ориентированного графа, связанного с данной цепью. Отправной точкой для вывода этих формул является матрица неопределенных проводимостей, которую мы теперь определяем.

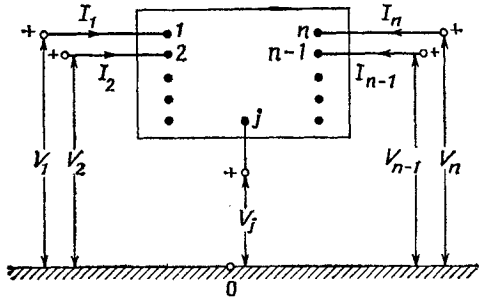


Рис. 13.4. Цепь с n зажимами.

Рассмотрим цепь N с n зажимами 1, 2, . . ., n. Пусть V_1, V_2, . . ., V_n — напряжения на этих зажимах относительно произвольного (внешнего к цепи) зажима 0. Пусть I_1, I_2, ..., I_n — токи, текущие к этим зажимам (рис. 13.4) при наличии внешних соединений.

Матрица $\hat{Y}$, описывающая эти токи и напряжения на входах, известна как матрица неопределенных проводимостей цепи N. Таким образом, $\hat{Y}V=I$, т. е.

$$\begin{bmatrix} \hat{y}_{11} & \hat{y}_{12} & \cdots & \hat{y}_{1n} \\ \hat{y}_{21} & \hat{y}_{22} & \cdots & \hat{y}_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & & \cdot \\ \hat{y}_{n1} & \hat{y}_{n2} & \cdots & \hat{y}_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \cdot \\ \cdot \\ \cdot \\ V_n \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ \cdot \\ \cdot \\ \cdot \\ I_n \end{bmatrix}. \tag{13.22}$$

Если говорить о цепи N как о n-полюсной, в которой i-й зажим и внешний зажим 0 образуют i-й полюс, то можно видеть, что матрица неопределенных проводимостей $\hat{Y}$ — матрица проводимостей короткого замыкания такой n-полюсной цепи. Таким образом,

$$\hat{y}_{jk} = \frac{I_j}{V_k}\bigg|_{\forall V_i=0,\ i\neq k}.$$

Другими словами, если подсоединить источник напряжения единичной мощности между зажимом k и внешним зажимом 0 и замкнуть все остальные зажимы на внешний зажим, то величина тока I_j, входящего в зажим j, равна $\hat{y}_{jk}$. Можно использовать эту интерпретацию для того, чтобы вычислить все элементы матрицы

$\hat{Y}$. Например, можно показать, что матрица неопределенных проводимостей цепи на рис. 13.5 имеет вид

$$\begin{bmatrix} 1+y & -y & -1 & 0 \\ -y & 2+y & -1 & -1 \\ -1 & -1-g_m & 2+g_p+g_m & -g_p \\ 0 & -1+g_m & -g_p-g_m & 1+g_p \end{bmatrix}. \qquad (13.23)$$

Этот пример заимствован из работы [13.3]. Далее будут установлены два полезных свойства матрицы неопределенных проводимостей.

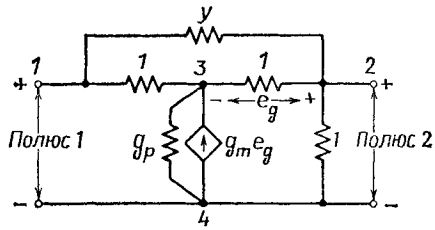


Рис. 13.5.

В соответствии с законом Кирхгофа для токов $I_1+I_2+\ldots+I_n=0$. Следовательно, $(\hat{y}_{11}+\hat{y}_{21}+\ldots+\hat{y}_{n1})V_1+(\hat{y}_{12}+\hat{y}_{22}+\ldots+\hat{y}_{n2})V_2+ +\ldots+(\hat{y}_{1n}+\hat{y}_{2n}+\ldots+\hat{y}_{nn})V_n=0$.

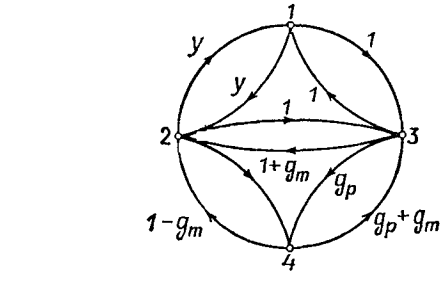
Заметим, что все напряжения V_1, V_2, . . ., V_n являются независимыми. Предположим, установлено, что $V_i=0$ для всех $i\neq k$. Тогда приведенные выше выражения сводятся к следующему: $(\hat{y}_{1k}+\hat{y}_{2k}+\ldots+\hat{y}_{nk})V_k=0$. Поскольку $V_k\neq 0$, отсюда следует, что $(\hat{y}_{1k}+\hat{y}_{2k}+\ldots+\hat{y}_{nk})=0$. Таким образом, сумма всех элементов в каждом столбце матрицы неопределенных проводимостей равна нулю. Аналогичный результат можно получить для строк матрицы $\hat{Y}$, а именно сумма всех элементов в любой строке матрицы $\hat{Y}$ равна нулю.

Таким образом, можно сформулировать следующий результат:

Теорема 13.4. Матрица неопределенных проводимостей электрической цепи является матрицей равных алгебраических дополнений.

Другой привлекательной чертой матрицы неопределенных проводимостей является то, что она дает базовое описание цепи в том смысле, что некоторые другие описания цепи получаются исходя из этой матрицы.

Например, матрицу узловых проводимостей с i-м выделенным элементом можно получить удалением i-й строки и i-го столбца из матрицы неопределенных проводимостей. Для более детального



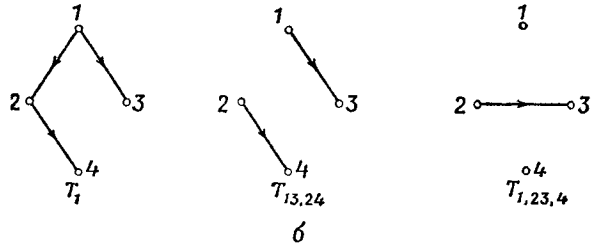


Рис. 13.6.

а — граф, соответствующий матрице неопределенных проводимостей (13. 23); *б* — ориентированное дерево, ориентированное 2-дерево, ориентированное 3-дерево.

обсуждения применения матрицы неопределенных проводимостей в анализе активных цепей рекомендуем работу [13.3].

Далее рассмотрим топологическую оценку алгебраических дополнений матрицы неопределенных проводимостей. Для данной матрицы неопределенных проводимостей $\hat{Y}$ цепи N, имеющей n узлов, сначала построим взвешенный ориентированный граф $G(\hat{Y})$ следующим образом: 1) $G(\hat{Y})$ имеет n узлов, отмеченных целыми числами 1, 2, . . ., n. 2) Если $\hat{y}_{ij} \neq 0$ для i, $j=1, 2, \ldots, n$, $i \neq j$, то существует ребро в графе $G(\hat{Y})$, направленное от узла i к узлу j с весом $-y_{ij}$. Например, ориентированный граф, связанный с матрицей неопределенных проводимостей (13.23), представлен на рис. 13.6, *а*.

В последующем изложении обозначим через T_i ориентированный остов в графе $G(\hat{Y})$ с узлом i в качестве корня. Ориентированное остовное 2-дерево в графе $G(\hat{Y})$ есть такое остовное 2-дерево в неориентированном графе, что каждая компонента ориентированного 2-дерева является ориентированным деревом в графе $G(\hat{Y})$. Обозначим через $T_{ij,\ kl}$ такое остовное ориентированное 2-дерево, что 1) узлы i и j находятся в одной компоненте, а узлы k и l — в разных компонентах и 2) узлы i и k являются корнями компонент

2-дерева $T_{ij,\,kl}$. Ориентированное остовное 3-дерево $T_{ij,\,rs}$ определяется аналогично. Узлы i, k и r являются корнями трех компонент 3-дерева $T_{ij,\,kl,\,rs}$. Примеры ориентированного остова, остовных 2-дерева и 3-дерева показаны на рис. 13.6, *б*.

Пусть также определены:

W_i — сумма произведений проводимостей всех ориентированных остовов T_i из графа $G(\hat{Y})$;

$W_{ij,\,kl}$ — сумма произведений проводимостей всех остовных ориентированных 2-деревьев $T_{ij,\,kl}$ из графа $G(\hat{Y})$;

$W_{ij,\,kl,\,rs}$ — сумма произведений проводимостей всех остовных ориентированных 3-деревьев $T_{ij,\,kl,\,rs}$ из графа $G(\hat{Y})$.

Можно показать (упражнение 6.18), что

$$\hat{\Delta}_{kk} = W_k. \tag{13.24}$$

В следующей теореме сформулировано соотношение между алгебраическими дополнениями 2-го порядка ($\hat{Y}$) и произведениями проводимостей соответствующего ориентированного 2-дерева.

Теорема 13.5. Пусть $\hat{Y}$ — матрица неопределенных проводимостей цепи N. Тогда $\hat{\Delta}_{ij,\,kk} = W_{ji,k}$.

Доказательство. Пусть G^* — ориентированный граф, полученный из графа $G(\hat{Y})$ изменением весов ребер (i, j), (k, j) и (i, k) со значений $-y_{ij}$, $-y_{kj}$ и $-y_{ik}$ на значения $-y_{ij}+K$, $-y_{kj}-K$ и $-y_{ik}-K$ соответственно. Пусть $\hat{Y}^*$ — соответствующая матрица алгебраических дополнений. Будем говорить о ребре (i, j) в графе G^* как состоящем из двух параллельных ребер (i, j) и $(i, j)_2$ с весами $-y_{ij}$ и K соответственно. Аналогично будем рассматривать ребро (k, j) как состоящее из двух параллельных ребер $(k, j)_1$ и $(k, j)_2$ с весами $-y_{kj}$ и $-K$ соответственно.

$$\hat{\Delta}^*_{kk} = W^*_k, \tag{13.25}$$

где значок $*$ используется для обозначения величин, относящихся к графам $\hat{Y}^*$ и G^*. Произведем разбиение множества S всех ориентированных остовов T^*_k из графа G^* на три множества S_1, S_2 и S_3, определенных следующим образом: $S_1 = \{T^*_k \mid T^*_k \in S$ и ни $(i, j)_2$, ни $(k, j)_2$ не принадлежат $T^*_k\}$, $S_2 = \{T^*_k \mid T^*_k \in S$, $(k, j)_2$ принадлежит T^*_k, и существует ориентированный путь из i в j во множестве $T^*_k\}$, $S_3 = S - (S_1 \cup S_2)$. Поскольку граф G можно получить из графа G^*, полагая $K=0$, то отсюда следует, что

$$\hat{\Delta}_{kk} = \sum_{T^*_k \in S} \text{произведения проводимостей } T^*_k. \tag{13.26}$$

Для любого ориентированного остова $T^*_k \in S_2$ $T^*_k - (k, j)_2$ является ориентированным остовным 2-деревом $T_{ji,\,k}$ из графа $G(\hat{Y})$ и наоборот. Таким образом,

$$\sum_{T^*_k \in S_2} \text{произведения проводимостей } T^*_k = -K \sum_{\forall T_{ij,\,k}} \text{произведения проводимостей } T_{ji,\,k} = -K\, W_{ji,k}. \tag{13.27}$$

Теперь отметим, что любое ориентированное дерево T^*_k в S_3 относится к одному из следующих типов: 1) T_k содержит $(i, j)_2$, 2) T^*_k содержит $(k, j)_2$ и в T^*_k

не существует ориентированного пути из j в i. Нетрудно видеть, что для каждого $T_k^* \in S_3$ первого типа существует единственное ориентированное дерево $\{T_k^* - (i, j)_2 \cup (k, j)_2\} \in S_3$, которое относится ко второму типу, и наоборот. Произведение проводимостей данного дерева имеет обратный знак по отношению к произведению проводимостей другого. Поэтому

$$\sum_{T_k^* \in S_3} \text{произведения проводимостей } T_k^* = 0. \tag{13.28}$$

Из выражения (13.26) с использованием равенства (13.28) получаем

$$\hat{\Delta}_{kk}^* = \hat{\Delta}_{kk} - KW_{ji,\,k}. \tag{13.29}$$

Наконец, имеем также следующее соотношение:

$$\hat{\Delta}_{kk}^* = \Delta_{kk} - K\hat{\Delta}_{ij,\,kk}. \tag{13.30}$$

Из выражений (13.29) и (13.30) получаем $\hat{\Delta}_{ij,\,kk} = W_{ji,\,k}$.
Нетрудно показать, используя теорему 13.5, что

$$\hat{\Delta}_{ii,\,jj,\,kl} = W_{i,\,j,\,lk}. \tag{13.31}$$

Теперь приступим к выводу топологических формул для функций холостого хода и короткого замыкания общей линейной цепи с сосредоточенными параметрами.

Рис. 13.7. Усложненная 2-полюсная цепь.

Рассмотрим 2-полюсную цепь N, представленную на рис. 13.7. Пусть Y — ее матрица проводимостей узлов с отмеченным узлом n. Эту матрицу можно получить, удалив n-ю строку и n-й столбец из матрицы неопределенных проводимостей $\hat{Y}$ цепи N. Тогда

$$Y = \hat{Y}_{nn}. \tag{13.32}$$

Исходя из узловых уравнений $YV = I$, можно получить матрицы полных сопротивлений короткого замыкания и проводимостей холостого хода цепи N в следующем виде:

$$Z_{oc} = \frac{1}{\Delta}\begin{bmatrix} \Delta_{11} & \Delta_{21} - \Delta_{31} \\ \Delta_{12} - \Delta_{13} & \Delta_{22} + \Delta_{33} - \Delta_{23} - \Delta_{32} \end{bmatrix}, \tag{13.33}$$

$$Y_{sc} = \frac{1}{\Delta_{11,\,22} + \Delta_{11,\,33} - \Delta_{11,\,23} - \Delta_{11,\,32}}\begin{bmatrix} \Delta_{22} + \Delta_{33} - \Delta_{23} - \Delta_{32} & \Delta_{31} - \Delta_{21} \\ \Delta_{13} - \Delta_{12} & \Delta_{11} \end{bmatrix}. \tag{13.34}$$

Чтобы получить топологические формулы для элементов Z_{oc} и Y_{sc}, сначала замечаем, используя формулу (13.24), что $\Delta = \hat{\Delta}_{nn} = W_n$.

Применяя также теорему 13.5, имеем

$$\Delta_{11} = \hat{\Delta}_{11,\ nn} = W_{1,\ n}, \tag{13.35}$$

$$\Delta_{21} - \Delta_{31} = \hat{\Delta}_{21,\ nn} - \hat{\Delta}_{31,\ nn} = W_{12,\ n} - W_{13,\ n} =$$
$$= W_{123,\ n} + W_{12,\ n3} - W_{132,\ n} - W_{13,\ n2} = W_{12,\ n3} - W_{13,\ n2}, \tag{13.36}$$

$$\Delta_{12} - \Delta_{13} = W_{21,\ 3n} - W_{31,\ 2n}. \tag{13.37}$$

Теперь

$$\Delta_{22} + \Delta_{33} - \Delta_{32} - \Delta_{23} =$$
$$= \hat{\Delta}_{22,\ nn} + \hat{\Delta}_{33,\ nn} - \hat{\Delta}_{32,\ nn} - \hat{\Delta}_{23,\ nn} = \hat{\Delta}_{22,33}. \tag{13.38}$$

Последний шаг в приведенном выше уравнении следует из тождества [13.4] $\hat{\Delta}_{pq,\ rs} = \hat{\Delta}_{pq,\ uv} + \hat{\Delta}_{rs,\ uv} - \hat{\Delta}_{ps,\ uv} - \hat{\Delta}_{rq,\ uv}$. Таким образом, из теоремы 13.5 следует

$$\Delta_{22} + \Delta_{33} - \Delta_{32} - \Delta_{23} = \hat{\Delta}_{22,\ 33} = W_{2,\ 3}. \tag{13.39}$$

Далее,

$$\Delta_{11,\ 22} + \Delta_{11,\ 33} - \Delta_{11,\ 23} - \Delta_{11,\ 32} =$$
$$= \hat{\Delta}_{11,\ 22,\ nn} + \hat{\Delta}_{11,\ 33,\ nn} - \hat{\Delta}_{11,\ 23,\ nn} - \hat{\Delta}_{11,\ 32,\ nn} =$$
$$= W_{1,\ 2,\ n} + W_{1,\ 3,\ n} - W_{1,\ 32,\ n} - W_{1,\ 23,\ n} =$$
$$= W_{13,\ 2,\ n} + W_{1,\ 2,\ n3} + W_{12,\ 3,\ n} + W_{1,\ 3,\ n2}. \tag{13.40}$$

На основе приведенных уравнений получаем следующий результат.

Теорема 13.6. Пусть N является 2-полюсной цепью с положительными и отрицательными зажимами полюсов, как показано на рис. 13.7. Тогда матрица полных сопротивлений холостого хода и матрица проводимостей короткого замыкания цепи N определяется выражениями

$$Z_{oc} = \frac{1}{W_n} \begin{bmatrix} W_{1,\ n} & W_{12,\ n3} - W_{13,\ n2} \\ W_{21,\ 3n} - W_{31,\ 2n} & W_{2,\ 3} \end{bmatrix},$$

$$Y_{sc} = \frac{1}{W_{13,\ 2,\ n} + W_{1,\ 2,\ n3} + W_{12,\ 3,\ n} + W_{1,\ 3,\ n2}} \begin{bmatrix} W_{2,\ 3} & W_{13,\ n2} - W_{12,\ n3} \\ W_{31,\ 2n} - W_{21,\ 3n} & W_{1,\ n} \end{bmatrix}.$$

Проиллюстрируем топологическое определение Z_{oc} и Y_{sc}, используя теорему 13.6.

Снова рассмотрим цепь N на рис. 13.5 с узлами 1 и 4, образующими полюс 1, и узлами 2 и 4, образующими полюс 2. Матрица неопределенных проводимостей $\hat{Y}$ цепи N дана в (13.23). Граф $G(\hat{Y})$, соответствующий $\hat{Y}$, показан на рис. 13.6, *а*.

Чтобы вычислить Z_{oc} и Y_{sc}, используя теорему 13.6, необходимо вычислить для графа $G(\hat{Y})$ значения W_4, $W_{1,\ 4}$, $W_{12,\ 4}$, $W_{21,\ 4}$, $W_{2,\ 4}$ и $W_{1,\ 2,\ 4}$. Сначала получим искомые остовы, остовные 2-де-

ревья и 3-деревья графа $G(\hat{Y})$:

T_4: $\{(2, 1), (4, 2), (4, 3)\}$, $\{(4, 2), (4, 3), (3, 1)\}$,
$\{(1, 2), (3, 1), (4, 3)\}\}$, $\{(2, 1), (4, 2); (1, 3)\}$,
$\{(2, 1), (2, 3), (4, 2)\}$, $\{(3, 1), (3, 2), (4, 3)\}$,
$\{(2, 1), (3, 2), (4, 3)\}$, $\{(3, 1), (2, 3), (4, 2)\}$.

[Заметим, что (i, j) является ребром, направленным от i к j]

$T_{1,4}$: $\{(1, 2), (4, 3)\}$, $\{(1, 2), (1, 3)\}$, $\{(1, 2), (2, 3)\}$, $\{(1, 3),(3, 2)\}$,
$\{(1, 3), (4, 2)\}$, $\{(3, 2), (4, 3)\}$, $\{(4, 2), (2, 3)\}$, $\{(4, 2), (4, 3)\}$.
$T_{2,4}$: $\{(2, 1), (4, 3)\}$, $\{(2, 1), (1, 3)\}$, $\{(2, 1), (2, 3)\}$,
$\{(3, 1), (2, 3)\}$, $\{(3, 1), (4, 3)\}$.
$T_{1\bar{2},4}$: $\{(1, 2), (1, 3)\}$, $\{(1, 2), (2, 3)\}$, $\{(1, 3), (3, 2)\}$, $\{(1, 2), (4,3)\}$.
$T_{\bar{2}1,4}$: $\{(2, 1), (1, 3)\}$, $\{(2, 1), (2, 3)\}$, $\{(3, 1), (2, 3)\}$, $\{(2, 1), (4,3)\}$.
$T_{1,2,4}$: $\{(1, 3)\}$, $\{(2, 3)\}$, $\{(4, 3)\}$.

Отсюда получаем следующие произведения проводимостей:

$$W_4 = g_m(y+1) + g_p(3y+2) + (2y+1),$$
$$W_{1,4} = (2+y)(2+g_p+g_m) - (1+g_m),$$
$$W_{2,4} = (1+y)(2+g_p+g_m) - 1,$$
$$W_{12,4} = y(2+g_p+g_m) + (1+g_m),$$
$$W_{21,4} = y(2+g_p+g_m) + 1,$$
$$W_{1,2,4} = 2+g_p+g_m.$$

Таким образом, получаем

$$Z_{oc} = \frac{1}{W_4}\begin{bmatrix} W_{1,4} & W_{12,4} \\ W_{21,4} & W_{\bar{2},4} \end{bmatrix} = \frac{1}{g_m(y+1)+g_p(3y+2)+(2y+1)} \times$$
$$\times \begin{bmatrix} (2+y)(2+g_p+g_m)-(1+g_m) & y(2+g_p+g_m)+(1+g_m) \\ y(2+g_p+g_m)+1 & (1+y)(2+g_p+g_m)-1 \end{bmatrix},$$

$$Y_{sc} = \frac{1}{W_{1,2,4}}\begin{bmatrix} W_{2,4} & -W_{12,4} \\ -W_{21,4} & W_{1,4} \end{bmatrix} =$$
$$= \frac{1}{2+g_p+g_m}\begin{bmatrix} (1+y)(2+g_p+g_m)-1 & -y(2+g_p+g_m)-(1+g_m) \\ -y(2+g_p+g_m)-1 & (2+y)(2+g_p+g_m)-(1+g_m) \end{bmatrix}.$$

13.3. Сопряженная цепь и вычисление чувствительности цепи

В этом разделе развивается метод вычисления чувствительности цепей. Этот метод основан на теореме Теллежена и понятии «сопряжение цепи» [13.5, 13.6]. Приведенные здесь рассуждения основаны на данных работы [13.6].

Пусть N — линейная неизменная во времени цепь с сосредоточенными параметрами, состоящая из сопротивлений, емкостей, индуктивностей, трансформаторов, гираторов, управляемых по напряжению источников тока и напряжений и управляемых по току источников тока и напряжения. Допустим, что N является 2-полюсной цепью, хотя последующие рассуждения справедливы даже тогда, когда цепь N имеет более двух полюсов.

Пусть $\tilde{N}$ является 2-полюсной цепью, которая топологически эквивалентна цепи N. Другими словами, граф цепи $\tilde{N}$ идентичен графу цепи N. Соответствующие элементы цепей N и $\tilde{N}$ будут обозначаться одинаковыми символами. Заметим, что элементы цепи $\tilde{N}$ и их значения пока не определены.

Пусть V_e и I_e — напряжение и ток соответственно, связанные с элементом e в цепи $\tilde{N}$, а ψ_e и λ_e — напряжение и ток соответственно, связанные с соответствующим элементом e в цепи $\tilde{N}$. Тогда V_i и I_i (i равно 1, 2) будут обозначать переменные напряжения и тока, связанные для полюсов цепи N, а ψ_i и λ_i (i равно 1, 2) будут обозначать соответствующие переменные для полюсов цепи $\tilde{N}$.

Применяя теорему Теллежена к цепям N и $\tilde{N}$, получим

$$V_1\lambda_1 + V_2\lambda_2 = \sum_e V_e\lambda_e, \tag{13.41а}$$

$$I_1\psi_1 + I_2\psi_2 = \sum_e I_e\psi_e, \tag{13.41б}$$

где суммирование проводится по всем элементам цепей N и $\tilde{N}$.

Предположим, что значения элементов цепи N изменились. Тогда теорема Теллежена, примененная к возмущенной цепи N и цепи $\tilde{N}$, дает следующий результат:

$$(V_1 + \Delta V_1)\lambda_1 + (V_2 + \Delta V_2)\lambda = \sum_e (V_e + \Delta V_e)\lambda_e, \tag{13.42а}$$

$$(I_1 + \Delta I_1)\psi_1 + (I_2 + \Delta I_2)\psi_2 = \sum_e (I_e + \Delta I_e)\psi_e. \tag{13.42б}$$

где ΔV и ΔI представляют собой изменения напряжения и тока, которые произошли в результате изменения величин элементов в цепи N.

Вычитая выражение (13.41) из выражения (13.42), получим

$$\Delta V_1\lambda_1+\Delta V_2\lambda_2=\sum_e \Delta V_e\lambda_e, \tag{13.43}$$

$$\Delta I_1\psi_1+\Delta I_2\psi_2=\sum_e \Delta I_e\psi_e. \tag{13.44}$$

Вычитая выражение (13.43) из выражения (13.44), имеем

$$(\Delta V_1\lambda_1-\Delta I_1\psi_1)+(\Delta V_2\lambda_2-\Delta I_2\psi_2)=\sum_e(\Delta V_e\lambda_e-\Delta I_e\psi_e). \tag{13.45}$$

Теперь желательно определить элементы в цепи $\tilde{N}$ таким образом, чтобы выражение (13.45) не зависело от всех ΔV_e и ΔI_e-членов.

Сначала рассмотрим резисторы. Имеем

$$V_R=RI_R, \tag{13.46}$$

где V_R и I_R — напряжение и ток, связанные с резистором. Пусть R изменяется до $R+\Delta R$. Тогда выражение $V_R+\Delta V_R=(R+\Delta R)(I_R+\Delta I_R)$ можно упростить, пренебрегая членами второго порядка малости:

$$V_R+\Delta V_R=RI_R+R\Delta I_R+I_R\Delta R. \tag{13.47}$$

Вычитая выражение (13.46) из выражения (13.47), получим

$$\Delta V_R=R\Delta I_R+I_R\Delta R. \tag{13.48}$$

Следовательно, члены в выражении (13.45), связанные с резисторами цепи N, можно записать в виде

$$\sum_R[(R\lambda_R-\psi_R)\Delta I_R+I_R\lambda_R\Delta R], \tag{13.49}$$

где суммирование проводится по всем резисторам цепи N.

Заметим, что индексы у величин λ и ψ не обязательно говорят об отношении к резисторам цепи $\tilde{N}$, но явно указывают на соответствие между элементами цепей N и $\tilde{N}$.

Если теперь выбрать

$$\psi_R=R\lambda_R, \tag{13.50}$$

то выражение (13.49) упрощается до выражения

$$\sum_R I_R\lambda_R\Delta R, \tag{13.51}$$

которое не зависит от ΔV_R и ΔI_R.

Уравнение (13.50) является соотношением для сопротивления величины R. Поэтому элемент в цепи $\tilde{N}$, который соответствует резистору величины R в цепи N, является также резистором величины R.

Таблица 13.1

Тип элемента	*Соотношение на ветви*	*Соотношение на ветви сопряжения*	*Чувствительность (компонента ϑ)*	*Компонента Δp*
Сопротивление	$V_R=RI_R$	$\psi_R=R\lambda_R$	$+I_R\lambda_R$	ΔR
Проводимость	$I_G=GV_G$	$\lambda_G=G\psi_G$	$-V_G\psi_G$	ΔG
Емкость	$I_C=j\omega CV_C$	$\lambda_C=j\omega C\psi_C$	$-j\omega V_C\psi_C$	ΔC
Взаимная емкость	$SI_S=j\omega V_S$	$S\lambda_S=j\omega\psi_S$	$+\frac{1}{j\omega}I_S\lambda_S$	ΔS
Индуктивность	$V_L=j\omega LI_L$	$\psi_L=j\omega L\lambda_L$	$+j\omega I_L\lambda_L$	ΔL
Взаимная индуктивность	$\Gamma V_\Gamma=j\omega I_\Gamma$	$\Gamma\psi_\Gamma=j\omega\lambda_\Gamma$	$-\frac{1}{j\omega}V_\Gamma\psi_\Gamma$	$\Delta\Gamma$
Трансформатор	$V_{T2}=nV_{T1}$ $I_{T1}=-nI_{T2}$	$\psi_{T2}=n\psi_{T1}$ $\lambda_{T1}=-n\lambda_{T2}$	$+(V_{T1}\lambda_{T2}+I_{T2}\psi_{T1})$	Δn
Гиратор	$V_{GY1}=\alpha I_{GY2}$ $V_{GY2}=-\alpha I_{GY1}$	$\psi_{GY1}=-\alpha\lambda_{GY2}$ $\psi_{GY2}=\alpha\lambda_{GY1}$	$+(I_{GY2}\lambda_{GY1}-I_{GY1}\lambda_{GY2})$	$\Delta\alpha$
Источник напряжения, управляемый по напряжению	$V_{VV2}=\mu V_{VV1}$ $I_{VV1}=0$	$\lambda_{VV1}=-\mu\lambda_{VV2}$ $\psi_{VV2}=0$	$+V_{VV1}\lambda_{VV2}$	$\Delta\mu$
Источник тока, управляемый по напряжению	$I_{VI2}=g_mV_{VI1}$ $I_{VI1}=0$	$\lambda_{VI1}=g_m\psi_{VI2}$ $\lambda_{VI2}=0$	$-V_{VI1}\psi_{VI2}$	Δg_m
Источник напряжения, управляемый по току	$V_{IV2}=r_mI_{IV1}$ $V_{IV1}=0$	$\psi_{IV1}=r_m\lambda_{IV2}$ $\psi_{IV2}=0$	$+I_{IV1}\lambda_{IV2}$	Δr_m
Источник тока, управляемый по току	$I_{II2}=\beta I_{II1}$ $V_{II1}=0$	$\psi_{II1}=-\beta\psi_{II2}$ $\lambda_{II2}=0$	$-I_{II1}\psi_{II2}$	$\Delta\beta$

Рассмотрим далее управляемый по напряжению источник напряжения, определяемый соотношениями $V_{VV2}=\mu V_{VV1}$ и $I_{VV1}=0$. Затем, пренебрегая членами второго порядка малости, имеем $\Delta V_{VV2}=\mu\Delta V_{VV1}+V_{VV1}\Delta\mu$ и $\Delta I_{VV1}=0$.

Члены выражения (13.45), соответствующие источнику, управляемому по напряжению, можно теперь переписать в виде

$$\sum[(\lambda_{VV1}+\mu\lambda_{VV2})\,\Delta V_{VV1}-\psi_{VV2}\Delta I_{VV2}+V_{VV1}\lambda_{VV2}\Delta\mu]. \quad (13.52)$$

Если теперь выбрать

$$\lambda_{VV1}=-\mu\lambda_{VV2},\ \psi_{VV2}=0. \tag{13.53}$$

то выражение (13.52) упрощается до $\Sigma V_{VV1}\lambda_{VV2}\Delta\mu$.

Заметим, что выражение (13.53) описывает источник, управляемый по току с коэффициентом усиления $-\mu$. Отметим также разницу между управляющими и зависимыми элементами в сопряженной цепи.

Члены выражения (13.45) для остальных типов элементов цепи N можно получить аналогичным способом.

Результаты сведены в табл. 13.1, причем соответствующие соотношения на ветвях должны быть выбраны для сопряженной цепи $\tilde{N}$.

Если элементы $\tilde{N}$ выбираются, как приведено в табл. 13.1, то выражение (13.45) можно упростить:

$$\sum(\Delta V_1\lambda_1-\Delta I_1\psi_1)+(\Delta V_2\lambda_2-\Delta I_2\psi_2)=\mathcal{G}^t\Delta p, \tag{13.54}$$

где $\mathcal{G}$ и Δp — векторы, компоненты которых приведены в табл. 13.1.

Цепь $\tilde{N}$, элементы которой определены в соответствии с табл. 13.1, называется *сопряженной* цепью N. Представления различных элементов цепи N и их соответствующих элементов в сопряженной цепи $\tilde{N}$ приведены на рис. 13.8.

Заметим, что сопряженная цепь $\tilde{N}$ находится в следующем отношении к исходной цепи:

1) Граф $\tilde{N}$ тождествен графу N.

2) Все резистивные, емкостные и индуктивные элементы цепи N соответствуют резистивным, емкостным и индуктивным элементам таких же величин цепи $\tilde{N}$.

3. Все трансформаторы с отношением витков $1:n$ цепи N соответствуют трансформаторам с отношением витков $1:n$ цепи $\tilde{N}$.

4. Все гираторы с гирационным отношением α в цепи N соответствуют гираторам с гирационным отношением $-\alpha$ цепи $\tilde{N}$ (или два полюса гиратора меняются местами).

5. Источники напряжения, управляемые по напряжению, с коэффициентом усиления по напряжению μ цепи N соответствуют источникам тока, управляемым по току, с коэффициентом усиления по току $-\mu$ цепи $\tilde{N}$, и роли управляющих и зависимых элементов цепи N меняются на обратные в цепи $\tilde{N}$.

6. Источники тока, управляемые по току, с коэффициентом усиления по току β в цепи N соответствуют источникам напряжения, управляемым по напряжению, с коэффициентом усиления по напряжению $-\beta$ в цепи $\tilde{N}$, и роли управляющих и зависимых элементов в цепи N меняются на обратные в цепи $\tilde{N}$.

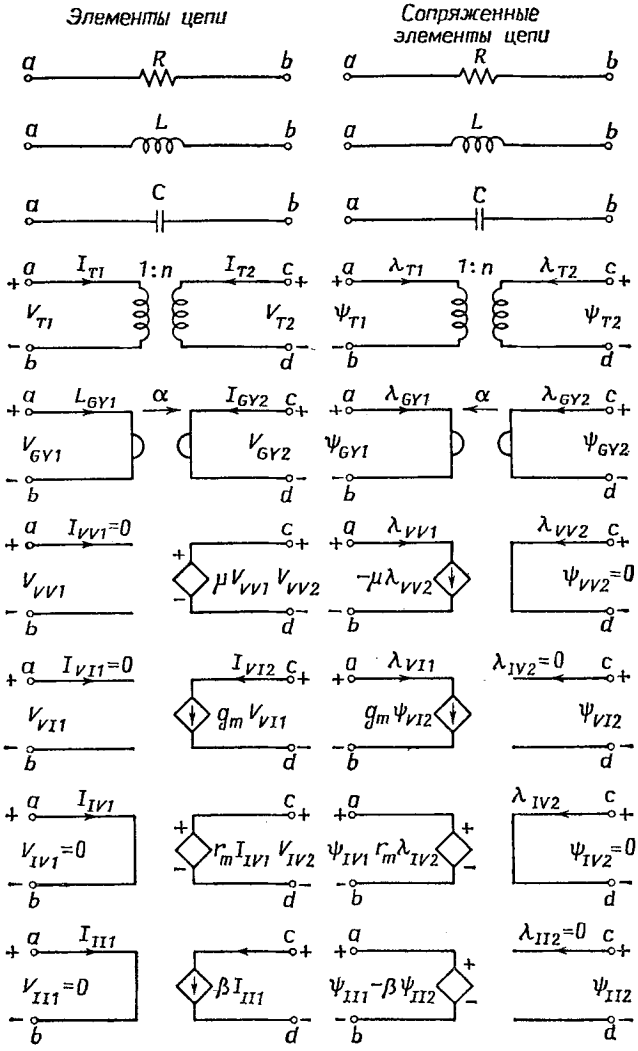


Рис. 13.8. Элементы цепи и их сопряжения.

7. Источники тока, управляемые по напряжению, и источники напряжения, управляемые по току, в цепи N соответствуют источникам тока, управляемым по напряжению, и источникам напряжения, управляемым по току соответственно в цепи $\tilde{N}$, и роли управляющих и зависимых элементов цепи N меняются на обратные в цепи $\tilde{N}$.

Проиллюстрируем применение сопряженной цепи в вычислении чувствительностей функции цепи. Чувствительность S_x^F функции цепи $F(s)$ по отношению к параметру x цепи определяется в виде $S_x^F = (x/F)(\partial F/\partial x)$. Очевидно, что чувствительность S_x^F является мерой воздействия на $F(s)$ приращения параметра x. Вычисление S_x^F существенно использует определение $\partial F/\partial x$. Это можно сделать следующим образом:

1. Выбираем полюсные переменные таким образом, чтобы левую часть выражения (13.54) упростить до ΔF. (Другими словами, возбудить цепь N и сопряженную цепь $\tilde{N}$ в их соответствующих полюсах.) Например, предположим, что $F(s)$ является отношением напряжений холостого хода, т. е. $F(s)=V_2(s)/V_1(s)$. Тогда возбуждаем цепи N и $\tilde{N}$ следующим образом:

а) подключаем к полюсу 1 цепи N независимый источник напряжения постоянной величины 1. Тогда $V_1=1$ и $\Delta V_1=0$;

б) разрываем полюс 2 в цепи N. Таким образом, $I_2=0$;

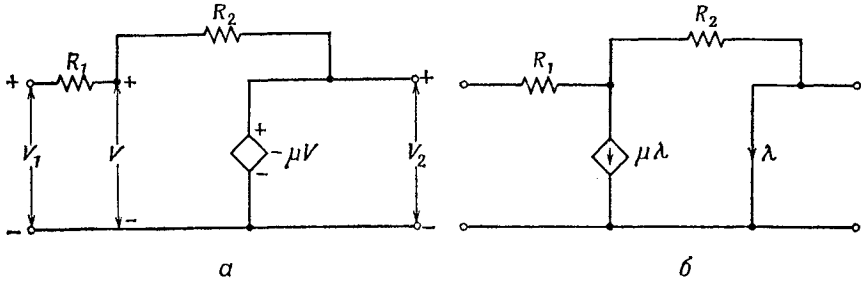
в) замыкаем накоротко полюс 1 цепи $\tilde{N}$, так что $\psi_1=0$;

г) подключаем к полюсу 2 цепи N независимый источник тока величины 1. Тогда $\lambda_2=1$.

Легко проверить, что приведенный выше выбор возбуждений полюсов упрощает левую часть выражения (13.54) до $\Delta V_2=\Delta F$.

2. Анализируем цепи N и $\tilde{N}$ после возбуждений, описанных выше и определяем все токи и напряжения на элементах.

3. Выражение в правой части (13.54), соответствующее параметру x, является произведением Δx и напряжений и(или) токов, соответствующих этому параметру (табл. 13.1). Поэтому $\partial F/\partial x$ для каждого параметра x можно определить после того, как напряжения и токи в цепях N и $\tilde{N}$ определены, как описано в п. 2.



а б

Рис. 13.9. Цепь и ее сопряжения.

Подобное вычисление чувствительности, использующее понятие «сопряженная цепь», требует анализа данной цепи N и ее сопряженной цепи $\tilde{N}$.

Проиллюстрируем на примере описанный выше метод вычисления чувствительности. Пример заимствован из работы [13.3].

Рассмотрим цепь N, представленную на рис. 13.9, *а*. Сопряженная цепь $\tilde{N}$ показана на рис. 13.9, *б*. Отношение напряжений холостого хода $F(s)=V_2(s)/V_1(s)$ цепи N задается функцией $F(s)=-\mu R_2/(R_1+$

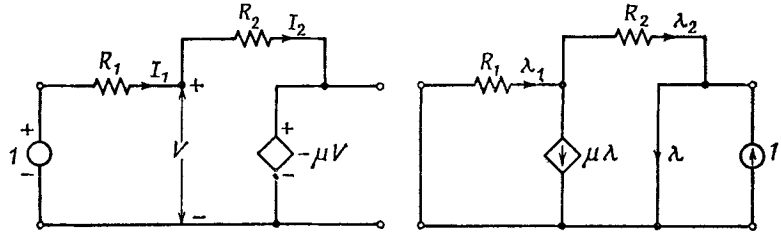


Рис. 13.10. Возбуждение полюсов цепи на рис. 13.9.

$+R_2+\mu R_1)$. Как описано ранее, возбуждаем цепи N и $\tilde{N}$, как показано на рис. 13.10. Из табл. 13.1 имеем

$$\frac{\partial F}{\partial(-\mu)}=V\lambda, \quad \frac{\partial F}{\partial R_1}=I_1\lambda_1, \quad \frac{\partial F}{\partial R_2}=I_2\lambda_2.$$

Легко проверить, что

$$I_1=I_2=\frac{1+\mu}{R_1+R_2+\mu R_1},$$

$$V=\frac{R_2}{R_1+R_2+\mu R_1},$$

$$\lambda=\frac{R_1+R_2}{\mu R_1+R_1+R_2},$$

$$\lambda_1=\lambda+\mu\lambda-1=\frac{\mu R_2}{\mu R_1+R_1+R_2},$$

$$\lambda_2=\lambda-1=\frac{-\mu R_1}{\mu R_1+R_1+R_2}.$$

В результате получаем

$$S_\mu^F=\frac{R_1+R_2}{R_1+R_2+\mu R_1},$$

$$S_{R_1}^F=\frac{-R_1(\mu+1)}{R_1+R_2+\mu R_1},$$

$$S_{R_2}^F=\frac{R_1(1+\mu)}{R_1+R_2+\mu R_1}.$$

В гл. 11 (упражнение 11.3) было показано, что матрица полных сопротивлений контура планарной цепи равна матрице проводимостей сечений ее двойственной цепи. Можно задаться вопросом, существует ли для данной цепи N другая такая цепь N_T, что матрица неопределенных проводимостей цепи N_T была бы равна

транспозиции матрицы неопределенных проводимостей N. Ответ будет утвердительным: существует. В работе [13.7] показано, как построить такую цепь N_T, названную *транспозицией* цепи N. Оказалось, что N_T является той же самой цепью, что и сопряженная к цепи N [13.8]. Авторы этой работы получили теорему, аналогичную теореме Теллежена, но применимую к любым двум планарным цепям, имеющим двойственную топологию. Используя эту теорему, они построили из данной планарной цепи новую цепь, названную *обобщенной двойственной транспозицией*. Если N_D является двойственной к цепи N, а N_{DT} является транспозицией цепи N_D, то можно показать, что N_{DT} является специальным случаем обобщенной двойственной транспозиции цепи N.

Понятия «транспозиция» и «обобщенная двойственная транспозиция помогли увидеть единство между различными реализациями цепей (опубликованными в литературе), которые кажутся различными, но фактически являются взаимосвязанными через операции транспозиции и обобщенной двойственной транспозиции.

Например, в работе [13.3] предлагаются две структуры цепи, чтобы реализовать функцию преобразования напряжения. Можно показать, что первая из них является обобщенной двойственной транспозицией второй. Еще пример: авторы работы [13.10] предложили две структуры; одну для того, чтобы реализовать функцию преобразования напряжения, а другую, чтобы реализовать функцию преобразования тока. И снова первая из них является транспозицией второй. Примеры, которые показывают, как получить, используя понятия «транспозиция» и «обобщенная двойственная транспозиция», различные эквивалентные структуры, реализующие данную функцию цепи, можно найти в работе [13.9].

13.4. Замечания, касающиеся литературы

Работа [13.11] особенно рекомендуется для дальнейшего чтения по детальному выводу топологических формул для функций цепи. Среди работ, в которых обсуждаются топологические формулы, можно назвать работы [13.12—13.14].

Формулы для обобщенных цепей, приведенные в этой главе, выведены с использованием соответствующих деревьев, сопоставленных матрице неопределенных проводимостей цепи. Несколько авторов представили формулы, которые могут быть записаны непосредственно по данной цепи. (Например, см. работу [13.15], где некоторые относящиеся к теме статьи включены как ссылки.) Можно посмотреть работу [13.12] для изучения интересных свойств схемных функций RLC-цепей, которые выводятся с использованием топологических формул.

С целью изучения общей формы теоремы Теллежена и ее применений можно рекомендовать работы [13.16—13.18]. Для ознаком-

ления с применением понятия сопряженной цепи в синтезе резистивных n-полюсных цепей с помощью компьютера см. работу [13.19]. Понятие «сопряженная цепь» также использовалось для того, чтобы установить инварианты чувствительности и границы суммы величин чувствительности элементов для функций цепи [13.20, 13.21].

Упражнения

13.1. а) Докажите, что матрица проводимостей RLC-цепи без взаимных индуктивностей будет иметь полюс при $S=0$ тогда и только тогда, когда существует путь, состоящий только из индуктивностей между зажимами полюса.
б) Докажите, что матрица проводимостей RLC-цепи без взаимных индуктивностей будет иметь полюс в бесконечности тогда и только тогда, когда существует путь, состоящий только из емкостей между зажимами полюса.

13.2. Пусть

$$z_{11}=\frac{a_0+a_1s+\ldots+a_ns^n}{D(s)},$$

$$z_{22}=\frac{b_0+b_1s+\ldots+b_ms^m}{D(s)},$$

$$z_{21}=\frac{c_0+c_1s+\ldots+c_rs^r}{D(s)}$$

являются функциями полных сопротивлений холостого хода 2-полюсной RLC-цепи без взаимных индуктивностей. Используя топологические формулы, покажите, что $a_k \geqslant |c_k|$ и $b_k \geqslant |c_k|$.

13.3. Выведите топологические формулы для передаточного отношения напряжения холостого хода и передаточного отношения токов короткого замыкания обобщенной линейной цепи.

13.4. Найдите чувствительность по отношению к полному сопротивлению R нулевой точки цепи, показанной на рис. 13.11.

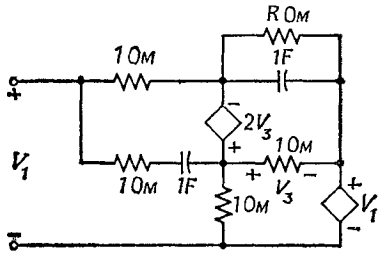


Рис. 13.11.

13.5. Покажите, что матрица неопределенных проводимостей цепи N является транспозицией матрицы неопределенных проводимостей своего сопряжения.
Примечание. Использовать теорему Теллежена.

13.6. В графовом представлении цепи, содержащей многополюсные элементы, каждый полюс представляется ребром. Пусть N и N_D являются планарными цепями, имеющими двойственные графы. Выведите теорему, аналогичную теореме Теллежена, которая использует только переменные токов или только переменные напряжений цепей N и N_D.

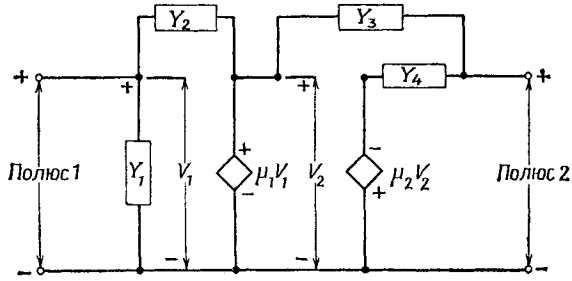


Рис. 13.12.

13.7. Пусть N является планарной цепью. Пусть N_D является двойственной к цепи N и пусть N_{DT} является транспозицией или сопряжением N_D. Используя теорему в упражнении 13.6 и процедуру, подобную той, которая описана в разд. 13.3, определите непосредственно из цепи N (без построения N_D) типы элементов и величины для цепи N_{DT} [13.9].

13.8. Для цепи N, представленной на рис. 13.12, постройте цепь N_{DT} (двойственную транспозицию цепи N), определенную в упражнении 13.7.

Часть III. Теория электрических цепей

14. Алгоритмы анализа графов

При исследовании отдельных практических задач появляются графы. Первым шагом в таких исследованиях было определение теоретико-графовых свойств рассматриваемой проблемы, которые помогли бы нам при определении метода ее решения. Так, в гл. 11—13 мы установили несколько полезных теоретико-графовых свойств электрических цепей и на основе использования этих свойств разработали различные методы вывода уравнений цепи. При изучении транспортных сетей наибольший интерес представляет определение максимального потока. В качестве первого шага в этом направлении мы выделяем в разд. 15.7 несколько свойств этих сетей, которые приведут нас к теореме о максимальном потоке и минимальном разрезе. Эта теорема образует основу алгоритма расстановки меток Форда — Фалкерсона для нахождения максимального потока. Обычно решение задачи включает анализ графа или проверку его на наличие того или иного свойства. Графы, возникающие при изучении реальных практических задач, очень громоздки и сложны. Поэтому для эффективного анализа таких графов необходима разработка соответствующих вычислительных алгоритмов.

В этой части книги мы обсудим некоторые алгоритмы на графах. Основное внимание мы уделяем установлению теоретической основы, на которой базируется конструирование алгоритмов. Мы также излагаем результаты, касающиеся вычислительной сложности некоторых из этих алгоритмов. В некоторых случаях вычислительная сложность всего алгоритма зависит главным образом от вычислительной сложности реализации определенных базовых операций, например объединения непересекающихся множеств (разд. 14.5). В таких случаях мы приводим ссылки на соответствующую литературу, которая в дальнейшем заинтересует читателя.

Вычислительная сложность алгоритма является мерой времени выполнения алгоритма. Поэтому она является функцией от размера входа. Для алгоритмов на графах результаты по их сложности могут быть выражены через число вершин и число ребер графа. В последующем о функции $g(n)$ будем говорить, что она есть $O(f(n))$ тогда и только тогда, когда существуют константы c и n_0, такие, что $|g(n)| \leqslant c|f(n)|$ для всех $n \geqslant n_0$. Кроме того, все

результаты по сложности будут относиться к анализу алгоритмов в худшем случае.

Существуют различные методы представления графа в вычислительной машине. Два из наиболее распространенных используют матрицу смежности (разд. 6.10) и список смежности. Представление графа матрицей смежности не очень эффективно в случае разреженных графов. При представлении графа списком смежности каждой вершине сопоставляется список, который содержит все ребра, инцидентные ей. Детальное обсуждение структур данных для представления графа может быть найдено в некоторых из ссылок, перечисленных в конце главы.

Существуют алгоритмы, которые трудны для построения, но доказательства корректности которых тривиальны. Другие алгоритмы легко представить, но доказательства их корректности весьма сложны. Более того, существуют непроанализированные тривиальные алгоритмы. Примеры таких алгоритмов имеются в разделах этой и следующей глав.

С целью последующего обсуждения разделим алгоритмы на две группы: алгоритмы анализа графов и алгоритмы решения задач оптимизации на графах. В этой главе мы исследуем алгоритмы первой группы. Вообще говоря, в этой главе обсуждаются алгоритмы, относящиеся к следующим основным темам:

1. Транзитивное замыкание.
2. Транзитивная ориентация.
3. Поиск в глубину.
4. Двусвязность и сильная связность.
5. Графы программ.

14.1. Транзитивное замыкание

Напомним (разд. 5.2), что бинарное отношение на некотором множестве есть набор упорядоченных пар элементов этого множества. Транзитивное замыкание бинарного отношения R — это отношение R^*, определенное следующим образом:

xR^*y тогда и только тогда, когда существует такая последовательность $x_0=x, x_1, x_2, \ldots, x_k=y$, что $k>0$ и $x_0Rx_1, x_1Rx_2, \ldots$ $\ldots, x_{k-1}Rx_k$. Ясно, что если xRy, то xR^*y. Следовательно, $R\subseteq R^*$. Можно легко показать, что R^* — транзитивно. В самом деле оно является наименьшим транзитивным отношением, включающим R. Таким образом, если R транзитивно, то $R^*=R$.

Как уже отмечалось в разд. 5.2, бинарное отношение можно представить ориентированным графом. Пусть G — ориентированный граф, представляющий отношение R. Ориентированный граф G^*, представляющий транзитивное замыкание R^* отношения R, называется транзитивным замыканием графа G. Из определения R^* следует, что ребро (x, y), $x\neq y$, принадлежит G^* тогда и только тогда,

когда в графе G существует ориентированный путь из вершины x в вершину y. Аналогично петля (x, x) принадлежит графу G^* тогда и только тогда, когда в графе G присутствует ориентированный цикл, включающий вершину x. Например, граф, изображенный на рис. 14.1, *б* является транзитивным замыканием графа на рис. 14.1, *а*.

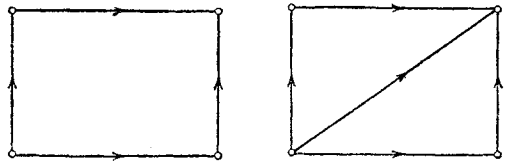


Рис. 14.1.
а — граф G; *б* — G^* — транзитивное замыкание графа G.

Предположим, что мы определяем *матрицу достижимости* ориентированного графа G на n вершинах как $n \times n$ (0,1)-матрицу, в которой элемент (i, j) равен 1 тогда и только тогда, когда существует ориентированный путь из вершины i в вершину j при $i \neq j$ или ориентированный цикл, содержащий вершину i при $i=j$. Другими словами, элемент (i, j) матрицы достижимости равен 1 тогда и только тогда, когда вершина j достижима из вершины i через последовательность ориентированных ребер. Очевидно, что матрица смежности графа G^* является в то же время матрицей достижимости графа G. Задача построения транзитивного замыкания ориентированного графа возникает в нескольких приложениях. Один из примеров описан Грисом [14.1]. В этом разделе мы обсуждаем элегантный и эффективный алгоритм вычисления транзитивного замыкания, предложенный Уоршоллом [14.2]. Мы обсудим также модификацию алгоритма Уоршолла, предложенную Уорреном [14.3].

Пусть G — ориентированный граф на n вершинах, которые обозначены целыми числами 1, 2, . . ., n. Пусть $G^0=G$. Алгоритм Уоршолла строит последовательность таких графов, что $G^i \subseteq G^{i+1}$, $0 \leqslant i \leqslant n-1$, и G^n — транзитивное замыкание графа G. Граф G^i $(i \geqslant 1)$ получается из графа G^{i-1} после обработки вершины i в графе G^{i-1}. Обработка вершины i в графе G^{i-1} включает добавление новых ребер в G^{i-1} следующим образом.

Пусть в графе G^{i-1} присутствуют ребра (i, k), (i, l), (i, m), ..., исходящие из вершины i. Тогда для каждого ребра (j, i), заходящего в вершину i, добавляем в графе G^{i-1} ребра (j, k), (j, l), (j, m), . . ., если эти ребра еще не присутствуют в графе G^{i-1}. Граф, получаемый после обработки вершины i, обозначается через G^i.

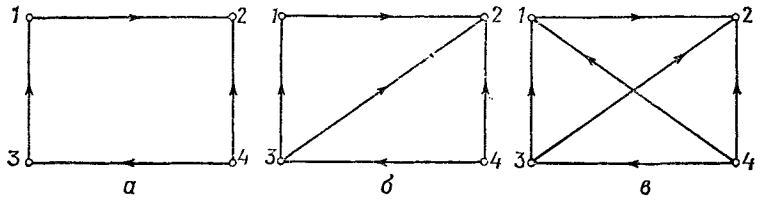


Рис. 14.2. Иллюстрация к алгоритму Уоршолла.
а — G^0; *б* — $G^1=G^2$; *в* — $G^3=G^4$.

Алгоритм Уоршолла представлен на рис. 14.2. Ясно, что $G^i \subseteq G^{i+1}$ при $i \geqslant 0$. Чтобы показать, что G^n — транзитивное замыкание графа G, необходимо доказать следующее:

Теорема 14.1. 1. Пусть для произвольных вершин s и t графа G существует такой ориентированный путь P из вершины s в вершину t, что все вершины, отличные от s и t, принадлежат множеству $\{1, 2, \ldots, i\}$. Тогда граф G^i содержит ребро (s, t).

2. Пусть для произвольной вершины s в графе G существует ориентированный цикл C, содержащий вершину s, и все вершины, отличные от s, принадлежат множеству $\{1, 2, \ldots, i\}$. Тогда граф G^i содержит петлю (s, s).

Доказательство. 1. Докажем индукцией по i. Ясно, что утверждение верно для графа G^1, поскольку в соответствии с построением Уоршолла при обработке вершины 1 вводится ребро (s, t), если граф G^0 (равный G) содержит ребра $(s, 1)$ и $(1, t)$. Пусть утверждение верно для всех G^k, $k<i$.

Предположим, что i не является внутренней вершиной пути P. Тогда из предположения индукции следует, что граф G^{i-1} содержит ребро (s, t). Следовательно, G^i также содержит ребро (s, t), потому что $G^{i-1} \leqslant G^i$. Предположим, что i — внутренняя вершина пути P. Тогда вновь из предположения индукции следует, что граф G^{i-1} содержит ребра (s, i) и (i, t). Поэтому при обработке вершины i в G^{i-1}) ребро (s, t) добавляется к графу G^i.

2. Доказательство аналогично доказательству п. 1.

Непосредственным следствием из этой теоремы является следующее.

Следствие 14.1.1. G^n — транзитивное замыкание графа G. Дадим формальное описание алгоритма Уоршолла. В этом описании граф G представляется матрицей смежности M, а символ V применяется для обозначения булева сложения.

Алгоритм 14.1. Транзитивное замыкание (Уоршолл).

S1. (Инициализация.) M — матрица смежности графа G.

S2. Выполнить S3 для $i=1, 2, \ldots, n$.

S3. Выполнить S4 для $j=1, 2, \ldots, n$.

S4. Если $M(j, i)=1$, выполнить S5 для $k=1, 2, \ldots, n$.

S5. $M(j, k)=M(j, k)\ VM(i, k)$.

S6. HALT. (M — матрица смежности графа G^*.)

Отметим, что M (когда алгоритм приступает к выполнению шага S3 для $i=p$) — матрица смежности графа G^{p-1}. Обработка диагональных элементов не приводит к добавлению новых ненулевых элементов. Сделаем несколько замечаний:

1. Алгоритм Уоршолла преобразует матрицу смежности M графа G в матрицу смежности его транзитивного замыкания соответствующим переписыванием матрицы M. Поэтому говорят, что алгоритм работает «на месте».
2. Алгоритм обрабатывает все ребра, заходящие в вершину, перед тем, как начинает обрабатывать следующую вершину. Другими словами, он обрабатывает матрицу M по столбцам. Следовательно, мы описываем алгоритм Уоршолла как *ориентированный по столбцам.*

3. При обработке вершины никакие новые ребра (т. е. ребра, которые не присутствовали в графе, когда началась обработка этой вершины), заходящие в эту вершину, не добавляются. Это означает, что при обработке вершины мы можем выбирать ребра, заходящие в вершину в произвольном порядке.
4. Предположим, что ребро (j, i), заходящее в вершину i, не присутствует в графе при обработке вершины i, но добавляется в последующем при обработке некоторой вершины $k\,(k>i)$. Ясно, что это ребро не рассматривается при обработке вершины i. Не будет оно обрабатываться и позднее, так как никакая вершина не обрабатывается более одного раза. Фактически такое ребро не приведет к добавлению каких-либо новых ребер.
5. Алгоритм Уоршолла называется работающим в один проход, так как каждая вершина обрабатывается точно один раз.

Предположим, что мы желаем модифицировать алгоритм Уоршолла так, чтобы он стал *ориентированным по строкам*. В таком алгоритме при обработке вершины обрабатываются все ребра, исходящие из этой вершины. Обработка ребра (i, j) заключается во введении ребра (i, k) для каждого ребра (j, k), исходящего из вершины j. Поэтому новые ребра, исходящие из вершины, можно добавить при обработке вершины «построчно». Некоторые из этих вновь добавленных ребер не будут обрабатываться до завершения обработки рассматриваемой вершины. Если обработка этих ребер необходима для вычисления транзитивного замыкания, то такая обработка может выполняться только при втором проходе. Таким образом, в общем случае ориентированный по строкам алгоритм может потребовать более одного прохода для вычисления транзитивного замыкания.

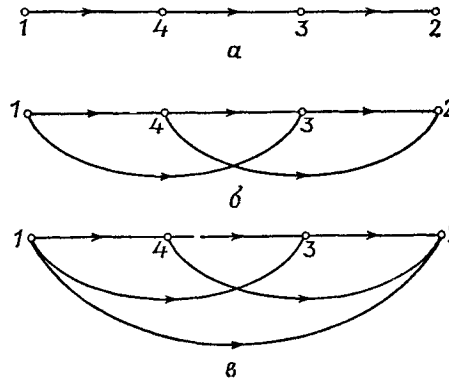


Рис. 14.3. Пример работы строкоориентированного алгоритма транзитивного замыкания.
а — граф G; *б* — граф G'; *в* — граф G^*.

Например, рассмотрим граф G на рис. 14.3, *а*. После обработки построчно вершин графа G мы получим граф G', представленный на рис. 14.3, *б*. Очевидно, что G' не является транзитивным замыканием графа G, так как ребро (1, 2) еще не добавлено. Можно отметить, что ребро (1, 3) не обрабатывается при этом проходе, поскольку оно добавляется только после обработки ребра (1, 4). То же самое верно и для ребра (4.2).

Предположим, что мы обрабатываем вершины графа G'. При этом втором проходе ребро (1, 2) добавляется при обработке вершины 1 и мы получаем транзитивное замыкание графа G^*, показанное на рис. 14.3, *в*. Таким образом, для графа, изображенного

на рис. 14.3, *а*, требуется два прохода ориентированного по строкам алгоритма.

Возникает вопрос: всегда ли достаточно двух проходов? Ответ на него положителен и Уоррен [14.3] продемонстрировал это, предложив изящный двухпроходной строкоориентированный алгоритм. В этом алгоритме при обработке вершины, например i, в первом проходе обрабатываются только ребра, связанные с вершинами, меньшими i, а во втором проходе — только ребра, связанные с вершинами, большими i. Другими словами, алгоритм преобразует матрицу смежности M графа G в матрицу смежности графа G^*, обрабатывая в первом проходе только элементы матрицы, расположенные ниже ее главной диагонали, а во втором проходе — только элементы матрицы, расположенные выше ее главной диагонали. Таким образом, при каждом проходе обрабатывается не более $n(n-1)/2$ ребер. Приведем описание модификации Уоррена, касающееся алгоритма Уоршолла.

Алгоритм 14.2. Транзитивное замыкание (Уоррен).
S1. M — матрица смежности графа G.
S2. Выполнить S3 для $i=2, 3, \ldots, n$.
S3. Выполнить S4 для $j=1, 2, \ldots, i-1$.
S4. Если $M(i, j)=1$, то выполнить S5 для $k=1, 2, \ldots, n$
S5. $M(i, k)=M(i, k) \vee M(j, k)$.
S6. Выполнить S7 для $i=1, 2, \ldots, n-1$.
S7. Выполнить S8 для $j=i+1, i+2, \ldots, n$.
S8. Если $M(i, j)=1$, то выполнить S9 для $k=1, 2, \ldots, n$.
S9. $M(i, k)=M(i, k) \vee M(j, k)$.
S10. HALT. (M — матрица смежности графа G^*.) Отметим, что в этом алгоритме шаги S2—S5 соответствуют первому проходу, а шаги S6—S9 — второму проходу.

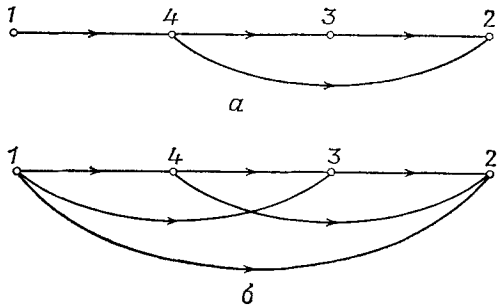


Рис. 14.4. Иллюстрация к алгоритму Уоррена.

Рассмотрим вновь граф, изображенный на рис. 14.3, *а*. В конце первого прохода алгоритма Уоррена мы получим граф, изображенный на рис. 14.4, *а*, а в конце второго — транзитивное замыкание графа G^*, представленное на рис. 14.4, *б*. Доказательство корректности алгоритма Уоррена основывается на следующей лемме:

Лемма 14.1. Предположим, что для двух произвольных вершин s и t в графе G существует ориентированный путь P из s в t. Тогда граф, полученный после обра-

ботки вершины s в первом проходе (шаги S2 — S5) алгоритма Уоррена, содержит ребро (s, r), где r — следующая за s вершина пути P причем $r>s$ или $r=t$.

Доказательство. Докажем индукцией по s.

Если $S=1$, то лемма справедлива, так как все следующие за 1 вершины пути P больше 1. Пусть лемма верна для всех $s<k$, и пусть $s=k$. Предположим, что (s, i) — первая дуга пути P. Если $i_1>s$, то ясно, что лемма верна. Если $i_1<s$, то по предположению индукции граф, получаемый после обработки вершины i_1 в первом проходе, содержит ребро (i_1, i_2), где i_2— следующая за i_1 вершина пути P, причем $i_2>i_1$ или $i_2=t$.

Если $i_2\neq t$ и $i_2<s$, то вновь по предположению индукции граф, получаемый после обработки вершины i_2 в первом проходе, содержит ребро (i_2, i_3), где i_3— следующая за i_2 вершина пути P, причем $i_3>i_2$ или $i_3=t$.

Если $i_3\neq t$ и $i_3<s$, то мы повторяем эти рассуждения для вершины i_3 до тех пор, пока мы не получим такую вершину i_m, что либо $i_m>s$, либо $i_m=t$. Таким образом, граф, который мы получаем перед обработкой вершины s, содержит такие ребра (s, i_1), (i_1, i_2), (i_2, i_3), . . ., (i_{m-1}, i_m), что при этом выполняются следующие соотношения:

1) вершина i_p следует за вершиной i_{p-1} в пути P, $p\geqslant 2$;
2) $i_{m-1}>i_{m-2}>i_{m-3}> \ldots >i_1$ и $i_k<s$ для $k\neq m$;
3) $i_m=t$ или $i_m>s$.

Приступаем к обработке вершины s. При обработке ребра (s, i_1) вводится ребро (s, i_2), поскольку присутствует ребро (i_1, i_2). Так как $i_2>i_1$, то в дальнейшем ребро (s, i_2) также обрабатывается. При обработке этого ребра вводится ребро (s, i_3), так как присутствует ребро (i_2, i_3) и т. д. Таким образом, когда обработка вершины s завершается, требуемое ребро (s, i_m) присутствует в конечном графе.

Теорема 14.2. Алгоритм Уоррена вычисляет транзитивное замыкание графа G.

Доказательство. Необходимо рассмотреть два случая:

Случай 1. Для любых двух вершин s и t в графе G существует ориентированный путь P из вершины s в вершину t. Пусть (i, j) — такое первое ребро пути P (в направлении от s к t), что $i\geqslant j$. Тогда из предыдущей леммы следует, что граф, который мы имеем перед началом второго прохода алгоритма Уоррена, содержит ребро (i, k), где k следует за i в P, причем $k=t$ или $k>i$. Таким образом, после завершения первого прохода существует такой путь P' : $S, i_1, i_2, \ldots$ . . ., i_m, t, что $s<i_1<i_2<\ldots<i_m$ и каждая вершина i_{j+1} следует за i_j в пути P.

При обработке вершины s во втором проходе ребро (s, i_1) встречается первым. Обработка этого ребра вызывает появление ребра (s, i_2), так как в графе присутствует ребро (i_1, i_2). Поскольку $i_2>i_1$, то ребро (s, i_2) в последующем подвергается обработке. Это в свою очередь приводит к введению ребра (s, i_3) и т. д. Таким образом, когда обработка вершины завершается, в полученном графе будет присутствовать ребро (s, t).

Случай 2. В графе G существует ориентированный цикл, содержащий вершину s. В этом случае мы можем привести доказательство, аналогичное приведенному выше, суть которого заключается в том, что после завершения обработки вершины s во втором проходе полученный граф будет содержать петлю (s, s).

Ясно, что оба алгоритма Уоршолла и Уоррена имеют в худшем случае сложность $O(n^3)$. Однако алгоритм Уоррена будет выполняться быстрее алгоритма Уоршолла в случае больших разряженных матриц, особенно в страничной среде. Уоррен [14.3] ссылается на другие строкоориентированные алгоритмы.

В работе [14.4] представлен $O(n^3/\log n)$-алгоритм. Он основан на алгоритме «четырех русских» для перемножения булевых матриц.

Автор работы [14.5] предложил $On(^{\log_2 7})$ -алгоритм для перемножения двух $n\times n$-матриц. Используя этот алгоритм, авторы работ [14.6, 14.7] представили $O(n^{\log_2 7}\log n)$-алгоритмы для задачи транзитивного замыкания. Другие алгоритмы, основанные на перемножении матриц, предложены в работах [14,8, 14.9]

В работе [14.10] рассматривается алгоритм, основанный на алгоритме Тарьяна для нахождения сильно связных компонент ориентированного графа

(разд. 14.4). Сильно связные компоненты данного графа определяет в качестве первого шага алгоритм Мунро.

Алгоритм Пардома [14.11] во многих случаях требует $O(n^2)$ шагов, хотя в худшем случае имеет сложность $O(n^3)$. Однако этот алгоритм гораздо сложнее других алгоритмов транзитивного замыкания.

Недавно Шнорр [14.12] представил алгоритм с предполагаемым временем работы $O(n+m^*)$, где m^* — ожидаемое число ребер в транзитивном замыкании.

В работе [14.13] рассматриваются вычислительные эксперименты с несколькими алгоритмами транзитивного замыкания.

14.2. Транзитивная ориентация

Неориентированный граф G *транзитивно ориентируем*, если мы можем ориентировать ребра графа G так, что полученный ориентированный граф транзитивен. Если граф G транзитивно ориентируем, то *транзитивную ориентацию* графа G мы будем обозначать $\vec{G}$.

Например, граф, изображенный на рис. 14.5, *а*, транзитивно ориентируем. Транзитивная ориентация этого графа представлена на рис. 14.5, *б*.

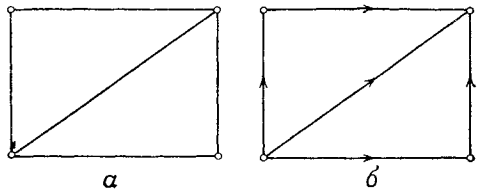


Рис. 14.5.

а — граф G; *б* — транзитивная ориентация графа G.

В этом разделе мы обсудим алгоритм Пнуели, Лемпеля и Ивена [14.14], осуществляющий проверку, является ли простой неориентированный граф транзитивно ориентируемым и получающим транзитивную ориентацию $\vec{G}$, если она существует. Для изложения этого алгоритма мы введем некоторые обозначения:

$i \rightarrow j$ означает, что вершины i и j связаны ребром, ориентированным из i в j;

$i \leftarrow j$ определяется аналогично;

$i—j$ означает, что существует ребро, связывающее вершины i и j;

$i \not\sim j$ означает, что не существует ребра, связывающего вершины i и j.

$i \not\rightarrow j$ означает, что либо $i \not\sim j$, либо $i \leftarrow j$, либо ребро $i—j$ не ориентировано.

$i \rightarrow j$, $i \leftarrow j$, $i—j$ и т. д. будут использоваться для обозначения соответствующих ребер.

Рассмотрим неориентированный граф $G=(V, E)$, который транзитивно ориентируем. Пусть $\vec{G}=(V, \vec{E})$ — транзитивная ориентация графа G.

Предположим, что существуют такие три вершины $i, j, k \in V$, что $i \to j$, j—k и $i \not\to k$; тогда из транзитивности $\vec{G}$ должно следовать $j \leftarrow k$. Аналогично если, для $i, j, k \in V$, $i \to j$, i—k и $j \not\to k$, то из транзитивности $\vec{G}$ должно следовать $i \to k$.

Эти замечания приводят к следующим простым правилам, которые образуют основу алгоритма Пнуели, Лемпеля и Ивена.

Правило R_1. При $i, j, k \in V$, если $i \to j$, j — k и $i \not\to k$, то ориентируем ребро j — k как $j \leftarrow k$.

Правило R_2. При $i, j, k \in V$, если $i \to j$, i — k и $j \not\to k$, то ориентируем ребро i — k как $i \to k$.

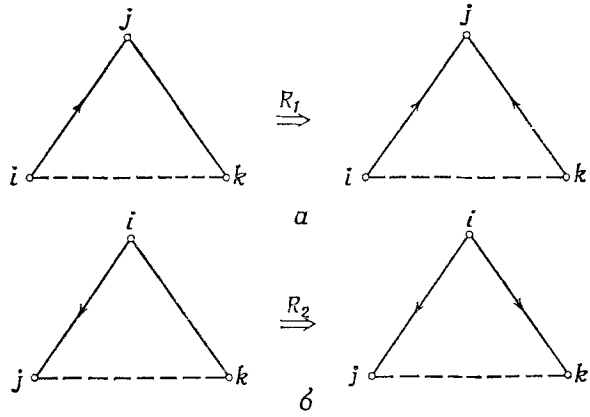


Рис. 14.6.
а — правило R_1; *б* — R_2.

Эти два правила иллюстрируются рис. 14.6, на котором штриховые линии указывают на отсутствие соответствующего ребра.

Представим описание алгоритма транзитивной ориентации графа.

Алгоритм 14.3. Транзитивная ориентация (Пнуели, Лемпеля и Ивена).

S1. Пусть дан простой неориентированный граф G. Положить $i=1$.

S2. (Начало фазы i.) Выбрать ребро e графа G и ориентировать его произвольным образом. Ориентировать, если возможно, ребра графа G, смежные с e, используя правило R_1 или R_2. Ориентированное ребро e пометить как «рассмотренное».

S3. Проверить, существует ли в графе G ориентированное ребро, которое не помечено как «рассмотренное»; если да, то идти к шагу S4. Иначе идти к шагу S6.

S4. Пусть $i \to j$ — ориентированное ребро в графе G, которое не помечено как «рассмотренное». Тогда для каждого ребра в графе G, инцидентного i или j, выполнить следующие действия (всякий раз, когда они применимы) и пометить $i \to j$ как «рассмотренное»:

Случай 1. Пусть j — k — рассматриваемое ребро.

а. (Применение правила R_1.) Если $i \not\to k$ и ребро j—k не ориентировано, то ориентировать j—k как $j \leftarrow k$.

б. (Противоречие с правилом R_1.) Если $i \not\to k$ и ребро j—k уже ориентировано как $j \to k$, то возникло противоречие с правилом R_1. Идти к шагу S9.

Случай 2. Пусть рассматриваемое ребро будет i—k

а. (Применение правила R_2.) Если $j \not\to k$ и ребро i—k не ориентировано, то ориентировать i—k как $i \to k$.

б. (Противоречие с правилом R_2.) Если $j \not\to k$ и ребро i—k уже ориентировано как $i \leftarrow k$ то возникло противоречие с правилом R_2. Идти к шагу S9.

S5. Идти к шагу S3.

S6. (Успешное окончание фазы i.) Проверить, все ли ребра графа G уже ориентированы. Если да, то идти к шагу S8. Иначе удалить из графа G все ориентированные ребра. Пусть G' — полученный в результате граф.

S7. Положить $G=G'$ и $i=i+1$. Идти к шагу S2.

S8. (Все ребра данного графа ориентированы в соответствии с правилами R_1 и R_2. Это — транзитивная ориентация данного графа.) HALT.

S9. (Граф G не является транзитивно ориентируемым.) HALT.

Основная обработка в этом алгоритме производится на шаге S4. На нем мы проверяем каждое ребро, смежное с ориентированным ребром $i \to j$. Если это ребро уже ориентировано, то мы проверяем, согласуется ли его ориентация и ориентация ребра $i \to j$ в соответствии с правилом R_1 или R_2. Если рассматриваемое ребро еще не ориентировано, то мы ориентируем его, если это возможно, используя правило R_1 или R_2.

Алгоритм состоит из различных фаз. Каждая фаза включает выполнение шага S2 и повторяет выполнение шага S4 столько раз, сколько ребер получает ориентацию. Если фаза заканчивается без обнаружения какого-либо противоречия с правилом R_1 или R_2, то это означает, что больше ни одно из ребер нельзя ориентировать в этой фазе с помощью этих двух правил и что ориентация, полученная в этой фазе, согласуется с используемыми правилами.

Алгоритм заканчивается, 1) когда обнаруживается противоречие с правилом R_1 или R_2 либо 2) когда ориентированы все ребра данного графа, так что ориентация согласуется с правилами R_1 и R_2. В первом случае граф не является транзитивно ориентируемым, а во втором — граф транзитивно ориентируем, и полученный ориентированный граф определяет его транзитивную ориентацию.

Сложность алгоритма зависит от сложности выполнения шага S4. Этот шаг выполняется не более m раз, где m — число ребер в данном графе. Каждое выполнение шага S4 включает в себя проверку всех ребер, смежных с ориентированным ребром. Поэтому число операций, необходимых для выполнения шага S4, пропорционально 2Δ, где Δ — максимальная степень данного графа. Таким образом, общая сложность этого алгоритма есть $O(2m\Delta)$.

Проиллюстрируем алгоритм двумя примерами. Сперва рассмотрим граф, представленный на рис. 14.7, *а*.

Фаза 1. Мы начинаем, ориентируя ребро 7—2 как 7→2. По правилу R_2 7→2 влечет ориентацию 7→4 и 7→5, а 7→4 — ориентацию 7→6. Правило R_1 не применимо ни к одному из ребер, смеж-

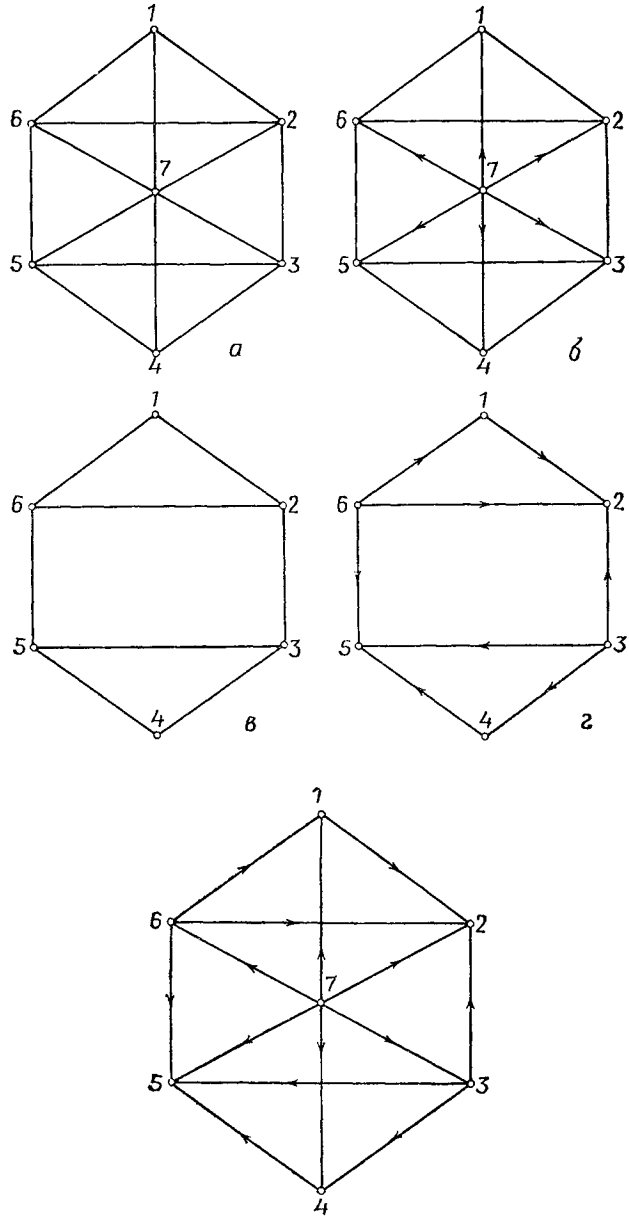


Рис. 14.7.

ных с ребром 7→2. По правилу R_2 7→5 и 7→6 влекут ориентации 7→1 и 7→3 соответственно. В этой фазе нельзя ориентировать никакие другие ребра. Мы можем убедиться, что все полученные ориентации ребер согласуются с правилами R_1 и R_2. Фаза 1 на этом заканчивается. Ребра, ориентированные в этой фазе, показаны на рис. 14.7, *б*.

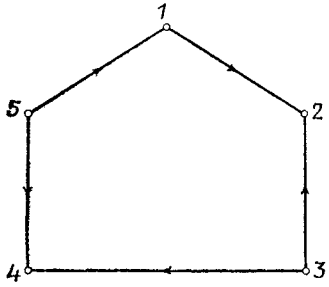


Рис. 14.8.

Удалим из графа G (рис. 14.7, *а*) все ребра, которые ориентированы в фазе 1. Полученный граф G' представлен на рис. 14.7, *в*. С этого момента начинается фаза 2, на которой рассматривается граф G'.

Фаза 2. Начнем с ориентации ребра 1—2 как 1→2. Это приводит к следующей последовательности ориентаций:

$$(1\to 2) \underset{R_1}{\Rightarrow} (2\leftarrow 3) \underset{R_2}{\Rightarrow} (3\to 5) \underset{R_1}{\Rightarrow} (6\to 5) \underset{R_1}{\Rightarrow} (4\to 5)$$
$$(2\leftarrow 3) \underset{R_2}{\Rightarrow} (3\to 4)$$
$$(6\to 5) \underset{R_2}{\Rightarrow} (6\to 1)$$
$$(6\to 5) \underset{R_2}{\Rightarrow} (6\to 2)$$

Таким образом, все ребра графа G' ориентированы, как показано на рис. 14.7, *г*. Эта ориентация графа G' согласована с правилами R_1 и R_2. Поэтому фаза 2 успешно завершается. Полученная транзитивная ориентация графа G представлена на рис. 14.7, *д*.

Рассмотрим другой граф, показанный на рис. 14.8. Начнем с ориентации ребра 1—2 как 1→2. Это ведет к следующей последовательности ориентаций:

$$(1\to 2) \underset{R_1}{\Rightarrow} (2\leftarrow 3) \underset{R_2}{\Rightarrow} (3\to 4) \underset{R_1}{\Rightarrow} (4\leftarrow 5) \underset{R_2}{\Rightarrow} (5\to 1) \underset{R_1}{\Rightarrow} (1\leftarrow 2),$$

которая требует, чтобы ребро 1—2 было ориентировано как 1←2, что противоречит ориентации, которая уже сопоставлена ребру 1—2. Таким образом, получено противоречие с правилом R_1. Следовательно, граф на рис. 14.8 не является транзитивно ориентируемым.

Теперь перейдем к доказательству корректности алгоритма 14.3. Для выполнения этого необходимо доказать два следующих утверждения:

Утверждение 1. Если алгоритм 14.3 завершается успешно (шаг S8), то полученный ориентированный граф является транзитивной ориентацией исходного графа.

Утверждение 2. Если граф транзитивно ориентируем, то алгоритм 14.3 завершается успешно.

Рассмотрим сначала утверждение 1.

Дан неориентированный граф $G=(V, E)$. Предположим, что алгоритм завершается успешно. Рассмотрим произвольную пару ребер $i \to j$ и $k \to l$, которые ориентированы на одной и той же фазе алгоритма. Тогда мы можем построить последовательность выводов, которая начинается с ребра $i \to j$ и заканчивается ориентацией ребра $k - l$ как $k \to l$. Такую последовательность мы будем называть *цепью вывода* из $i \to j$ к $k \to l$. Заметим, что может быть несколько цепей вывода из $i \to j$ к $k \to l$. Например, в ориентированном графе на рис. 14.7, *г* цепи вывода из $(2 \leftarrow 6)$ к $(3 \to 4)$ могут иметь вид

$$(2 \leftarrow 6) \Rightarrow (3 \to 2) \Rightarrow (3 \to 4),$$
$$(2 \leftarrow 6) \Rightarrow (5 \leftarrow 6) \Rightarrow (3 \to 5) \Rightarrow (3 \to 2) \Rightarrow (3 \to 4).$$

Таким образом, ясно, что можно говорить о кратчайшей цепи вывода между любой парой ориентированных ребер, которые были ориентированы на одной и той же фазе алгоритма 14.3. Доказательство утверждения 1 основывается на следующей лемме:

Лемма 14.2. После успешного завершения фазы 1 в графе G не может быть таких трех вершин i, j, k, что $i \to j$, $j \to k$ и $i \not\to k$.

Доказательство. Напомним, что $i \not\to k$ означает, что либо $i \not\leftarrow k$, либо $i \leftarrow k$, либо ребро $i—k$ не ориентировано. Ясно, что должно существовать ребро, связывающее вершины i и k. Иначе мы получили бы противоречие, так как по правилу R_1 $i \to j$ влечет $j \leftarrow k$.

Предположим, что запрещенная ситуация типа $i \to j$ и $j \to k$ с $i \not\leftarrow k$ возникает после завершения фазы 1 алгоритма 14.3. Тогда выберем среди всех цепей вывода, которые приводят к запрещенной ситуации, цепь, кратчайшую с минимальным числом ориентированных ребер, заходящих в k. Ясно, что любая такая цепь должна иметь длину не менее трех. Пусть одна из таких цепей имеет вид

$$(i \to j) \Rightarrow (\alpha_1) \Rightarrow (\alpha_2) \Rightarrow \cdots \Rightarrow (\alpha_{p-1}) \Rightarrow (j \to k).$$

При этом α_{p-1} должно быть либо $j \to j'$ для некоторого j', либо $k' \to k$ для некоторого k. Таким образом, необходимо рассмотреть два случая.

Случай 1. Пусть α_{p-1} будет $j \to j'$. Тогда вывод $(\alpha_{p-1}) \Rightarrow (j \to k)$ требует, чтобы $j' \not\to k$. Далее, $i \to j'$, так как иначе цепь вывода

$$(i \to j) \Rightarrow (\alpha_1) \Rightarrow \cdots \Rightarrow (\alpha_{p-2}) \Rightarrow (j \to j')$$

приводила бы к запрещенной ситуации $i \to j$ и $j \to j'$ с $i \not\to j'$. Однако эта цепь короче исходной, что противоречит предположению о ее минимальности.

Случай, иллюстрирующий эти рассуждения, представлен на рис. 14.9, *а*, на котором штриховая линия указывает на отсутствие соответствующего ребра. Таким образом, $i \to j'$ и $j' \not\to k$ влечет за собой $i \to k$, согласно правилу R_2. Однако это противоречит нашему предположению, что $i \not\to k$.

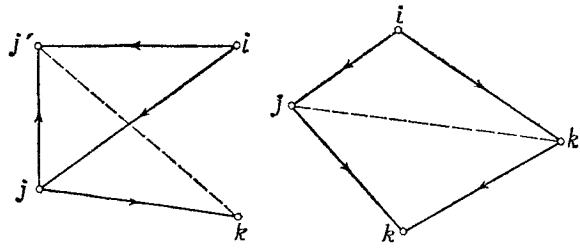


Рис. 14.9.

Случай 2. Пусть α_{p-1} будет $k' \to k$. Как и в предыдущем случае, вывод $(\alpha_{p-1}) \Rightarrow \Rightarrow (i \to k)$ требует, чтобы $j \not\to k'$. Далее $i \to k'$, так как иначе $k' \to k$ влекло бы $i \to k$, что противоречит предположению, что $i \not\to k$. Кроме того, из $i \to y$ и $j \not\to k'$ следует, что $i \to k'$.

Случай, иллюстрирующий эти рассуждения, представлен на рис. 14.9, *б*. Тогда цепь вывода

$$(i \to k') \Rightarrow (i \to j) \Rightarrow (\alpha_1) \Rightarrow \cdots \Rightarrow (\alpha_{p-2}) \Rightarrow (k' \to k),$$

ведет к запрещенной ситуации $i \to k'$ и $k' \to k$ с $i \not\to k$ имеет ту же длину, что и исходная цепь, но включает меньше ребер, входящих в k, на единицу. Это вновь противоречит предположению, которое было сделано относительно выбора исходной цепи вывода.

Пусть E' — множество ребер, которые ориентируются на первой фазе, и пусть $\vec{E}'$ — соответствующее множество ориентированных ребер. Следующий результат является прямым следствием леммы 14.2.

Теорема 14.3. Подграф $\vec{G}'(V, \vec{E}')$ является транзитивным.

Теорема 14.4. Если алгоритм 14.3 завершается успешно, то он обеспечивает транзитивную ориентацию графа.

Доказательство. Проводится индукцией по числу фаз в алгоритме. Если алгоритм ориентирует все ребра данного графа в фазе 1, то, согласно теореме 14.3, получаемая ориентация транзитивна.

Пусть данный граф $G=(V, E)$ ориентируется за p фаз. Пусть E' — множество ребер, которые получают ориентацию на первой фазе. Тогда, согласно теореме 14.3, подграф $\vec{G}=(V, \vec{E}')$ транзитивен. Далее, так как подграф $G''=(V, E-E')$ ориентируем за $p-1$ фаз, то из индуктивного предположения следует, что $\vec{G}''=(V, \vec{E}-\vec{E}')$ транзитивен. Теперь мы докажем, что ориентированный граф $\vec{G}=(V, \vec{E})$ транзитивен.

Предположим, что граф $\vec{G}$ не транзитивен, т. е. в нем существуют такие три вершины i, j, k, что $i \to j$ и $j \to k$ но $i \not\to k$. Тогда оба ребра $i \to j$ и $j \to k$ не могут принадлежать $\vec{E}'$ или $\vec{E}-\vec{E}'$, так как оба графа $\vec{G}'$ и $\vec{G}''$ транзитивны. Не нарушая общности, предположим, что $i \to j$ принадлежат $\vec{E}'$, а $j \to k$ — $\vec{E}-\vec{E}'$. Тогда должно существовать ребро $i-k$, соединяющее вершины i и k; иначе $i \to j$

влекло бы $j \leftarrow k$ по правилу R_1. Предположим, что ребро $i—k$ ориентировано как $i \leftarrow k$ на фазе 1. Тогда граф $\vec{G}'$ не транзитивен, и мы приходим к противоречию. С другой стороны, если оно ориентировано как $i \leftarrow k$ на более поздней фазе, то граф $\vec{G}''$ не транзитивен — вновь приходим к противоречию.

Таким образом, невозможно существование в графе G таких трех вершин i, j, k, что $i \rightarrow j$ и $j \rightarrow k$, но $i \not\rightarrow k$. Следовательно, граф $\vec{G}$ транзитивен. Утверждение 1 доказано.

Перейдем к доказательству утверждения 2. Рассмотрим граф $G=(V, E)$, который транзитивно ориентируем. Ясно, что если мы изменим ориентацию всех ребер в любой транзитивной ориентации графа G, то получаемый ориентированный граф также является транзитивной ориентацией графа G.

Выберем какое-либо ребро графа G и ориентируем его произвольным образом. Пусть это будет ребро $i \rightarrow j$. Теперь если мы приступим к ориентации других ребер, используя правила R_1 и R_2, то ребра, ориентированные таким образом, будут иметь ту же самую ориентацию во всех возможных транзитивных ориентациях графа, в которых ребро $i—j$ ориентировано как $i \rightarrow j$. Это верно потому, что если ориентация ребра $i—j$ определена, то ориентации, выведенные по правилам R_1 и R_2, обязательны для транзитивной ориентируемости. Поэтому отсюда вытекает, что если мы применим алгоритм 14.3 к транзитивно ориентируемому графу, то фаза 1 закончится успешно без появления противоречий с правилом R_1 или R_2. Ребра, ориентированные на первой фазе, будут иметь эту же ориентацию в некоторой транзитивной ориентации графа G.

Если бы мы смогли доказать, что граф $G''=(V, E—E')$, где E' — множество ребер ориентированных на первой фазе, также транзитивно ориентируем, то отсюда вытекало бы, что фаза 2, а также все последующие фазы закончатся успешно, получая транзитивную ориентацию графа G. Таким образом, доказательство утверждения 2 эквивалентно установлению транзитивной ориентируемости графа $G''=(V, E—E')$. Для выполнения этого поступим следующим образом.

Назовем ребра множества E' *помеченными ребрами*, а инцидентные им вершины — *помеченными вершинами*. Обозначим множество помеченных вершин через V'. Отметим, что непомеченные ребра могут быть инцидентны помеченным вершинам.

Лемма 14.3. В графе G не может быть трех таких помеченных вершин i, j, k, что ребра $i—j$ и $j—k$ не помечены, а ребро $i—k$ помечено.

Доказательство. Предположим, что запрещенная ситуация указанного в лемме типа существует. Другими словами, существует такая тройка помеченных вершин i, j, k, что ребра $i—j$ и $j—k$ не помечены, а ребро $i—k$ помечено. Для каждой такой тройки i, j, k существует ребро $j—l$ (для некоторого l), которое помечено, так как j — помеченная вершина. Следовательно, существует цепь вывода из помеченного ребра $i—k$ к помеченному ребру $j—l$,

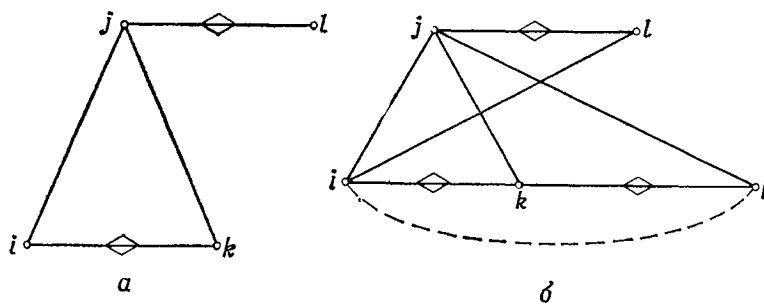


Рис. 14.10.

Выберем запрещенную ситуацию с такими помеченными вершинами i, j, k, что цепь вывода P из i—k и j—l является кратчайшей среди всех таких цепей, ведущих к запрещенным ситуациям. Такая конфигурация представлена на рис. 14.10, *а*, на котором ромб на ребре указывает, что это ребро помечено, а штриховая линия указывает, что соответствующее ребро отсутствует.

Следующее после i—k помеченное ребро в кратчайшей цепи P есть либо i—p, либо k—p для некоторого p. Мы, не нарушая общности, предположим, что это ребро есть k—p. Следовательно $i \not\rightarrow p$, так как иначе ребро k—p не было бы помечено из-за присутствия ребра i—k. Далее, существует ребро i—l, соединяющее i и l, так как иначе ребро i—j было бы помечено.

Таким образом, p и l различны. Ребро j—p также присутствует в графе, так как иначе ребро j—k было бы помечено.

Соотношения, установленные до этого момента, изображены на рис. 14.10, *б*. Ребро j—p не может быть помеченным, так как из этого следовало бы, что ребро i—j было бы помеченным. Таким образом, мы имеем более короткую цепь вывода из ребра k—p к ребру j—l, ведущую к другой запрещенной ситуации, когда ребро k—p помечено, а ребра k—j и j—p не помечены. Это противоречит допущению.

Теорема 14.5. Если граф $G=(V, E)$ транзитивно ориентируем, то граф $G''=(V, E-E')$ также транзитивно ориентируем.

Доказательство. Так как правила R_1 и R_2 помечают только смежные ребра, то граф $G'=(V', E')$ связен. Рассмотрим произвольную вершину $v \in V-V'$. Если v связана с какой-либо вершиной $v \in V'$, то она должна быть связана со всеми вершинами V', которые смежны с v', так как иначе ребро v—v' было бы помечено. Поскольку граф G' связен, то отсюда следует, что вершина v должна быть смежна со всеми вершинами в V.

Пусть $\vec{G}$ — транзитивная ориентация графа $\vec{G}$, в которой ориентация ребер из $\vec{E}'$ согласуется с ориентацией в графе $\vec{G}$.

Разобьем множество $V-V'$ на четыре подмножества следующим образом:

$$A=\{i \mid i \in V-V', \text{ и для всех } j \in V' \ i \longrightarrow j \text{ в } \vec{G}\},$$

$$B=\{i \mid i \in V-V', \text{ и для всех } j \in V' \ j \longrightarrow i \text{ в } \vec{G}\},$$

$$C=\{i \mid i \in V-V', \text{ и для всех } j \in V' \ i \not\rightarrow j \text{ в } \vec{G}\},$$

$$D=V-(V' \cup A \cup B \cup C).$$

Отметим, что D состоит из всех тех вершин множества $V-V'$, которые смежны со всеми вершинами множества V', но не все ребра, соединяющие вершину в D и вершины в V', ориентированы в одном и том же направлении.

Транзитивность $\vec{G}$ влечет следующие соотношения между различными подмножествами V:

1) для всех $i \in A$, $j \in D$, $k \in B$ $i \rightarrow j$, $j \rightarrow k$ и $i \rightarrow k$;

2) для всех $i \in C$ и $j \in D$ $i \not\rightarrow j$;

3) все ребра, связывающие A и C, ориентированы из A в C.

4) все ребра, связывающие B и C, ориентированы из C в B.

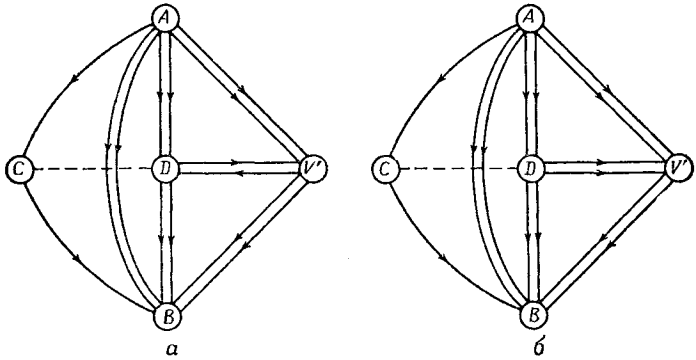
Эти соотношения представлены на рис. 14.11, *а*.

Изменим ориентацию всех ребер, ориентированных из V' в D, так что все ребра, связывающие V' и D, станут ориентированы из D в V'. Полученная ориентация показана на рис. 14.11, *б*. Мы утверждаем, что эта ориентация транзитивна. Для доказательства этого мы должны показать, что если в графе на рис. 14.11, *б* $i \rightarrow j$ и $j \rightarrow k$, то $i \rightarrow k$ для любых i, j и k. Ясно, что это верно, если ни у одного из этих ребер не менялась ориентация.

Таким образом, мы должны рассмотреть только четыре следующих случая: 1) $i \in D$, $j \in V'$, $k \in V'$; 2) $i \in D$, $j \in V'$, $k \in B$; 3) $j \in D$, $k \in V'$, $i \in A$; 4) $j \in D$, $k \in V'$, $i \in D$.

Во всех четырех случаях $i \to k$, как показано на рис. 14.11, *б*. Таким образом, ориентация, изображенная на рис. 14.11, *б*, транзитивна. Удалим из графа с рис. 14.11, *б* все ребра E', а именно все помеченные ребра.

Предположим, что в полученном графе существуют такие вершины i, j и k, что $i \to j$ и $j \to k$, но $i \not\to k$. Здесь $i \not\to k$ означает, что $i \not\to k$, так как из $i \leftarrow k$ вытекало бы, что ориентация, изображенная на рис. 14.11, *б*, была бы нетранзитивной. Если



а *б*

Рис. 14.11.

ребро i—k не принадлежит E—E', то оно должно принадлежать E', в противном случае ориентация, изображенная на рис. 14.11, *б*, вновь была бы нетранзитивной. Таким образом, i и k принадлежат V'.

Так как не существует вершины вне V', которая имела бы ребро, заходящее в нее из V', и ребро, исходящее из нее в V', то j также принадлежит V'.

Таким образом, мы имеем помеченные вершины i, j, k, непомеченные ребра i—j и j—k и помеченное ребро i—k. Это невозможно в силу леммы 14.3.

Поэтому ориентированный граф, получаемый после удаления ребер $\vec{E}'$ из графа на рис. 14.11, *б*, транзитивен. Следовательно, $G''=(V, E-E')$ транзитивно ориентируем. Таким образом мы установили верность утверждения 2 и, следовательно, корректность алгоритма 14.3.

Теперь должно быть ясно, что алгоритм транзитивной ориентации, рассмотренный выше, является примером алгоритма, который прост, но доказательство корректности которого очень запутанно. Более ранний алгоритм для решения этой задачи приведен в работе [14.15]. Авторы работ [14.14, 14,16] ввели понятие «графы перестановок» и установили структурное соответствие между данными и транзитивно ориентируемыми графами. Они рассмотрели также алгоритм, позволяющий проверить, является ли данный граф графом перестановок.

Некоторые задачи графов, которые трудноразрешимы в общем случае, становятся простыми при рассмотрении лишь транзитивно ориентируемых графов. Примерами таких задач могут служить задачи выделения максимальной клики и минимальной раскраски. Эти задачи возникают при изучении размещения памяти и решении задачи трассировки [14.16]. Изложение этих задач можно найти в работах [14.17, 14.18].

14.3. Поиск в глубину

В этом разделе мы опишем последовательный метод обхода графа. Этот метод, известный как *поиск в глубину* или, короче, ПВГ, оказался весьма полезным при построении ряда эффектив-

ных алгоритмов. Некоторые из них обсуждаются в остальных разделах этой главы. Изложение материала в данном разделе базируется на результатах работы [14.19].

14.3.1. ПВГ в неориентированном графе

Вначале рассмотрим ПВГ в неориентированном графе. Мы будем предполагать, что рассматриваемые графы связны. Если граф не связен, то ПВГ выполняется отдельно в каждой компоненте графа. Мы будем также предполагать, что в графе нет петель.

ПВГ в неориентированном графе выполняется следующим образом:

В графе G выбираем произвольную вершину, например v, и начинаем из нее поиск. Начальная вершина v, называемая *корнем ПВГ*, после этого считается *пройденной*.

Затем выбираем ребро (v, w), инцидентное вершине v, и проходим его, чтобы попасть в вершину w. Ориентируем при этом ребро из v в w. Ребро (v, w) после этих действий считается *просмотренным* и называется *ребром дерева*. Вершина v называется *отцом* вершины w и обозначается как FATHER (w).

В общем случае, когда мы находимся в какой-либо вершине x, возникают две возможности:

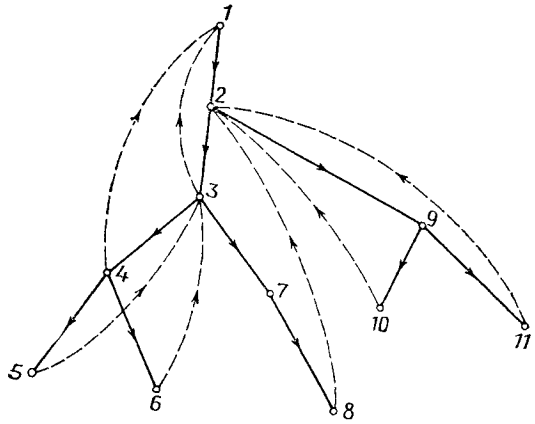
1. Если все ребра, инцидентные x, уже просмотрены, то мы возвращаемся к отцу x и продолжаем поиск из FATHER (x). Вершина x с этого момента называется *полностью сканированной*.
2. Если существуют непросмотренные ребра, инцидентные x, то мы выбираем одно из таких ребер (x, y) и ориентируем его из x в y. Ребро (x, y) с этого момента считается просмотренным. Необходимо рассмотреть два случая:

Случай 1. Если y ранее не была пройдена, то мы проходим ребро (x, y), вершину y и продолжаем поиск из вершины y. В этом случае ребро (x, y) называется *ребром дерева* и x=FATHER (y).

Случай 2. Если y ранее была пройдена, то мы продолжаем поиск другого непросмотренного ребра, инцидентного x. В этом случае ребро (x, y) называется *обратным ребром*. Во время поиска в глубину, когда вершину x проходят в первый раз, ей сопоставляется такое целое число DFN (x), что DFN (x) равно i, если x является i-й по порядку прохождения вершиной. DFN (x) называется *глубиной* x. Ясно, что глубина указывает порядок, в котором проходят вершины при поиске в глубину.

ПВГ завершается, когда мы возвращаемся в корень и все вершины графа пройдены.

Из описания видно, что поиск в глубину разбивает ребра графа G на ребра дерева и обратные ребра. Легко показать, что ребра дерева образуют остов графа G. ПВГ вводит ориентацию на ребра графа G. Получаемый в результате ориентированный граф мы будем



Вершина	*Список смежности*
1	(1,2), (1,3), (1,4)
2	(2,1), (2,3), (2,8), (2,9), (2,10), (2,11)
3	(3,1), (3,2), (3,4), (3,5), (3,6), (3,7)
4	(4,1), (4,3), (4,5), (4,6)
5	(5,3), (5,4)
6	(6,3), (6,4)
7	(7,3), (7,8)
8	(8,2), (8,7)
9	(9,2), (9,10), (9,11)
10	(10,2), (10,9)
11	(11,2), (11,9)

Рис. 14.12. ПВГ в неориентированном графе.

обозначать через $\hat{G}$. Ребра дерева с ориентацией, налагаемой ПВГ, будут образовывать ориентированный остов $\hat{G}$. Этот ориентированный остов будем называть *деревом ПВГ*. Отметим, что способ прохождения графа не единствен, так как ребра, инцидентные вершине, могут выбираться для рассмотрения в произвольном порядке.

В качестве примера на рис. 14.12 показан ПВГ в неориентированном графе. На этом рисунке ребра дерева изображены сплошными, а обратные ребра — штриховыми линиями. Для каждой вершины приведен список инцидентных ей ребер. Такой список для вершины v называется *списком смежности* вершины v и определяет порядок, в котором ребра, инцидентные вершине v, выбираются для просмотра.

Представим формальное описание алгоритма ПВГ. При этом рассматриваемый граф не обязательно должен быть связным.

Массив MARK, используемый в алгоритме, имеет по одному элементу для каждой вершины. Сначала мы устанавливаем MARK $(v)=0$ для всех вершин графа, указывая тем самым, что ни одна вершина еще не пройдена. Когда вершина проходится, мы устанавливаем соответствующий элемент массива MARK равным 1. Массивы DFN и FATHER определены ранее. TREE и BACK — два множества, хранящие ребра дерева и обратные ребра соответственно по мере их выделения.

Алгоритм 14.4. ПВГ в неориентированном графе.

S1. G — данный граф. Пусть TREE$=\varnothing$, BACK$=\varnothing$ и $i=1$. Для каждой вершины v графа G установить FATHER $(v)=0$ и MARK $(v)=0$.

S2. (Начало ПВГ в компоненте графа G). Выбрать какую-либо вершину, например вершину r, для которой MARK $(r)=0$. Положить DFN $(r)=i$, MARK $(r)=1$ и $v=r$. (Вершина r называется *корнем* рассматриваемой компоненты.)

S3. Если все ребра, инцидентные вершине v, уже помечены как «просмотренные», то идти к шагу S5 (v — полностью сканирована). Иначе выбрать ребро (v, w), которое еще не помечено как «просмотренное», и идти к шагу S4.

S4. Ориентировать ребро (v, w) от v к w и пометить его как «просмотренное». Выполнить следующие действия и идти к шагу S3.
1. Если MARK $(w)=0$, то положить $i=i+1$, DFN $(w)=i$, TREE$=$TREE $\cup \{(v, w)\}$, MARK $(w)=1$, FATHER $(w)=v$, $v=w$.
2. Если MARK $(w)=1$, то положить BACK$=$BACK $\cup \{(v, w)\}$.

S5. Если FATHER $(v)\neq 0$ (т. е. v — не корень рассматриваемой компоненты), то положить $v=$FATHER (v) и идти к шагу S3. Иначе идти к шагу S6.

S6. Если для всех вершин x MARK $(x)=1$, то идти к шагу S7, иначе положить $i=i+1$ и идти к шагу S2.

S7. (ПВГ завершен) HALT.

Пусть T — дерево ПВГ связного неориентированного графа. Как мы упоминали ранее, T — ориентированный остов графа $\hat{G}$. Для дальнейшего обсуждения нам необходимо ввести некоторые термины.

Если в дереве T существует ориентированный путь из вершины v в вершину w, то v называют *предком* w, а w — *потомком* v. Более того, если $v\neq w$, то v называется *собственным предком* w, а w — *собственным потомком* v. Если (v, w) — ориентированное ребро T, то v — *отец* w, а w — *сын* v. Отметим, что вершина может иметь более одного сына. Вершина v и все ее потомки формируют поддерево дерева T с вершиной v в качестве корня этого поддерева.

Две вершины v и w являются *соотносимыми*, если одна из них — потомок другой. Иначе v и w являются *несоотносимыми*. Если v и w несоотносимы и DFN $(v)<$DFN (w), то мы будем говорить, что v находится *слева* от w, иначе v находится *справа* от w. Ребра графа G, связывающие несоотносимые вершины, называются *пересекающими ребрами*.

Покажем сейчас, что в графе G не существует пересекающих ребер.

Пусть v_1 и v_2 — какие-либо несоотносимые вершины дерева T. Тогда ясно, что существуют такие две вершины s_1 и s_2, что 1) FATHER (s_1)=FATHER (s_2) и 2) v_1 и v_2 — потомки s_1 и s_2 соответственно (рис. 14.13).

Пусть T_1 и T_2 — поддеревья дерева T с корнями s_1 и s_2 соответственно. Не нарушая общности, предположим, что DFN $(s_1)<$ $<$DFN (s_2). Тогда из алгоритма ПВГ следует, что вершины поддерева T_2 проходятся только после того, как вершина s_1 полностью сканирована и не может существовать ребра, связывающего v_1 и v_2. Если бы такое ребро существовало, то v_2 была бы пройдена до того, как было бы завершено сканирование s_1. Таким образом доказано следующее утверждение.

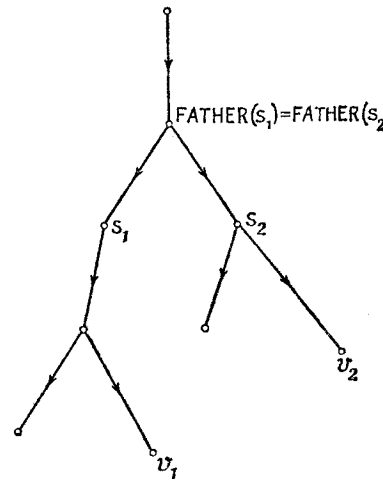


Рис. 14.13.

Теорема 14.6. Если (v, w) — ребро в связном неориентированном графе G, то в любом дереве ПВГ этого графа либо v — потомок w, либо наоборот.

Отсутствие пересекающих ребер в неориентированном графе является важным свойством, которое образует основу алгоритма, обсуждаемого в следующем разделе, для выделения двусвязных компонент графа.

14.3.2. ПВГ в ориентированном графе

Поиск в глубину в ориентированном графе в основном совпадает с поиском в неориентированном графе. Главное отличие в этом случае заключается в том, что ребра графа проходятся только в соответствии с ориентацией. Как следствие этого ограничения, ребра в ориентированном графе G разбиваются поиском в глубину в G на четыре категории (а не на две, как в случае неориентированного графа). Непросмотренное ребро (v, w), встречающееся, когда мы находимся в вершине v, можно классифицировать следующим образом:

Случай 1. Вершина w еще не пройдена.

В этом случае (v, w) — *ребро дерева*.

Случай 2. Вершина w уже была пройдена.

а. Если w — потомок v в лесе ПВГ (т. е. подграфе на ребрах дерева), то ребро (v, w) называется *прямым ребром*.

б. Если w — предок v в лесе ПВГ, то ребро (v, w) называется *обратным ребром*.

в. Если v и w несоотносимы в лесе ПВГ и DFN $(w)<$DFN (v), то ребро (v, w) является *пересекающим ребром*. Отметим, что не су-

ществует пересекающих ребер типа (v, w), где $DFN(w)>DFN(v)$. Доказательство этого утверждения проводится аналогично доказательству теоремы 14.6. Сделаем несколько замечаний:

1. Ребро (v, w) при $DFN(w)>DFN(v)$ является либо ребром дерева, либо прямым ребром. Во время поиска различить ребро дерева и прямое ребро довольно просто, так как ребро дерева всегда ведет к новой вершине.

2. Ребро (v, w) при $DFN(w)<DFN(v)$ является либо обратным, либо пересекающим ребром. Такое ребро является обратным ребром, если w еще не полностью сканирована в тот момент, когда встречается ребро (v, w) при просмотре ребер, исходящих из v.

3. Лес ПВГ (подграф на ребрах дерева) может быть несвязным, даже если рассматриваемый ориентированный граф связен. Первую вершину, проходимую в каждой компоненте леса ПВГ, будем называть *корнем* соответствующей компоненты. Представим описание алгоритма ПВГ для ориентированного графа. В этом алгоритме мы используем новый массив SCAN, в котором для каждой вершины графа присутствует только один элемент. Вначале мы полагаем $SCAN(v)=0$ для каждой вершины v, тем самым указывая, что ни одна из вершин полностью не сканирована. Как только вершина полностью сканируется, соответствующий элемент в массиве SCAN полагается равным единице. Как отмечалось ранее, когда встречается ребро (v, w), для которого $DFN(w)<DFN(v)$, мы классифицируем его как обратное ребро, если $SCAN(w)=0$, или как пересекающее ребро в противном случае. Кроме того, мы используем два массива FORWARD и CROSS, в которых хранятся прямые и пересекающие ребра соответственно.

Алгоритм 14.5. ПВГ в ориентированном графе.

S1. Пусть G — данный ориентированный граф без петель. Пусть $TREE=\varnothing$, $FORWARD=\varnothing$, $BACK=\varnothing$, $CROSS=\varnothing$ и $i=1$. Для каждой вершины v графа G положить $MARK(v)=0$, $FATHER(v)=0$ и $SCAN(v)=0$.

S2. (Начало ПВГ с нового корня.) Взять произвольную вершину, например r, для которой $MARK(r)=0$. Положить $DFN(r)=i$, $MARK(r)=1$, $v=r$.

S3. Если все ребра, исходящие из вершины v, уже помечены как «просмотренные», то $SCAN(v)=1$ и идти к шагу S5 (v — полностью сканирована). Иначе выбрать ребро (v, w), которое еще не помечено как «просмотренное», и идти к шагу S4.

S4. Пометить ребро (v, w) как «просмотренное», выполнить действия 1 и 2 и идти к шагу S3.
1) Если $MARK(w)=0$, то положить $i=i+1$, $DFN(w)=i$, $TREE=TREE \cup \{(v, w)\}$, $MARK(w)=1$, $FATHER(w)=v$, $v=w$.
2) В противном случае положить $FORWARD=FORWARD \cup \{(v, w)\}$, если $DFN(w)>DFN(v)$, $BACK=BACK \cup \{v, w)\}$, если $DEN(w)<DFN(v)$ и $SCAN(w)=0$, иначе $CROSS=CROSS \cup \{(v, w)\}$.

S5. Если $FATHER(v)\neq 0$ (т. е. v — не корень), то положить $v=FATHER(v)$ и идти к шагу S3. Иначе идти к шагу S6.

S6. Если для любой вершины $MARK(x)=1$, то идти к шагу S7, иначе положить $i=i+1$ и идти к шагу S2.

S7. (ПВГ завершен.) HALT.

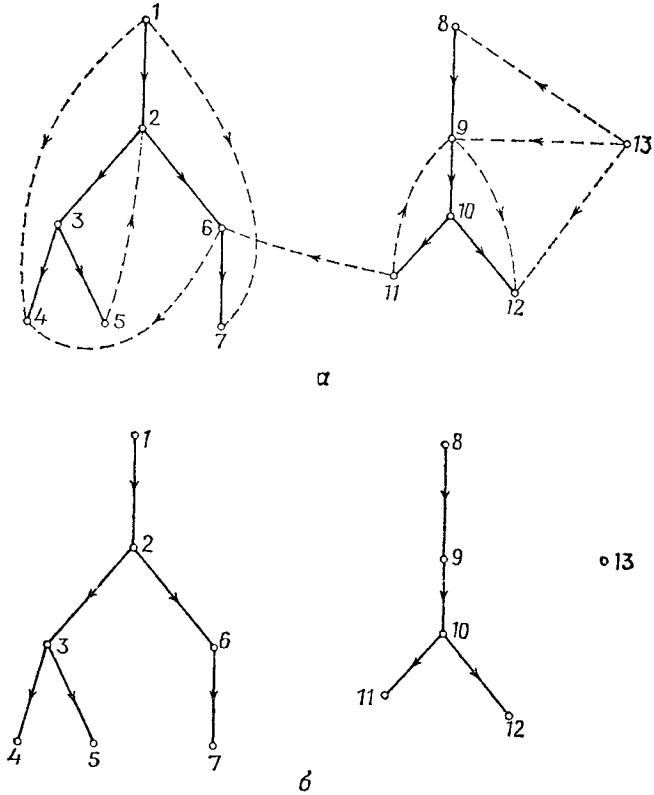


Рис. 14.14.
а — ПВГ в ориентированном графе G; *б* — лес ПВГ для графа G.

На рис. 14.14, *а* представлен ПВГ в ориентированном графе. Для каждой вершины указана ее глубина. Ребра дерева изображены сплошными, а другие ребра — штриховыми линиями. Лес ПВГ изображен отдельно на рис. 14.14, *б*. Мы отмечали ранее, что лес ПВГ ориентированного графа не может быть связным, даже если этот граф связен. Это видно на рис. 14.14, *б*. Тогда возникает задача определения достаточных условий для связности леса ПВГ. Докажем, что лес ПВГ сильно связного графа связен. Фактически мы будем доказывать более общий результат.

Пусть T — лес ПВГ в ориентированном графе $G=(V, E)$. Пусть $G_i=(V_i, E_i)$, где $|V_i|\geqslant 2$ — сильно связная компонента графа G. Рассмотрим две произвольные вершины v и w графа G_i. Не нарушая общности, предположим, что $\mathrm{DFN}(v)<\mathrm{DFN}(w)$. Так как G_i сильно связен, то существует ориентированный путь P из v в w в графе G_i. Пусть x — вершина пути P с наименьшей глубиной, и пусть T_x —

поддерево дерева T с корнем x. Отметим, что только пересекающие и обратные ребра могут выходить из поддерева T_x. Так как эти ребра ведут к вершинам, имеющим меньшую глубину, чем DFN(x), то отсюда следует, что по достижении в пути P вершины из T_x все последующие в пути P вершины будут также принадлежать T_x, в частности w также принадлежит T_x. Поэтому она является потомком x. Так как DFN$(x)\leqslant$DFN$(v)<$DFN(w), то из алгоритма ПВГ следует, что v также принадлежит T_x. Таким образом, две любые вершины v и w графа G_i имеют общего предка, который также принадлежит G_i.

Из вышеизложенного мы можем сделать вывод, что все вершины G_i имеют общего предка r_i, который также принадлежит G_i. Среди всех общих предков в T для вершин из G_i вершина r_i имеет наибольшую глубину. Более того, легко показать, что если v — вершина G_i, то любая вершина на пути дерева из r_i к v также будет в G_i. Таким образом, подграф T, порожденный множеством вершин V_i, связен, т. е. доказана следующая теорема:

Теорема 14.7. Пусть $G_i=(V_iE_i)$ — сильно связная компонента ориентированного графа $G=(V, E)$. Если T — лес ПВГ, то подграф T, порожденный множеством вершин V_i, связен.

Следующее утверждение является непосредственным следствием из этой теоремы.

Следствие 14.7.1. Лес ПВГ сильно связного графа связен. Легко показать, что алгоритмы ПВГ 14.4 и 14.5 имеют сложность $O(n+m)$, где n — число вершин, а m — число ребер графа.

14.4. Двусвязность и сильная связность

В этом разделе мы рассматриваем алгоритмы Хопкрофта и Тарьяна [14.20] и Тарьяна [14.19] определения двусвязных и сильно связных компонент графа. Эти алгоритмы основываются на поиске в глубину. Мы начинаем обсуждение с алгоритма двусвязности.

14.4.1. Двусвязность

Напомним (гл. 8), что двусвязный граф — это связный граф без точек сочленения. Максимальный двусвязный подграф графа называется *двусвязной компонентой* графа [1].

Решающим этапом разработки алгоритма двусвязности является определение простого критерия, который можно использовать для выделения точек сочленения. Такой критерий формулируется и обосновывается в следующих двух леммах.

Пусть $G=(V, E)$ — связный неориентированный граф. Пусть T — дерево ПВГ этого графа с вершиной r в качестве корня. Тогда мы имеем следующее:

[1] Отметим, что двусвязная компонента есть то же самое, что и блок, определенный в разд. 1.7.

Лемма 14.4. Вершина $v \neq r$ является точкой сочленения графа G тогда и только тогда, когда для некоторого сына s вершины v нет обратных ребер, соединяющих какие-либо потомки s в T (включая ее саму) с собственным предком v.

Доказательство. Пусть G' — граф, который получается после удаления вершины v из графа G. По определению v — точка сочленения графа тогда и только тогда, когда граф G' несвязен.

Пусть $s_1, s_2, \ldots, s_k$ — сыновья v в T. Для каждого i, $1 \leqslant i \leqslant k$, пусть V_i — множество потомков вершины s_i (включая ее саму) и пусть G_i — подграф G', порожденный множеством вершин V_i. Более того, пусть $V'' = V' - \bigcup_{i=1}^{k} V_i$, где $V' = V - \{v\}$, и пусть G'' — подграф, порожденный множеством вершин V''. Отметим, что все собственные предки v принадлежат множеству V''.

Объединение множеств вершин графов $G_1, G_2, \ldots, G_k$ и G'', являющихся подграфами графа G, равно множеству вершин графа G'. Мы можем показать, что все эти подграфы связны. Далее, в соответствии с теоремой 14.6, не существует ребер, соединяющих вершины, принадлежащие различным графам G_i. Отсюда следует, что G' будет связен тогда и только тогда, когда для любого i $(1 \leqslant i \leqslant k)$ существует ребро (a, b) между вершинами $a \in V_i$ и $b \in V''$. Такое ребро обязательно будет обратным ребром, а вершина b — собственным предком вершины v. Поэтому мы можем сделать вывод о том, что граф G' будет связен тогда и только тогда, когда для каждого сына s_i вершины v существует обратное ребро между некоторым потомком вершины s_i (включая ее саму) и собственным предком вершины v.

Лемма 14.5. Корневая вершина r — точка сочленения графа G тогда и только тогда, когда она имеет более одного сына.

Доказательство в этом случае аналогично доказательству леммы 14.4.

В дальнейшем мы будем ссылаться на вершины, указывая ее глубину. Чтобы вложить в процедуру ПВГ критерий, определенный леммами 14.4 и 14.5, мы определяем для каждой вершины v графа G:

$$\mathrm{LOW}(v) = \min(\{v\} \cup \{w \mid \text{существует такое обратное ребро } (x, w), \text{ что } x \text{ — потомок } v \text{ и } w \text{ — собственный предок } v \text{ в } T\}). \quad (14.1)$$

Используя значения LOW, определенные выше, мы можем переформулировать критерий, предложенный в лемме 14.4, как следующую теорему:

Теорема 14.8. Вершина $v \neq r$ — точка сочленения графа G тогда и только тогда, когда v имеет сына s, для которого $\mathrm{LOM}(s) \geqslant v$.

Отметим, что значение LOW (v) соответствует вершине, помеченной наименьшей глубиной и достижимой из v с помощью ориентированного пути, содержащего не более одного обратного ребра, т. е. мы можем переформулировать (14.1) следующим образом: $\mathrm{LOW}(v) = \min(\{v\} \cup \{\mathrm{LOW}(s) \mid s$ — сын $v\} \cup \{w \mid (v, w)$ — обратное ребро$\})$. Это эквивалентное определение LOW (v) предполагает следующие шаги по его вычислению:

1. Когда v проходится в ПВГ в первый раз, полагаем LOW (v) равным глубине v.
2. Когда рассматривается обратное ребро (v, w), инцидентное v, полагаем LOW (v) равным наименьшему из текущего значения и глубины w.
3. Когда ПВГ возвращается к v после полного сканирования сына s этой вершины, полагаем LOW (v) равным наименьшему из текущего значения и LOW (s).

Отметим, что для любой вершины v вычисление LOW (v) заканчивается при завершении ее сканирования.

Рассмотрим вопрос о выделении ребер, принадлежащих двусвязным компонентам. Для этого мы используем массив STACK. Вначале массив STACK пуст. По мере просмотра ребер они добавляются в STACK сверху.

Предположим, что ПВГ возвращается в вершину v после полного сканирования сына этой вершины s. В этот момент вычисление LOW (s) завершается. Предположим, обнаружено, что $\mathrm{LOW}(s) \geqslant v$. Тогда, согласно теореме 14.8, v — точка сочленения. Далее, если s — первая вершина с этим свойством, то можно легко определить, что ребро (v, s) вместе с ребрами, инцидентными s и его потомкам, образуют двусвязную компоненту. Эти ребра являются в точности теми ребрами,

которые лежат в массиве STACK сверху, включая ребро (v, s). Они удаляются из STACK, и начиная с этого момента алгоритм ведет себя так, как если бы он был применен к графу G', который получается из графа G удалением ребер уже выделенной двусвязной компоненты.

Например, дерево ПВГ связного графа может иметь вид, изображенный на рис. 14.15. $G_1, G_2, \ldots, G_5$ — двусвязные компоненты в том порядке, в котором они выделяются.

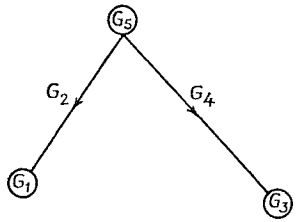


Рис. 14.15. G_1, G_2, G_3, G_4, G_5 — двусвязные компоненты графа.

Представим описание алгоритма определения двусвязности. Этот алгоритм во многом подобен алгоритму 14.4 с включением соответствующих шагов для вычисления LOW (v) и выделения точек сочленения и ребер, принадлежащих различным двусвязным компонентам. Отметим, что в этом алгоритме корневая вершина рассматривается как точка сочленения, даже если она таковой не является для выделения двусвязной компоненты, содержащей r.

Алгоритм 14.6. Двусвязность.

S1. G — данный связный граф. Для каждой вершины v графа G положить FATHER $(v)=0$ и MARK $(v)=0$. Положить $i=1$ и STACK$=\varnothing$.

S2. Выбрать произвольную вершину, например r, с MARK $(r)=0$. Положить DFN $(r)=i$, LOW $(r)=i$, MARK $(r)=1$, $v=r$.

S3. Если все ребра, инцидентные v, уже помечены как просмотренные, то идти к шагу S5. Иначе выбрать ребро (v, w), которое еще не помечено как «просмотренное». Пометить это ребро «просмотренным», добавить его в STACK сверху и идти к шагу S4.

S4. Выполнить следующие действия и идти к шагу S3.

1. Если MARK $(w)=0$, то положить $i=i+1$, DFN $(w)=i$, LOW $(w)=i$, FATHER $(w)=v$, MARK $(w)=1$, $v=w$.

2. Если MARK $(w)=1$, то положить LOW $(v)=\min$ {LOW (v), DFN (w)}.

S5. Если FATHER $(v)\neq 0$, то идти к шагу S6, иначе идти к шагу S8.

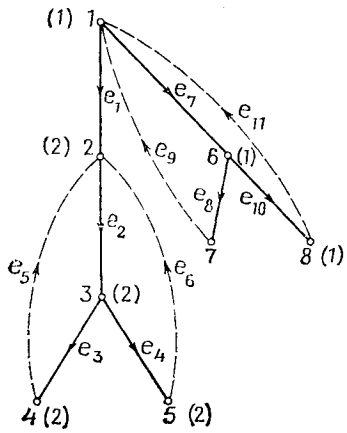


Рис. 14.16. Иллюстрация к алгоритму 14.6.

Значения элементов массива LOW приведены в скобках; $\{e_2, e_3, e_4, e_5, e_6\}$, $\{e_1\}$, $\{e_7, e_8, e_9 e_{10}, e_{11}\}$ — двусвязные компоненты графа.

S6. Если LOW (v)$\geqslant$DFN (FATHER (v)), то удалить все ребра из STACK сверху до ребра (FATHER (v), v) включительно. (Выделена компонента двусвязности.)

S7. Положить LOW (FATHER (v))$=\min\{$LOW (v), LOW (FATHER (v))$\}$, $v=$ =FATHER (v) и идти к шагу S3.

S8. (Все компоненты двусвязности выделены.) HALT.

На рис. 14.16 иллюстрируется применение этого алгоритма.

14.4.2. Сильная связность

Напомним, что граф сильно связен, если для любых двух его вершин v и w существуют ориентированные пути из v к w и из w к v, а максимальный сильно связный подграф графа G называется сильно связной компонентой этого графа.

Рассмотрим ориентированный граф $G=(V, E)$. Пусть $G_1=(V_1, E_1)$, $G_2=(V_2, E_2)$, . . ., $G_k=(V_k, E_k)$ — сильно связные компоненты графа G. Пусть T — лес ПВГ графа G, а T_1, T_2, . . ., T_k — порожденные подграфы T с множествами вершин V_1, V_2, . . ., V_k соответственно. Из теоремы 14.7 следует, что T_1, T_2, . . ., T_k — связные графы.

Пусть r_i $(1\leqslant i\leqslant k)$ — корень T_i. Предположим, что если $i<j$, то ПВГ завершается в вершине r_i ранее, чем в r_j. Тогда для каждого $i<j$ либо r_i левее r_j, либо r_i — потомок r_j в T. G_i будет состоять из тех вершин, которые являются потомками r_i, но не входят ни в один из графов G_1, G_2, . . ., G_{i-1}.

Первым шагом в разработке алгоритма сильной связности является определение простого критерия, который можно использовать для выделения корня при выполнении ПВГ. Следующие замечания будут полезны при выводе такого критерия. Они являются прямыми следствиями того факта, что в графе, получаемом с помощью стягивания всех ребер в каждом из множества E_1, E_2, ..., E_k (разд. 5.1), нет ориентированных циклов.

1. Не существует обратных ребер вида (v, w), где $v\in V_i$, а $w\in V_j$, $i\neq j$. Другими словами, все обратные ребра, исходящие из вершин множества V_i, заканчиваются также в вершинах множества V_i.

2. Не существует пересекающих ребер вида (v, w), где $v\in V_i$, $w\in V_j$, $i\neq j$ и r_j — предок r_i. Таким образом, для каждого пересекающего ребра (v, w) выполняются следующие два условия:

а. Если $v\in V_i$ и $w\in V_j$ для некоторых i и j, $i\neq j$, то r_j находится слева от r_i.

б. Для некоторого i $v\in V_i$ и $w\in V_i$. Предположив, что вершины именуются их глубинами, определим для каждого v графа G LOWLINK$=\min(\{v\}\cup\{w|$ существует пересекающее или обратное ребро из потомка v в w и w находится в той же самой компоненте сильной связности, что и $v\})$.

Предположим, что $v\in V_i$. Тогда из вышеприведенного определения следует, что LOWLINK (v) — наименьшая вершина в V_i, до которой существует путь из v, содержащий не более одного

обратного или пересекающего ребра. Отсюда следует, что все ребра такого ориентированного пути обязательно будут принадлежать графу G_i. В качестве непосредственного следствия мы получаем утверждение

$$\text{LOWLINK}(r_i)=r_i \text{ для всех } 1\leqslant i\leqslant k. \tag{14.2}$$

Предположим, что $v\in V_i$ и $v\neq r_i$. Тогда в графе G_i существует ориентированный путь P из v в r_i. Такой ориентированный путь P должен обязательно содержать обратные или пересекающие ребра, так как $r_i<v$ и только пересекающие или обратные ребра ведут к вершинам с меньшей глубиной. Другими словами, P содержит вершину $w<v$. Итак, для $v\neq r_i$ мы получаем

$$\text{LOWLINK}(v)<v. \tag{14.3}$$

Объединяя выражения (14.2) и (14.3), получим следующую теорему, которая характеризует корни сильно связных компонент ориентированного графа:

Теорема 14.9. Вершина v — корень сильно связной компоненты ориентированного графа тогда и только тогда, когда LOWLINK $(v)=v$.

Следующие действия могут быть полезны для вычисления LOWLINK (v) при выполнении ПВГ.

1. При прохождении v в первый раз устанавливаем LOWLINK (v) равным глубине v.
2. Если рассматривается обратное ребро (v, w), то полагаем LOWLINK (v) равным минимуму из его текущего значения и глубины w.
3. Если исследуется пресекающее ребро (v, w), для которого v и w принадлежат одной и той же сильно связной компоненте, то полагаем LOWLINK (v) равным минимуму из его текущего значения и глубины w.
4. Когда мы возвращаемся в v после полного сканирования сына s вершины v, то полагаем LOWLINK (v) равным минимуму из его текущего значения и LOWLINK (s).

Для выполнения шага 3 необходимо проверить, находится ли вершина w в той же сильно связной компоненте, что и v. Для этого мы используем массив STACK1, в который вершины графа G добавляются в порядке прохождения их при ПВГ. STACK1 используется также при определении вершин, принадлежащих какой-либо сильно связной компоненте.

Пусть v — первая вершина при поиске в глубину, для которой выполнено соотношение LOWLINK $(v)=v$. Тогда, согласно теореме 14.9, v — корень и фактически является r_1. В этот момент вершины сверху STACK1 до вершины v включительно являются вершинами, образующими граф G_1. Таким образом, граф G_1 легко выделяется. Эти вершины удаляются из STACK1. После этого алгоритм применяется к графу G', который получается из графа G удалением вершин графа G_1.

Положим, как и при рассмотрении реализации шага 3 в вычислении LOWLINK, что $v\in V_i$ и (v, w) — пересекающее ребро, встретившееся при рассмотрении ребер, инцидентных v. Предположим, что w не принадлежит той же самой сильно связной компоненте, которой принадлежит v. Тогда она должна принадлежать сильно связной компоненте G_j, корень r_j которой находится левее r_i (см. выше замечание 2а). Вершины такой компоненты должны быть уже выделены и более не должны находиться в STACK1. Таким образом, w будет в той же сильно связной компоненте, что и v, тогда и только тогда, когда w присутствует в STACK1.

Представим описание алгоритма сильной связности. Он совпадает с алгоритмом 14.5 с точностью до включения соответствующих шагов для вычисления ве-

личин LOWLINK и выделения вершин различных сильно связных компонент. Мы используем в этом алгоритме массив POINT. Вначале POINT $(v)=0$ для всех вершин v. Это указывает на то, что ни одна из вершин еще не включена в массив STACK1. POINT (v) устанавливается в 1, когда v добавляется в STACK1, и в 0, когда удаляется из STACK1.

Алгоритм 14.7. (Сильная связность.)

S1. G — данный ориентированный граф. Для каждой вершины v в G положить MARK $(v)=0$, FATHER $(v)=0$, POINT $(v)=0$. Положить $i=1$ и STACK1 $=\varnothing$.

S2. Выбрать произвольную вершину, например r, с MARK $(r)=0$. Положить DFN $(r)=i$, LOWLINK $(r)=i$, MARK $(r)=1$. Добавить r в STACK1, положить POINT $(r)=1$ и $v=r$.

S3. Если все ребра, инцидентные v, уже помечены как «просмотренные», то идти к шагу S5. Иначе выбрать ребро (v, w), которое еще не помечено как «просмотренное». Пометить его «просмотренным» и идти к шагу S4.

S4. Выполнить следующие действия и идти к шагу S3:
1) Если MARK $(w)=0$, то положить $i=i+1$, DFN $(w)=i$, LOWLINK $(w)=i$, FATHER $(w)=v$, MARK $(w)=1$. Добавить w в STACK 1 и положить POINT $(w)=1$ и $v=w$.
2) Если MARK $(w)=1$, DFN $(w)<$DFN (v) и POINT $(w)=1$, то положить LOWLINK $(v)=\min$ {LOWLINK (v), DFN (w)}.

S5. Если LOWLINK $(v)=$DFN (v), то удалить все вершины сверху STACK1 до вершины v включительно. (Эти вершины образуют сильно связную компоненту). Затем положить POINT $(x)=0$ для всех вершин x, удаляемых из STACK1.

S6. Если FATHER $(v)=0$, идти к шагу S7, иначе положить
1) LOWLINK (FATHER (v))$=\min$ {LOWLINK (FATHER (v)), LOWLINK (v)},
2) $v=$FATHER (v) и идти к шагу S3.

S7. Если для каждой вершины x MARK $(x)=1$, то идти к шагу S8, иначе идти к шагу S2.

S8. (Все сильно связные компоненты выделены.) HALT.

Рис. 14.17 иллюстрирует работу этого алгоритма.

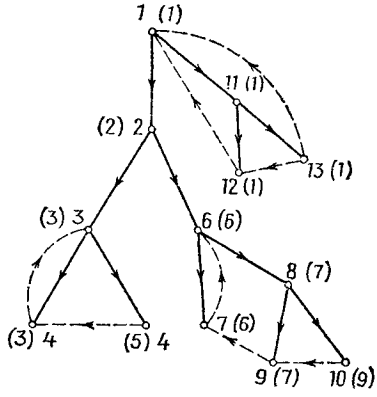


Рис. 14.17. Иллюстрация алгоритма 14.7.
Значения элементов массива LOWLINK приведены в скобках, {3, 4, 5}, {6, 7, 8, 9, 10}, {2} и {1, 11, 12, 13} — сильно связные компоненты графа.

14.5. Сводимость графа программы

Графом программы называется ориентированный граф G с выделенной вершиной s, из которой существует ориентированный путь в любую другую вершину графа G. Другими словами, любая

вершина графа G достижима из s. Вершина s называется *начальной вершиной* графа G. Мы полагаем, что в графе программы нет параллельных ребер. Это предположение не приводит к нарушению общности при дальнейшем изложении материала в этой главе.

Потоки управления в программе для ЭВМ можно промоделировать графом программы, в котором каждая вершина соответствует блоку команд, выполняемых последовательно. Такое представление оказывается очень полезным при изучении некоторых вопросов, связанных с задачей оптимизации кода программы.

Многие методы оптимизации кода применимы только тогда, когда граф программы обладает специальным свойством, называемым *сводимостью* [14.21—14.29].

Сводимость графа программы G определяется исходя из следующих двух преобразований графа G:

S_1: Удалить петлю (v, v) из графа G.

S_2: Если (v, w) — единственное ребро, входящее в w и $w \neq s$, то удалить вершину w. Для каждого ребра (w, x) графа G добавить ребро (v, x), если (v, x) еще не принадлежит графу G. (Это преобразование называется *коллапсированием* вершины w в вершину v.)

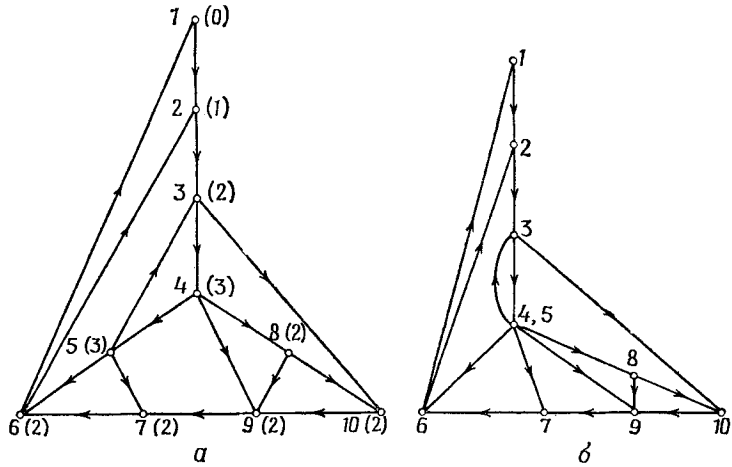


Рис. 14.18.

а — сводимый граф программы G; значения элементов массива HIGHPT1 приведены в скобках; *б* — граф, получающийся после коллапсирования в графе G вершины *5* в вершину *4*.

Например, коллапсирование вершины 5 в вершину 4 в графе программы, изображенном на рис. 14.18, *а*, приводит к графу на рис. 14.18, *б*.

Граф программы *сводим*, если его можно преобразовать в граф, состоящий только из вершины s, многократным применением преобразований S_1 и S_2.

Например, граф на рис. 14.18, *а* является сводимым. Это можно установить последовательным применением коллапсирования вершин в следующем порядке: 5, 8, 3, 10, 9, 7, 6, 2.

Кок [14.22] и Аллен [14.27] первыми сформулировали понятие «сводимость», и их определение сделано на основе метода, называемого *интервальным анализом*. Определение, приведенное выше, предложено Хектом и Ульманом [14.30] и эквивалентно определению Кока и Аллена.

Если граф сводим, то можно показать [14.30], что любой граф G', получаемый из графа G с помощью одного или нескольких применений преобразований S_1 и S_2, также сводим. Таким образом, порядок применения преобразований при проверке сводимости несуществен. Некоторые интересные классы программ, например программы «go-to-less», приводят к графам, которые обязательно сводимы [14.30], и большинство программ можно промоделировать сводимым графом посредством процесса «расщепления вершин» [14.31].

Предположим, что мы желаем проверить сводимость графа G. Это можно осуществить первоначальным удалением петель с помощью преобразования S_1 и последующим подсчетом числа ребер, заходящих в каждую вершину. Затем мы можем найти вершину w с единственным заходящим в него ребром (v, w) и применить преобразование S_2, коллапсируя w в v. Мы можем повторять этот процесс до тех пор, пока граф не будет сжат полностью либо пока не обнаружится, что он несводим. Ясно, что каждое применение преобразования S_2 требует $O(n)$ шагов, где n — число вершин графа G. Таким образом, сложность этого алгоритма $O(n^2)$. Хопкрофт и Ульман [14.32] улучшили этот алгоритм до сложности $O(m \log m)$, где m — число ребер графа G. Тарьян [14.33, 14.34] позже представил алгоритм, который был лучше алгоритма Хопкрофта и Ульмана.

Хект и Ульман [14.30, 14.35] дали несколько полезных структурных характеризаций графов программ. Одна из них определяется следующей теоремой:

Теорема 14.10. Пусть G — граф программы с начальной вершиной s. Он сводим тогда и только тогда, когда не существует вершин $v \neq s$ и $w \neq s$, ориентированных путей P_1 из s в v и P_2 из s в w, а также такого ориентированного цикла C, содержащего v и w, что он не имеет общих ребер, а только по одной общей вершине с каждым из путей P_1 и P_2 (рис. 14.19).

Доказательство этой теоремы приведено в работах [14.30, 14.36].

Перейдем к обсуждению алгоритма Тарьяна для проверки сводимости графа программы. Этот алгоритм базируется на характеризации сводимых графов, которую мы докажем, применяя теорему 14.10, и использует ПВГ. Наше обсуждение основывается на данных работы [14.33].

Пусть G — граф программы с начальной вершиной s. Пусть T — дерево ПВГ графа G с s в качестве корня. Далее мы будем ссылаться на вершины по их глубине.

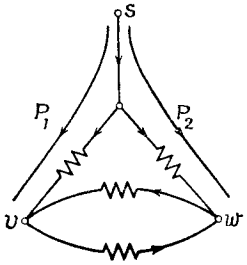


Рис. 14.19. Основной несводимый граф.

Теорема 14.11. Граф G сводим тогда и только тогда, когда он не содержит ориентированного пути P из s в некоторую такую вершину v, что v является собственным предком в T некоторой другой вершины из P.

Доказательство. Предположим, что граф G — несводим. Тогда существуют вершины v и w, ориентированные пути P_1 и P_2 и цикл C, которые удовлетворяют условию теоремы 14.10. Предположим, что $v<w$. Пусть C_1— часть цикла C, ведущая из v к w. Тогда C_1 содержит некоторый общий предок как для v, так и для w, а ориентированный путь, состоящий из P_2 с последующей частью цикла C, ведущей из w в u, удовлетворяет условию теоремы.

И наоборот, предположим, что существует ориентированный путь P, который удовлетворяет условию теоремы. Тогда пусть v — первая вершина пути P, которая является собственным предком некоторой более ранней вершины P. Пусть w — первая вершина пути P, которая является потомком v. Пусть P_1— часть T из s в v, а P_2— часть P из s в w. Пусть также C — ориентированный цикл, состоящий из части P, ведущей из w в v, с последующей частью дерева, ведущей из v в w. Тогда мы получаем, что v, w, p_1, p_2, C удовлетворяют условиям теоремы 14.10. Следовательно, G — несводимый граф.

Пусть для любой вершины v HIGHPT1 (v) — такой наивысший собственный предок v, что существует ориентированный путь из v в HIGHPT 1 (v) и P не включает в себя собственных предков v, исключая HIGHPT 1 (v). Мы полагаем HIGHPT 1 (v)=0, если не существует ориентированного пути из v в какой-либо собственный предок v. На рис. 14.18, *а* указаны в скобках значения HIGHPT 1 для соответствующих вершин.

Отметим, что при вычислении HIGHPT 1 (v) мы можем игнорировать прямые ребра, так как если P — ориентированный путь из v в w и P не содержит предков v, кроме v и w, то мы можем сопоставить каждому прямому ребру из P путь из ребер дерева или части его и имеем, кроме того, ориентированный путь из v в w, который не содержит предков v, кроме v и w.

Алгоритм Тарьяна основан на следующей характеризации графов программ:

Теорема 14.12. G — сводимый граф тогда и только тогда, когда нет таких вершины v и ребра (u, v), заходящего в v, что $w<$HIGHPT1 (v), где w — наивысший общий предок u и v.

Доказательство. Предположим, что G — несводимый граф. Тогда, согласно теореме 14.11, существует ориентированный путь P из s в v, где v — собственный предок некоторой другой вершины из P. Выберем наиболее короткий путь P. Пусть w — первая вершина P, которая является потомком v. Тогда все вершины, кроме v, которые следуют за w в P, являются потомками v в T. Другими словами, часть P из w в v не включает собственных предков вершины w, кроме v. Поэтому HIGHPT 1 (w)$\geqslant v$. Таким образом, w, HIGHPT 1 (w) и ребро P, входящее в w, удовлетворяют условию теоремы.

Предположим, что условия теоремы выполняются. Тогда ребро (u, v) не является обратным ребром, потому что в таком случае наибольший общий предок w для u и v равен v и $w=v \geqslant \mathrm{HIGHPT}\,1(v)$. Таким образом, (u, v) — либо прямое, либо пересекающее ребро. Пусть P_1— ориентированный путь из v в $\mathrm{HIGHPT1}(v)$, который не проходит ни через один собственный предок v, кроме $\mathrm{HIGHPT}\,1(v)$. Тогда ориентированный путь, состоящий из ребер дерева, из s в u с последующим ребром (u, v) и путем P_1, удовлетворяет условию теоремы 14.11. Поэтому G — несводимый граф.

Отсюда должно быть ясно, что проверка сводимости графа G, использующая приведенную выше теорему, включает следующие основные шаги:

1. Выполнить ПВГ на G с s в качестве корня.
2. Вычислить $\mathrm{HIGHPT1}(v)$ для каждой вершины v графа G.
3. Для пересекающих ребер проверить при вычислении HIGHPT1 выполнение условий теоремы 14.12.
4. Для прямых ребер проверить после вычисления HIGHPT1 выполнение условий теоремы 14.12.

Отметим, что, как мы видели раньше, прямые ребра могут игнорироваться при вычислении HIGHPT1. Из доказательства теоремы 14.12 следует, что обратные ребра не нужны при проверке условий упомянутой теоремы.

Для вычисления значений HIGHPT1 мы упорядочиваем обратные ребра (u, v) по глубине v. Затем мы обрабатываем обратные ребра в порядке от большей к меньшей вершине v. Первоначально все вершины не помечены. Чтобы обработать обратное ребро (u, v), мы должны подниматься вдоль пути в дереве из u в v, помечая каждую текущую непомеченную вершину глубиной вершины v. (Мы не помечаем саму вершину v.) Если вершина w уже помечена, то мы проверяем все пересекающие ребра, заходящие в w. Если (z, w) — такое пересекающее ребро (рис. 14.20), то мы поднимаемся вдоль пути в дереве из z в v, помечая каждую непомеченную вершину глубиной вершины v. Если z не является потомком v, то граф G несводим по теореме 14.12 и вычисление прекращается. Мы продолжаем помечать вершины, пока не будут рассмотрены все пересекающие ребра, входящие в уже помеченные вершины, после чего мы рассматриваем следующее обратное ребро. При рассмотрении всех обратных ребер метки будут представлять собой значения HIGHPT1 для соответствующих вершин. Каждая непомеченная вершина имеет значение HIGHPT1, равное 0.

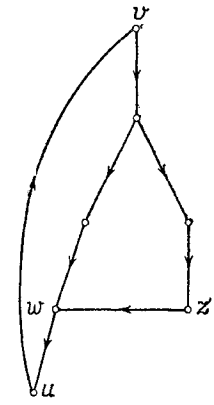


Рис. 14.20.

Опишем алгоритм Тарьяна проверки сводимости. В этом алгоритме мы используем n очередей, называемых ячейками, по одной для каждой вершины. Ячейка $\mathrm{BUCKET}(w)$, соответствующая

вершине w, содержит список обратных ребер (u, w), заходящих в вершину w. При обработке обратного ребра (u, w) нам необходимо хранить вершины, из которых можно попасть в u с помощью ориентированных путей. Для этой цели используется множество CHECK.

Алгоритм 14.8. (Сводимость графа программы (Тарьян).)

S1. Выполнить ПВГ на данном графе программы G на n вершинах. Обозначить вершины их глубинами. Упорядочить обратные ребра (u, v) графа G по глубинам v.
S2. Для $i=1, 2, \ldots, n$ положить HIGHPT 1 $(i)=0$, BUCKET (i) = «пустой список».
S3. Добавить все обратные ребра (u, w) в BUCKET (w).
S4. Положить $w=n-1$.
S5. Если BUCKET (w) пуст, то идти к шагу S6, иначе идти к шагу S7.
S6. Положить $w=w-1$. Если $w<1$, то идти к шагу S16, иначе идти к шагу S5.
S7. (Начало обработки нового обратного ребра.) Удалить обратное ребро (x, w) из BUCKET (w) и положить CHECK $=\{x\}$.
S8. Если CHECK пуст (обработка обратного ребра завершена), то идти к шагу S5, иначе идти к шагу S9.
S9. Удалить u из CHECK.
S10. Если u — потомок w, то идти к шагу S11, иначе идти к шагу S17.
S11. Если $u=w$, то идти к шагу S8, иначе идти к шагу S12.
S12. Если HIGHPT 1 $(u)=0$, то идти к шагу S13, иначе идти к шагу S15.
S13. Положить HIGHPT 1 $(u)=w$.
S14. Для каждого пересекающего ребра (v, u) добавить v в CHECK.
S15. Положить $u=$FATHER (u). Идти к шагу S11.
S16. Если $u \geqslant$HIGHPT1 (v) для каждого прямого ребра (u, v) (граф сводим), то HALT, иначе идти к шагу S17.
S17. HALT. Граф несводим.

Отметим, что шаг S10 в алгоритме 14.8 требует, чтобы мы имели возможность определить, является ли вершина w потомком другой вершины u. Пусть ND (u) — число потомков вершины u в T. Тогда мы можем показать, что w — потомок u тогда и только тогда, когда $u \leqslant w < u+\mathrm{ND}(u)$ (упражнение 14.3). Мы можем вычислить ND (u) непосредственно при выполнении ПВГ.

Эффективность алгоритма 14.8 существенно зависит от эффективности вычисления HIGHPT1. Чтобы сделать вычисление эффективным, необходимо избежать проверки вершин, которые уже были помечены. Тарьян предложил процедуру для обеспечения этого. Следующее рассуждение образует основу этой процедуры.

Полагаем, что на шаге S12 мы проверяем вершину u, чтобы ее пометить. Пусть на этой стадии u' — наибольший непомеченный собственный предок u. Это означает, что все собственные предки u, кроме u', которые лежат между u' и u в T, уже помечены. Таким образом, u' — следующая вершина, которая будет рассмотрена. Более того, когда u помечена, то все вершины, для которых u является наибольшим непомеченным собственным предком, будут иметь u' в качестве своего наибольшего непомеченного собственного предка. Для реализации этого метода мы будем использовать

множества, пронумерованные в пределах 1—n. Вершина $w \neq 1$ будет находиться в множестве v, т. е. в SET(v), если v является наибольшим непомеченным собственным предком w. Так как вершина 1 никогда не помечается, то каждая вершина всегда находится в каком-либо множестве. Первоначально вершина помещается в множество, номер которого является номером ее отца в T. Таким образом, i первоначально помещается в SET(FATHER(i)) для $i=2, 3, \ldots, n$. Чтобы пройти шаг S15, мы находим номер u' множества, содержащего u, и считаем его новым u. Когда u становится помеченной, мы соединяем множества, пронумерованные u и u', и формируем новое множество с номером u'. Таким образом, u' становится наибольшим непомеченным собственным предком всех вершин из старого множества SET(u).

Иными словами, метод, описанный выше, можно реализовать путем замены шагов S12 — S15 в алгоритме 14.8 на следующую последовательность шагов. Множества SET инициализируются указанным ранее способом.

S12′. Положить u' — номер множества, содержащего u.
S13′. Если HIGHPT 1(u)=0, то идти к шагу S 14, иначе идти к шагу S16.
S14′. Положить 1) HIGHPT 1 (u)=w и 2) SET (u')=SET (u) $\cup$ SET (u').
S15′. Для каждого пересекающего ребра (v, u) добавить v в CHECK.
S16′. Положить $u=u'$. Идти к шагу S11.

Можно показать, что такой модифицированный способ вычисления HIGHPT1 требует $O(n)$ объединений множеств, $O(m+n)$ выполнений шага S12′ и $O(m+n)$ теоретико-множественных операций. Если мы используем алгоритм Фишера [14.37], а также Хопкрофта и Ульмана [14.38] для выполнения объединений непересекающихся множеств и шага S12, то из анализа, проведенного Тарьяном [14.39], следует, что алгоритм сводимости имеет сложность $O(m\,\alpha(m, n))$, где $\alpha(m, m)$ — очень медленно растущая функция, которая связана с обращением функции Аккермана $A(p, q)$ и определяется следующим образом: $\alpha(m, n)=\min\{z \geqslant 3 | A(z, 4\lceil m/n \rceil) > \log_2 n\}$. *Функция Аккермана* имеет вид

$$A(p, q)=\begin{cases} 2q, & p=0. \\ 0, & q=0,\ p \geqslant 1. \\ 2, & p \geqslant 1,\ q=1. \\ A(p-1, A(p, q-1)), & p \geqslant 1,\ q \geqslant 2. \end{cases} \qquad (14.4)$$

Отметим, что функция Аккермана является очень быстрорастущей функцией. Поэтому A (3, 4) — очень большое число, и можно показать, что $\alpha(m, n) \leqslant 3$, если $m \neq n$ и $\log_2 n < A$ (3, 4). Алгоритм выполнения теоретико-множественных операций, упомянутый ранее, описан в работах [14.40, 14.41]. Алгоритм 14.8 неконструктивен в том смысле, что он не дает нам порядка, в котором необходимо

коллапсировать вершины при сведении сводимого графа. Однако, как мы увидим позже, эту информацию можно легко получить при работе алгоритма.

Будем приписывать вершинам при выполнении ПВГ номера, называемые SNUMBER, со значениями в пределах n — 1 в порядке завершения сканирования вершин. Фактически это тот же порядок, что и порядок, в котором устанавливаются в 1 соответствующие элементы массива SCAN в алгоритме 14.5. Можно легко проверить что

1) если (v, w) — ребро дерева, то SNUMBER (v)<SNUMBER (w);
2) если (v, w) — пересекающее ребро, то SNUMBER (v)<SNUMBER (w);
3) если (v, w) — обратное ребро, то SNUMBER (v)>SNUMBER (w);
4) если (v, w) — прямое ребро, то SNUMBER (v)<SNUMBER (w).

На рис. 14.21 в скобках указаны величины SNUMBER для соответствующих вершин графа, представленного на рис. 14.18, *а*.

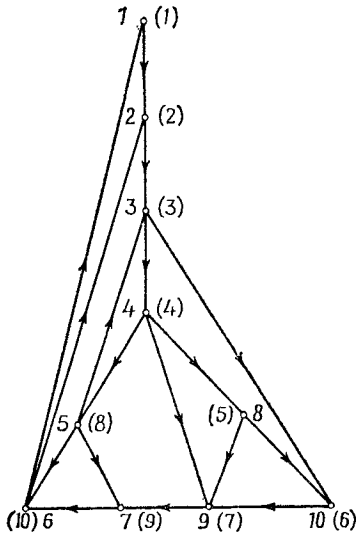


Рис. 14.21. Граф рис. 14.18, *а* со значениями элементов массива SNUMBER, указанными в скобках.

Предположим, что используется алгоритм сводимости и каждый раз, когда мы помечаем вершину v, мы приписываем ей пару (HIGHPT1 (v), SNUMBER (v)). Когда алгоритм заканчивается, мы упорядочиваем вершины так, что вершина, помеченная парой (x_1, y_1), появляется раньше вершины, помеченной парой (x_2, y_2), тогда и только тогда, когда $x_1 > x_2$ или $x_1 = x_2$ и $y_1 < y_2$. Этот порядок вершин называется *порядком редукции*. Отметим, что непомеченной вершине v приписывается пара (0, SNUMBER (v)).

Предположим, что v_a, v_b, v_c...— порядок редукции для сводимого графа программы G. Пусть T — дерево ПВГ для графа G с вершиной s в качестве корня. Используя теорему 14.12 и свойства чисел SNUMBER, указанные ранее, легко показать, что ребро дерева (u, v_a) является единственным ребром, заходящим в v_a. Предположим, что мы коллапсируем вершину v_a в вершину u, и пусть G' — получаемый сводимый граф. Пусть также T' — дерево, получаемое из T сжатием ребра (u, v_a). Тогда ясно, что T' — дерево ПВГ для графа G'.

1. Пересекающее ребро G соответствует пересекающему ребру либо прямому ребру графа G' либо не имеет соответствия.

2. Прямое ребро графа G соответствует прямому ребру графа G' либо не имеет соответствия.
3. Обратное ребро графа G соответствует обратному ребру графа G' либо не имеет соответствия.

Можно показать, что соответствующие значения HIGHPT1 для вершин графа G' совпадают со значениями для вершин графа G. То же самое справедливо и для значений SNUMBER. Таким образом, v_b, v_c, ... является порядком редукции для графа G'. Повторяя приведенные выше аргументы, получаем доказательство следующей теоремы.

Теорема 14.13. Если граф программы G сводим, то мы можем коллапсировать вершины графа G в порядке редукции, используя преобразование S_2 (с промежуточным применением преобразования S_1). Например, из величин HIGHPT 1, приведенных на рис. 14.18, *а*, и величин SNUMBER, приведенных на рис. 14.21, порядком редукции для графа с рис. 14.18, *а* является следующая последовательность: 4, 5, 3, 8, 10, 9, 7, 6, 2.

14.6. Доминаторы в графе программы

Пусть G — граф программы с начальной вершиной s. Если в нем вершина v входит в каждый ориентированный путь из s в w, то v называется *доминатором* w и обозначается DOM (w). Если v — доминатор w и любой другой доминатор w доминирует также и над v, то v называется *непосредственным доминатором* w и обозначается IDOM (w). Например, в графе программы G, представленном на рис. 14.22, *а*, вершина 1 является непосредственным доминатором вершины 9. Можно показать, что каждая вершина графа программы $G=(V, E)$, кроме начальной вершины s, имеет единственный непосредственный доминатор. Ребра $\{(\text{IDOM}(w), w) | w \in V-\{s\}\}$ образуют ориентированное дерево с корнем s, называемое *деревом доминаторов* графа G. Вершина v доминирует над вершиной w тогда и только тогда, когда она является собственным предком w в дереве доминаторов. Если граф G представляет собой схему потоков управления в программе ЭВМ, то дерево доминаторов дает информацию о том, какой вид выполнения программы является надежным. Дерево доминаторов графа программы, изображенного на рис. 14.22, *а*, представлено на рис. 14.22, *б*.

Рассмотрим алгоритм Ленгауэра и Тарьяна [14.42] для нахождения дерева доминаторов графа программы. Этот алгоритм является более простым и быстродействующим вариантом алгоритма, представленного ранее Тарьяном [14.43].

Пусть G — граф программы с начальной вершиной s. Пусть T — дерево ПВГ для этого графа. В последующем мы будем идентифицировать вершины графа G их глубиной. Более того, $x \overset{*}{\rightarrow} y$ означает, что x является предком y и T, $x \overset{*}{\rightarrow} y$ означает, что $x \overset{*}{\rightarrow} y$ в T и $x \neq y$, а $x \rightarrow y$ означает, что x является отцом y в T.

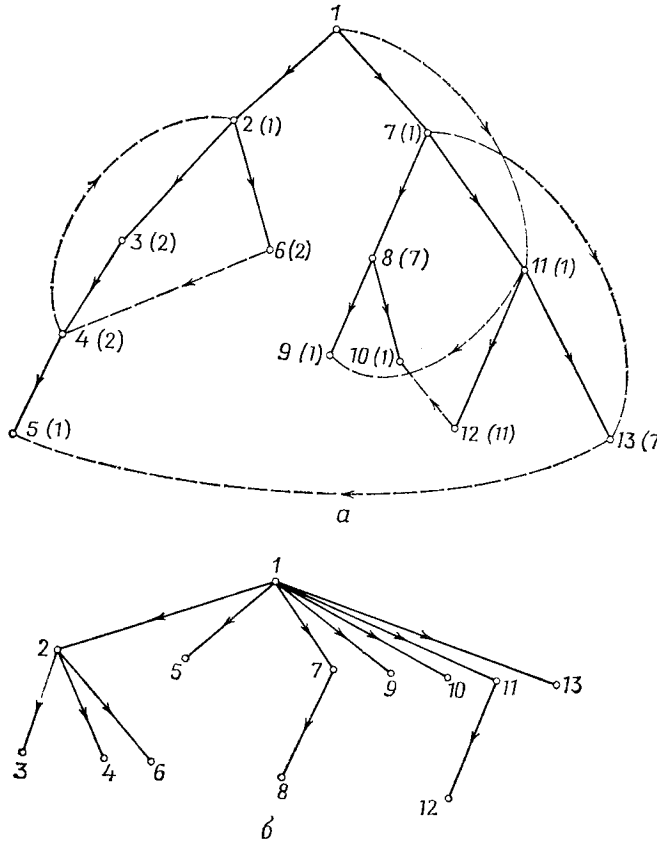


Рис. 14.22.
а — граф программы G; *б* — дерево доминаторов графа G.

Следующие две леммы необходимы для разработки алгоритма. Лемма 14.6 легко выводится из положений, ведущих к теореме 14.7. Лемму 14.7 можно доказать с помощью этих же положений.

Лемма 14.6. Если v и w — вершины графа G, для которых верно, что $v<w$, то любой ориентированный путь из v в w должен содержать общего предка вершин v и w в T.

Лемма 14.7. Пусть $w\neq s$, $v\overset{*}{\to}w$ и P — ориентированный путь из s в w. Пусть x — последняя вершина P, для которой $x<v$, и пусть y — первая вершина, следующая за x в P и удовлетворяющая условиям: $v\to y\overset{*}{\to}w$. Если $Q: x=v_0, v_1, v_2, \ldots, v_k=y$ — часть пути из x в y, то $v_i>y$ для $1\leqslant i\leqslant k-1$.

Определим для каждой вершины $w\neq s$: SDOM $(w)=\min\{v|$ существует такой ориентированный путь $v=v_0, v_1, v_2, \ldots, v_k=w$, что $v_i>w$ для $1\leqslant i\leqslant k-1\}$.

SDOM (w) будем называть *полудоминатором* w. Из определения следует, что

$$\text{SDOM}(w)<w. \qquad (14.5)$$

На рис. 14.22, *а* сплошные ребра являются ребрами дерева, а штриховые не являются ребрами дерева по отношению к ПВГ. Полудоминатор каждой вершины указан в скобках рядом с номером вершины.

Алгоритм Ленгауэра и Тарьяна в качестве первого шага вычисляет полудоминаторы для всех вершин. Затем полудоминаторы используются для вычисления непосредственных доминаторов вершин. Следующая теорема предоставляет способ их вычисления.

Теорема 14.14. Для любой вершины $w \neq s$,

$$\mathrm{SDOM}(w) = \min(\{v \mid (v, w) \in E \text{ и } v < w\} \cup \{\mathrm{SDOM}(u) \mid u > w \text{ и существует такое ребро } (v, w), \text{ что } u \overset{*}{\longrightarrow} v\}). \qquad (14.6)$$

Доказательство. Пусть x равно правой части выражения (14.6). Можно показать, используя определение полудоминаторов, что $\mathrm{SDOM}(w) \leqslant x$.

Чтобы доказать, что $\mathrm{SDOM}(w) \geqslant x$, положим $y = \mathrm{SDOM}(w)$, и пусть $y = v_0, v_1, v_2, \ldots, v_k = w$ — такой ориентированный путь, что $v_i > w$ для $1 \leqslant i \leqslant k-1$. Если $k=1$, то в соответствии с выражением (14.5) $(y, w) \in E$ и $y < w$. Таким образом, $\mathrm{SDOM}(w) = y \geqslant x$. Предположим, что $k > 1$. Пусть j — минимальное число, для которого верно, что $j \geqslant 1$ и $v_j \overset{*}{\longrightarrow} v_{k-1}$. Такое число j существует, так как в качестве него можно использовать число $k-1$. Потребуем, чтобы $v_i > v_j$ при $1 \leqslant i \leqslant j-1$.

Предположим обратное, что $v_i < v_j$ для некоторых $1 \leqslant i \leqslant j-1$. Тогда выберем такое число i, что $1 \leqslant i \leqslant j-1$ и v_i — минимально. Согласно лемме 14.6, $v_i \overset{*}{\longrightarrow} v_j$, что противоречит выбору числа j. Это доказывает правомерность приведенного выше требования.

В свою очередь это влечет следующее утверждение: $\mathrm{SDOM}(w) = y \geqslant \mathrm{SDOM}(v_j)$. Так как $v_j > w$, $v_j \overset{*}{\longrightarrow} v_{k-1}$ и $(v_{k-1}, w) \in E$, то из выражения (14.6) получаем, что $\mathrm{SDOM}(v_j) \geqslant x$. Следовательно, $\mathrm{SDOM}(w) \geqslant x$. Таким образом, либо $k=1$, либо $k>1$, а мы имеем $\mathrm{SDOM}(w) \geqslant x$, и теорема доказана.

Перейдем к обсуждению способа определения непосредственных доминаторов по полудоминаторам.

Доказательство следующих трех лемм достаточно просто и не приводится. В доказательстве леммы 14.9 используется лемма 14.6.

Лемма 14.8. Для любой вершины $w \neq s$ $\mathrm{IDOM}(w) \overset{+}{\longrightarrow} w$.

Лемма 14.9. Для любой вершины $w \neq s$ пусть $v = \mathrm{SDOM}(w)$, тогда $v \overset{+}{\longrightarrow} w$.

Лемма 14.10. Для любой вершины $w \neq s$ пусть $v = \mathrm{SDOM}(w)$, тогда $\mathrm{IDOM}(w) \overset{+}{\longrightarrow} v$.

Лемма 14.11. Пусть для вершин v и w выполняется условие $v \overset{*}{\longrightarrow} w$. Тогда либо $v \overset{*}{\longrightarrow} \mathrm{IDOM}(w)$, либо $\mathrm{IDOM}(w) \overset{*}{\longrightarrow} \mathrm{IDOM}(v)$.

Доказательство. Пусть x — произвольный собственный потомок $\mathrm{IDOM}(v)$, который является собственным предком v. Тогда существует ориентированный путь из s в v, который не включает x. Конкатенацией этого пути с путем в T из v в w мы получим ориентированный путь из s в w, который не включает x. Таким образом, $\mathrm{IDOM}(w)$ должен быть либо потомком v, либо предком $\mathrm{IDOM}(v)$.

Используя приведенные выше леммы, мы докажем два результата, которые предоставляют способ определения непосредственных доминаторов по известным полудоминаторам.

Теорема 14.15. Пусть $w \neq s$ и $v = \mathrm{SDOM}(w)$. Предположим, что для каждого u, такого, что $v \overset{+}{\longrightarrow} u \overset{*}{\longrightarrow} w$, выполняется неравенство $\mathrm{SDOM}(u) \geqslant \mathrm{SDOM}(w)$. Тогда $\mathrm{IDOM}(w) = v$.

Доказательство. По лемме 14.10 $\mathrm{IDOM}(w) \overset{*}{\longrightarrow} v$, и поэтому, чтобы доказать, что $\mathrm{IDOM}(w) = v$, достаточно показать, что v доминирует над w.

Рассмотрим произвольный ориентированный путь P из s в w. Пусть x — последняя вершина на этом пути, для которой $x<v$. Если такой вершины не существует, то $v=s$ доминирует над w. Иначе пусть y — первая вершина, следующая за x по пути и удовлетворяющая условию $v\overset{*}{\rightarrow}y\overset{*}{\rightarrow}w$. Пусть Q: $x=v_0, v_1, v_2, \ldots, v_k=y$ — часть пути P от x к y. Тогда по лемме 14.7 $v_i>y$ для $1\leqslant i\leqslant k-1$. А из этого вместе с определением полудоминаторов следует, что $\mathrm{SDOM}(y)\leqslant x<v=\mathrm{SDOM}(w)$. Поэтому $\mathrm{SDOM}(y)<\mathrm{SDOM}(w)$.

По предположению теоремы $\mathrm{SDOM}(u)\geqslant\mathrm{SDOM}(w)$ для каждого u, удовлетворяющего условию $v\overset{+}{\rightarrow}u\overset{*}{\rightarrow}w$. Поэтому y не может быть собственным потоком v. Так как y удовлетворяет условию $v\overset{*}{\rightarrow}y\overset{*}{\rightarrow}w$, то $y=v$ и v входит в путь P. Из произвольности пути P следует, что v доминирует над w.

Теорема 14.16. Пусть $w\neq s$ и $v=\mathrm{SDOM}(w)$. Пусть u — вершина, для которой $\mathrm{SDOM}(u)$ минимальна среди вершин u, удовлетворяющих условию $v\overset{+}{\rightarrow}u\overset{*}{\rightarrow}w$. Тогда $\mathrm{SDOM}(u)\leqslant\mathrm{SDOM}(w)$ и $\mathrm{IDOM}(u)=\mathrm{IDOM}(w)$.

Доказательство. Пусть z — вершина, для которой $v\overset{*}{\rightarrow}z\rightarrow w$. Тогда $\mathrm{SDOM}(u)\leqslant\mathrm{SDOM}(z)\leqslant v=\mathrm{SDOM}(w)$. По лемме 14.10 $\mathrm{IDOM}(w)$ — предок v и, следовательно, собственный предок u. Таким образом, по лемме 14.11 $\mathrm{IDOM}(w)\overset{*}{\rightarrow}\mathrm{IDOM}(u)$. Чтобы доказать, что $\mathrm{IDOM}(u)=\mathrm{IDOM}(w)$, достаточно показать, что $\mathrm{IDOM}(u)$ доминирует над w.

Рассмотрим произвольный ориентированный путь P из s в w. Пусть x — последняя вершина в P, удовлетворяющая условию $x<\mathrm{IDOM}(u)$. Если не существует такой вершины x, то $\mathrm{IDOM}(u)=s$ доминирует над w. Иначе пусть y — первая вершина, следующая за x в P и удовлетворяющая условию $\mathrm{IDOM}(u)\overset{*}{\rightarrow}y\overset{*}{\rightarrow}w$. Как и в доказательстве теоремы 14.15, мы можем показать, используя лемму 14.7, что $\mathrm{SDOM}(y)\leqslant x$. Так как по лемме 14.10 $\mathrm{IDOM}(u)\leqslant\mathrm{SDOM}(u)$, то мы имеем $\mathrm{SDOM}(y)\leqslant x<\mathrm{IDOM}(u)\leqslant\mathrm{SDOM}(u)$. Следовательно, $\mathrm{SDOM}(y)<\mathrm{SDOM}(u)$.

Так как u — минимальный полудоминатор среди вершин пути в дереве из z в w, то y не может быть и собственным потомком v. Более того, y не может быть ни собственным потомком $\mathrm{IDOM}(u)$, ни предком u, поскольку в этом случае ориентированный путь, состоящий из пути в дереве из s в $\mathrm{SDOM}(y)$ с последующим таким путем $\mathrm{SDOM}(y)=v_0, v_1, v_2, \ldots, v_k$, что $v_i>y$ для $1\leqslant i\leqslant k-1$, и следующим затем путем в дереве из y в u не включал бы $\mathrm{IDOM}(u)$. Однако пути из s в u, не включающего $\mathrm{IDOM}(u)$, не существует.

Единственной остающейся возможностью является то, что $\mathrm{IDOM}(u)=y$. Таким образом, $\mathrm{IDOM}(u)$ лежит на ориентированном пути P из s в w. Так как путь выбран произвольно, то $\mathrm{IDOM}(u)$ доминирует над w.

Следующий результат является непосредственным следствием теорем 14.15 и 14.16.

Теорема 14.17. Пусть $w\neq s$ и $v=\mathrm{SDOM}(w)$. Пусть u — вершина, для которой SDOM является минимальной среди вершин u, удовлетворяющих условию $v\overset{+}{\rightarrow}u\overset{*}{\rightarrow}w$. Тогда

$$\mathrm{IDOM}(w)=\begin{cases} v, & \text{если } \mathrm{SDOM}(u)=\mathrm{SDOM}(w),\\ \mathrm{IDOM}(u) & \text{в других случаях.}\end{cases}$$

Сейчас мы уже готовы описать алгоритм доминаторов Ленгауэра и Тарьяна. Главными шагами в этом алгоритме являются следующие действия:

Алгоритм 14.9. (Доминаторы Ленгауэра и Тарьяна.)

S1. Выполнить ПВГ на данном графе программы $G=(V, E)$ с начальной вершиной в качестве корня.

S2. Вычислить полудоминаторы всех вершин, применяя теорему 14.14. Провести вычисления для всех вершин последовательно в порядке увеличения глубин.

S3. Неявно определить непосредственные доминаторы для каждой вершины, применяя теорему 14.17.
S4. Явно выделить непосредственные доминаторы для каждой вершины, проводя вычисления в порядке возрастания глубины вершин.

Реализация шага S1 тривиальна. В последующем мы будем обозначать вершины их глубинами, полученными на шаге S1.

При описании шагов S2 — S4 мы используем массивы: FATHER (определенный так же, как и в разд. 14.3), SEMI, BUCKET и DOM, определенные ниже.

SEMI (w). 1) До вычисления полудоминатора w имеем SEMI (w)=w. 2) После вычисления полудоминатора w SEMI (w)=SDOM (w).

BUCKET (w). Это — множество вершин, полудоминатором которых является w.

DOM (w). 1) Если после шага S 3 полудоминатор w есть ее непосредственный доминатор, то DOM (w) есть непосредственный доминатор w, иначе DOM (w) есть такая вершина v, что $v<w$ и непосредственный доминатор v является также непосредственным доминатором w. 2) После шага S 4 DOM (w) есть непосредственный доминатор w. После выполнения шага S1 алгоритм проходит шаги S2 и S3 одновременно, обрабатывая вершины $w\neq 1$ в порядке уменьшения их глубин. При этом во время вычисления алгоритм поддерживает лес, содержащийся в дереве ПВГ графа G. Лес состоит из множества вершин V и множества ребер {(FATHER (w), w)|вершина w уже обработана}. Алгоритм использует процедуры построения леса и выделения из него информации.

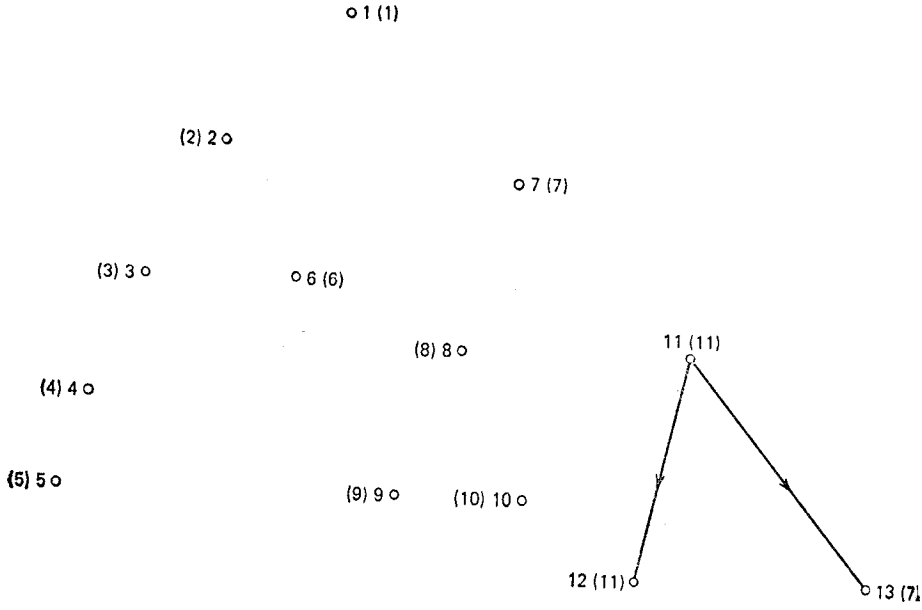
Этими процедурами являются LINK (v, w). Добавить ребро (v, w) в лес.

EVAL (v). 1) Если v — корень дерева в лесу, то EVAL (v) равно v. 2) Иначе, пусть r — корень дерева в лесу, которое содержит v, и пусть u — вершина, для которой SEMI (u) — минимальная среди вершин, удовлетворяющих условию $r \overset{+}{\to} u \overset{*}{\to} v$, тогда EVAL ($v$)=$u$.

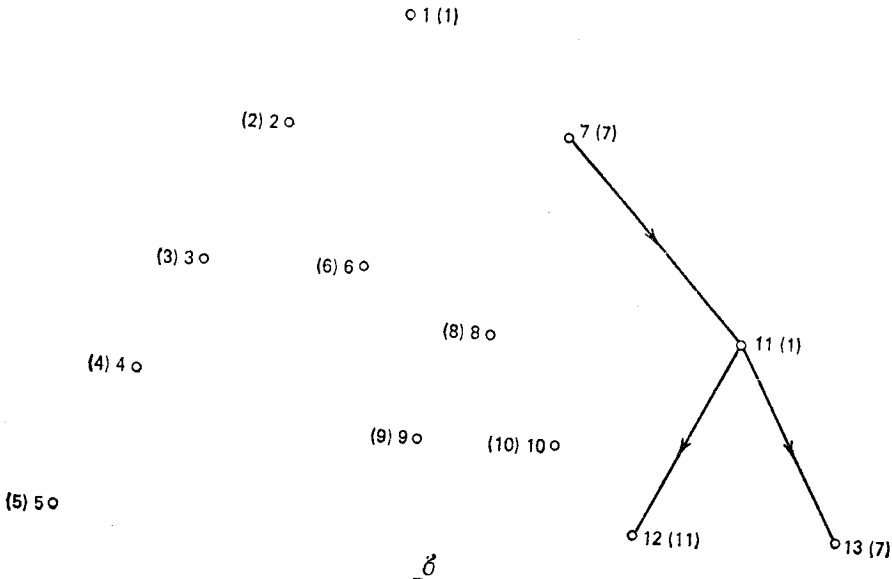
Для обработки вершины w алгоритм вычисляет полудоминатор w, применяя для этого теорему 14.14. Таким образом, алгоритм принимает вид SEMI (w)=min {SEMI ($EVAL\{v\}$)|(v, w) $\in E$}.

По завершении вычисления SEMI (w) есть полудоминатор w. Это следует из теоремы 14.14 и определения EVAL (v). Рассчитав SEMI (w), алгоритм добавляет w в BUCKET (SEMI (w)) и добавляет новое ребро в лес, используя LINK (FATHER (w), w). На этом шаг S2 для w завершается. Затем алгоритм выполняет шаг S3, рассматривая каждую вершину из BUCKET (FATHER (w)).

Пусть v — такая вершина. Алгоритм неявно определяет непосредственный доминатор v с помощью теоремы 14.17. Пусть u=EVAL (v), тогда u есть вершина, для которой выполняется FATHER (w) $\overset{+}{\to}$


1 (1)
(2) 2
7 (7)
(3) 3
6 (6)
(8) 8
11 (11)
(4) 4
(5) 5
(9) 9
(10) 10
12 (11)
13 (7)

а


1 (1)
(2) 2
7 (7)
(3) 3
(6) 6
(8) 8
11 (1)
(4) 4
(9) 9
(10) 10
(5) 5
12 (11)
13 (7)

б

Рис. 14.23.

$\overset{+}{\rightarrow} u \overset{*}{\rightarrow} v$ и полудоминатор которой минимален. Если SEMI (u)=SEMI (v), то FATHER (w) есть непосредственный доминатор v и алгоритм полагает DOM (v)=FATHER (w). Иначе u и w имеют один и тот же непосредственный доминатор и алгоритм полагает DOM (v)=u. Этим завершается шаг S3 для v.

На шаге S4 алгоритм просматривает вершины в порядке возрастания их глубин, выделяя непосредственные доминаторы, неявно вычисленные на шаге S3. Таким образом, шаг S4 включает в себя следующие действия:

Для каждого $i=2, 3, \ldots, n$, если DOM (i)$\neq$SEMI (i), DOM (i)= =DOM (DOM (i)).

Для иллюстрации работы алгоритма доминаторов рассмотрим граф программы, изображенный на рис. 14.22, *а*. Лес, полученный перед обработкой вершины 11, также изображен на рис. 14.23, *а*. Элементы массива SEMI на этой стадии указаны в скобках следом за соответствующими вершинами. Рассмотрим вершину 11. Ребра (1, 11) и (7, 11) заходят в вершину 11. Поэтому SEMI (11)=min {SEMI (EVAL (1)), SEMI (EVAL (7))}.

EVAL (1)=1, так как вершина 1 является корнем дерева в лесу. По той же причине EVAL (7)=7. Таким образом, SEMI (11)=min {SEMI (1), SEMI (7)}=min {1, 7}=1. Алгоритм добавляет ребро (7, 11) в лесу и вершину 11 в BUCKET (SEMI (11))=BUCKET (1). Новый лес с элементами массива SEMI представлен на рис. 14.23, *б*. На этом завершается шаг *S*2 для вершины 11.

Алгоритм рассматривает BUCKET (FATHER (11))=BUCKET (7). Вершина 13 является единственной вершиной, полудоминатор которой равен 7. Поэтому BUCKET (7)={13}. EVAL (13)=11, так как SEMI (11) является минимальной среди вершин u, для которых выполняется $7 \overset{*}{\rightarrow} u \overset{*}{\rightarrow} 13$ (рис. 14.23, *б*). Так как SEMI (13)$\neq$ SEMI (11), то алгоритм устанавливает DOM (13)=11. На этом завершается шаг S 3 для вершины 11.

Вершина	*Массив* DOM	*Полудоминатор* (SEMI)
2	1	1
3	2	2
4	2	2
5	1	1
6	2	2
7	1	1
8	7	7
9	1	1
10	1	1
11	1	1
12	11	11
13	11	**7**

После того как будут выполнены шаги S2 и S 3 для всех вершин $w \neq 1$, будут доступны и полудоминаторы всех вершин. Значения элементов массива DOM и полудоминаторов представлены выше Для каждой вершины $w \neq 13$ DOM (w)=SEMI (w). До сих пор для всех вершин w, кроме 13, IDOM (w)=DOM (w). Для вершины 13 получаем DOM (13)=DOM (DOM (13))=DOM (11)=1. Поэтому IDOM (13)=1. На этом завершается шаг S4 алгоритма, и мы получаем дерево доминаторов, представленное на рис. 14.22, *б*.

Ясно, что сложность приведенного выше алгоритма существенно зависит от реализации инструкции LINK и EVAL. Тарьян [14.44] обсуждает два метода реализации, которые используют сжатие пути. Один из них описывается ниже.

Чтобы представить лес, построенный при выполнении инструкций типа LINK, алгоритм использует два массива: LABEL и ANCESTOR. Вначале ANCESTOR $(v)=0$ и LABEL $(v)=v$ для каждой вершины v. В общем случае ANCESTOR $(v)=0$, только если v — корень дерева в лесу, иначе ANCESTOR (v) является предком v в лесу.

Алгоритм поддерживает метки так, что они удовлетворяют следующему свойству: пусть v — произвольная вершина, а r — корень дерева в лесу, содержащем v, и пусть $v=v_k, v_{k-1}, \ldots, v_0=r$ будут такими, что ANCESTOR $(v_i)=v_{i-1}$ для $1 \leqslant i \leqslant k$. Пусть x — вершина, для которой SEMI (x) — минимальная среди вершин $x \in \{\text{LABEL}(v_i) \mid 1 \leqslant i \leqslant k\}$. Тогда справедливо следующее свойство: x — вершина, для которой SEMI (x) — минимальная вершина среди вершин x, удовлетворяющих условию $r \overset{+}{\rightarrow} x \overset{*}{\rightarrow} v$. Чтобы выполнить LINK (v, w), алгоритм полагает ANCESTOR $(w)=v$. Чтобы выполнить EVAL (v), алгоритм следует за указаниями на предка и определяет такую последовательность $v=v_k, v_{k-1}, \ldots, v_0=r$, что ANCESTOR $(v_i)=v_{i-1}$ для $1 \leqslant i \leqslant k$. Если $v=r$, то алгоритм устанавливает EVAL $(v)=v$. Иначе алгоритм полагает ANCESTOR $(v_i)=r$ для $2 \leqslant i \leqslant k$, одновременно изменяя метки следующим образом (чтобы поддерживать упомянутое ранее свойство):

Если SEMI (LABEL) $(v_{i-1}))<$SEMI (LABEL (v_i)), то LABEL$(v_i)=$ =LABEL (v_{i-1}). Затем алгоритм устанавливает EVAL $(v)=$ =LABEL (v).

Тарьян [14.44] показал, что сложность реализации $(n-1)$ инструкций LINK и $(m+n-1)$ инструкций EVAL с помощью описанного ранее метода равна $O(m \log n)$. Если мы используем более изощренную реализацию инструкций LINK и EVAL, которая также описана в работе [14.44], то алгоритм потребовал бы для своей реализации $O(m\, \alpha\, (m, n))$ шагов, где $\alpha(m, n)$ — обращение функции Аккермана, определенное в предыдущем разделе.

Другие алгоритмы доминаторов представлены в работах [14.45, 14.46].

14.7. Замечания, касающиеся литературы

Различные графовые и комбинаторные алгоритмы описаны в работах [14. 40, 14.47—14.53]. Обсуждение методов представления графов имеется в работах [14.40, 14.41].

Время выполнения алгоритмов, описанных в этой книге, ограничено сверху некоторым полиномом от числа вершин и числа ребер графа. Пусть $\mathcal{P}$ — класс всех проблем, которые можно решить алгоритмами с полиномиальной сложностью. Существует большое число проблем, для которых не известно, существуют ли алгоритмы с полиномиальным временем решения. Многие из них можно решить за полиномиальное время недетерминированными алгоритмами. Класс таких проблем обозначается через $\mathcal{NP}$. Проблема является $\mathcal{NP}$-трудной, если детерминированный алгоритм с полиномиальной сложностью для ее решения можно использовать для поиска детерминированного алгоритма с полиномиальной сложностью для решения любой проблемы из $\mathcal{NP}$. $\mathcal{NP}$ — трудная проблема из $\mathcal{NP}$ называется $\mathcal{NP}$-полной. Таким образом, если детерминированный алгоритм с полиномиальной сложностью для какой-либо проблемы из класса $\mathcal{NP}$-полных существует, то такой алгоритм существует для каждой проблемы в $\mathcal{NP}$. В основополагающей работе [14.54] показано, что проблема выполнимости $\mathcal{NP}$-полна. В работе [14.55] демонстрируется $\mathcal{NP}$-полнота большого класса проблем, а в работах [14.56, 14.57] дается элегантное введение в теорию $\mathcal{NP}$-полноты и доказывается $\mathcal{NP}$-полнота нескольких графовых проблем. Упоминания об этом имеются также в работах [14.40, 14.51, 14.57]. В указанных работах имеются сообщения о многих интересных графовых алгоритмах. Некоторые из них перечислены в конце следующей главы.

Упражнения

14.1. Транзитивная редукция ориентированного графа $G=(V, E)$ определяется как граф $G'=(V', E')$ с минимальным числом ребер, транзитивное замыкание которого совпадает с графом G. Сконструируйте алгоритм нахождения транзитивной редукции ориентированного графа. Как этот алгоритм связан с алгоритмом транзитивного замыкания [14.58]?

14.2. Используя ПВГ, сконструируйте алгоритмы для следующих проблем:

а) топологическая сортировка вершин графа;

б) выделение мостов графа;

в) выделение остовного леса;

г) выделение множества базисных циклов и базисных разрезающих множеств;

д) проверка графа на двудольность;

в) проверка, является ли заданное множество ребер разрезом графа.

14.3. Рассмотрим граф G; пусть T — остовный лес ПВГ для графа G. Пусть $\mathrm{ND}(v)$ — число потомков v (включая ее саму). Докажите, что вершина w является потомком v тогда и только тогда, когда выполняется соотношение $\mathrm{DFN}(v) \leqslant \mathrm{DFN}(w) < \mathrm{DFN}(v) + \mathrm{ND}(v)$.

14.4. Модифицируйте алгоритм ПВГ так, чтобы он включал шаги для вычисления $\mathrm{ND}(v)$.

14.5. Пусть T — дерево ПВГ в связном графе G. Пусть G_s— полный подграф графа G. Покажите, что все вершины G_s лежат на одном ориентированном пути в T.
14.6. Пусть T — дерево ПВГ в ориентированном графе G. Покажите, что если C — ориентированный цикл графа G, а v — вершина, входящая в C и имеющая минимальную глубину, то v — предок в T любой вершины цикла C.
14.7. Поиск в ширину (ПВШ) просматривает связный граф G следующим образом:

S1. В начале ни одна из вершин G не помечена.
S2. Выбирать вершину s и пометить ее как 0.
S3. Положить $i=0$.
S4. Пусть S — множество всех непомеченных вершин, смежных по меньшей мере с одной вершиной с меткой i.
S5. Если S — пустое, то STOP, иначе пометить все вершины в S меткой $i+1$.
S6. Положить $i=i+1$ и идти к шагу S4.

Покажите, что ребра, проходимые при пометке вершин, образуют остов графа G.
14.8. Покажите как ПВШ можно использовать для вычисления расстояния от вершины s до всех вершин связного графа G. (Под расстоянием между вершинами u и v мы понимаем длину u—v-пути, включающего наименьшее число ребер.)
14.9. Покажите, что множество обратных ребер сводимого графа программы уникально, т. е. все остовные леса ПВГ имеют одно и то же множество обратных ребер [14.35].
14.10. Используйте результат теоремы 6.10 для построения алгоритма нахождения всех остовов связного графа.
14.11. Пусть T — дерево на n вершинах. Пусть $\{1, 2, \ldots, n\}$ — множество вершин T. Мы можем сопоставить единственную последовательность $(t_1, t_2, \ldots, t_{n-2})$ дереву T следующим образом: пусть s_1— первая вершина степени 1 в T, тогда вершина, смежная s_1, есть первый элемент последовательности t_1. Удалим s_1 из T. Если s_2— первая вершина степени 1 в T—s_1, то вершина, смежная с s_2 в T—s_1, есть t_2. Удалим s_2 и будем повторять операцию до тех пор, пока не будет определена вершина t_{n-2}. Последовательность $(t_1, t_2, \ldots, t_{n-2})$ называется последовательностью Прюфера, связанной с T. Ясно, что два различных дерева имеют различные последовательности Прюфера.

Дана последовательность $(t_1, t_2, \ldots, t_{n-2})$, для которой $t_i \in \{1, 2, \ldots, n\}$. Сконструируйте алгоритм построения дерева T, для которого $(t_1, t_2, \ldots, t_{n-2})$ есть последовательность Прюфера (число $(n$—$2)$ буквенных последовательностей, которые могут быть построены из множества $\{1, 2, \ldots, n\}$, равно n^{n-2}). Каждая такая последовательность является последовательностью Прюфера для остова графа K_n. Поэтому существует взаимно-однозначное соответствие между последовательностями, построенными из каких-либо n—2 букв (не обязательно различных), принадлежащих множеству $\{1, 2, \ldots, n\}$, и остовами графа K_n. Таким образом, число остовов графа K_n равно n^{n-2}. Это доказательство приведено Прюфером [14.59].
14.12. Используйте результат теоремы 9.10 для построения алгоритма нахождения хроматического числа графа.

15. Оптимизационные алгоритмы

В этой главе мы рассмотрим несколько алгоритмов решения оптимизационных задач на графах. Эти задачи возникают в различных приложениях, связанных с исследованием операций и вычислительной техникой. Мы обсудим алгоритмы, касающиеся следующих основных тем:

1. Кратчайшие пути.
2. Оптимальные деревья.
3. Паросочетания в графе.
4. Потоки в сетях.
5. Оптимальные ветвления.

15.1. Кратчайшие пути

Пусть G — связный ориентированный граф, в котором каждому ориентированному ребру сопоставлено положительное вещественное число, называемое длиной ребра. Длина ребра, направленного из вершины i в вершину j, обозначается через $w(i, j)$. Если в графе отсутствует ребро, направленное из вершины i в вершину j, то $w(i, j)=\infty$. Длина ориентированного пути в графе G — это сумма длин ребер, входящих в путь. Ориентированный s — t-путь, имеющий минимальную длину, называется кратчайшим путем из s в t. Длина кратчайшего ориентированного s — t-пути называется расстоянием из s в t и обозначается через $d(s, t)$. Ясно, что $d(i, i)=0$ для любого i.

В этом разделе мы рассмотрим следующие задачи:

1. Найти кратчайшие пути из данной вершины s ко всем другим вершинам графа G.
2. Найти кратчайшие пути между всеми упорядоченными парами вершин графа G.

Эти две задачи возникают в нескольких проблемах оптимизации. Например, определение потока минимальной стоимости в транспортной сети включает определение кратчайшего пути из источника до стока сети [15.1].

15.1.1. Кратчайшие пути из данной вершины S ко всем другим вершинам графа

Рассмотрим алгоритм Дейкстры [15.2], который определяет кратчайшие пути из данной вершины ко всем другим вершинам связного ориентированного графа на n вершинах. Следующее соображение составляет основу алгоритма Дейкстры.

Пусть V — множество вершин графа G, а S — такое подмножество V, что $s \in S$. Обозначим через $\bar{S}$ дополнение S до V. Таким образом, $\bar{S}=V-S$.

Пусть P — путь минимальной длины среди всех ориентированных путей из s к вершинам в $\bar{S}$. Длина пути P называется расстоянием $d(s, \bar{S})$ от s до $\bar{S}$. Пусть $P: s, \ldots, u, v$. Тогда ясно, что $v \in \bar{S}$ и $u \in S$. Далее, часть пути P из s в u должна быть наикратчайшим s — u-путем, вершины которого принадлежат S. Поэтому $d(s, S)=d(s, u)+w(u, v)$. Отсюда видно, что расстояние $d(s, \bar{S})$ можно вычислить по формуле

$$d(s, \bar{S}) = \min_{\substack{u \in S \\ v \in \bar{S}}} \{d(s, u) + w(u, v)\}. \tag{15.1}$$

Тогда если $v \in \bar{S}$ такое, что $d(s, \bar{S})=d(s, u)+w(u, v)$ для некоторого $u \in S$, то

$$d(s, v) = d(s, \bar{S}). \tag{15.2}$$

Алгоритм Дейкстры порождает последовательность $S_0=\{s\}$, S_1, S_2, ... таких подмножеств V, что выполняются следующие условия:

1. Если $s=u_0$, u_1, $u_2, \ldots, u_{n-1}$ — такие вершины из множества V, что $d(s, u_1) \leqslant d(s, u_2) \leqslant d(s, u_3) \leqslant \ldots \leqslant d(s, u_{n-1})$, то $S_i=\{s, u_1, u_2, \ldots, u_i\}$ для $i>0$.

2. Когда множество S_i определено, известны и кратчайшие пути из s в u_1, u_2, ..., u_i. Для множеств S_i, определенных выше соотношением (15.2), справедливо равенство

$$d(s, u_{i+1}) = d(s, \bar{S}_i). \tag{15.3}$$

Таким образом, построение S_{i+1} по S_i включает в себя вычисление $d(s, \bar{S}_i)$. Подмножества S_1, S_2, ..., S_{n-1} можно построить в соответствии с выражением (15.1) следующим образом:

$$d(s, \bar{S}_0) = \min_{\substack{u \in S_0 \\ v \in \bar{S}_0}} \{d(s, u) + w(u, v)\} = \min_{v \in S_0} \{w(s, v)\}.$$

Следовательно, согласно формуле (15.3), u_1 — вершина, для которой выполняется равенство

$$d(s, u_1) = \min_{v \in \bar{S}_0} \{w(s, v)\}. \tag{15.4}$$

Если P_i — кратчайший путь из вершины s к вершине u_i, то P_1: s, u_1.

Предположим, что подмножества S_0, S_1, ..., S_i и пути P_1, P_2, ..., P_i уже определены. Тогда для определения S_{i+1} мы вычис-

ляем $d(s, \bar{S}_i)$, используя выражение (15.1). Из равенства (15.3) следует, что u_{i+1} — вершина в $\bar{S}_i$, обладающая свойством $d(s, u_{i+1}) = d(s, \bar{S}_i)$.

Из выражения (15.1) следует, что существует такая вершина $j \in S_i$, что $d(s, u_{i+1}) = d(s, u_j) + w(u_j, u_{i+1})$. Поэтому мы можем получить P_{i+1}, присоединяя ребро (u_j, u_{i+1}) к пути P_j.

Если нам необходимо получить кратчайший путь только до другой данной вершины t, то мы можем завершить вычисления после получения первого множества S_i, которое содержит t.

Ясно, что в этой процедуре необходимо вычислять минимум в выражении (15.1) на каждом этапе. Если этот минимум действительно вычислялся бы на каждом этапе, то определение S_i по S_{i-1} потребовало бы $(i-1)\times(n-i)$ сложений и $\{i(n-i)-1\}$ сравнений. Общее число операций для выполнения всего алгоритма составило бы $\sum_{i=1}^{n-1}\{(2i-1)\ (n-i)-1\}$, т. е. в общей сложности $O(n^3)$ шагов. Однако многие из этих сложений и сравнений не обязательно должны каждый раз повторяться.

Дейкстра в своем алгоритме устранил такие повторения, сохранив полученную на одном из этапов информацию для последующих этапов. Это достигается процедурой расстановки меток, которая, как мы увидим, уменьшает сложность алгоритма до $O(n^2)$. Следующие соображения приводят нас к процедуре расстановки меток Дейкстры. Предположим, что мы осуществляем расстановку меток так, что для $i=0, 1, 2, \ldots$ метка $l_i(v)$ вершины v удовлетворяет следующим требованиям:

1. $l_0(s)=0$ и $l_0(v)=\infty$ для всех $v \neq S$.

2. Для $i \geqslant 1$ $l_i(v) = d(s, v)$ для всех $v \in S_{i-1}$, $l_i(v) = \min\limits_{u \in S_{i-1}} \{d(s, u) + w(u, v)\}$ для всех $v \in \bar{S}_{i-1}$. Ясно, что u_i — вершина, для которой $d(s, u_i) = d(s, \bar{S}_{i-1}) = \min\limits_{v \in \bar{S}_{i-1}} \{l_i(v)\}$. Теперь мы можем определить $l_{i+1}(v)$ по $l_i(v)$:

1. Для $v \in S_i$, $l_{i+1}(v) = l_i(v) = d(s, v)$.

2. Для $v \in \bar{S}_i$

$$l_{i+1}(v) = \min_{u \in S_i} \{d(s, u) + w(u, v)\} = \min \{l_i(v), d(s, u_i) + w(u_i, v)\} = \min \{l_i(v), l_i(u_i) + w(u_i, v)\}. \quad (15.5)$$

Тогда u_{i+1} выбирается так, что

$$d(s, u_{i+1}) = d(s, \bar{S}_i) = \min_{v \in \bar{S}_i} \{l_{i+1}(v)\}. \quad (15.6)$$

Отметим, что метка u_i не меняется после определения S_i.

Таким образом, алгоритм Дейкстры начинает работу с меток $l_0(s)=0$ и $l_0(v)=\infty$ для всех $v \neq s$. По мере выполнения алгоритма

метки модифицируются в соответствии с выражением (15.5). Метки $l_{n-1}(v)$ определяют расстояние от s до v.

Ясно, что определение u_{i+1} включает вычисление $l_{i+1}(v)$ для всех $v \in \bar{S}_i$ и последующее нахождение минимума среди этих меток. Для $i \geqslant 1$ первое вычисление по формуле (15.5) требует $(n - i - 1)$ сложений и $\{n - i - 1\}$ сравнений, тогда как последние действия по формуле (15.6) требуют $\{(n-i)-2\}$ сравнений. Таким образом, ясно, что сложность алгоритма Дейкстры равна $O(n^2)$.

Представим описание алгоритма Дейкстры. В этом описании LABEL — это массив, в котором хранятся текущие метки вершин. Вершины становятся постоянно помеченными, когда они оказываются равными u_i для какого-либо i. Мы используем массив PERM, чтобы указать, какие вершины постоянно помечены. Если PERM$(v)=1$, то v является постоянно помеченной вершиной. Отметим, что в таком случае метка v равна $d(s, v)$. Вначале PERM$(s)=1$ и PERM $(v)=0$ для всех $v \neq s$.

PRED — массив указателей на вершины, из которых осуществлен переход в вершины с неизменной меткой. Если вершина v помечена неизменной меткой, то v, PRED(v), PRED(PRED (v)), . . ., s есть вершины, составляющие кратчайший ориентированный путь из s в v.

Алгоритм 15.1. Кратчайшие пути (Дейкстра).

S0. G — данный ориентированный граф с взвешенными ребрами. Требуется найти кратчайшие пути из вершины s во все остальные вершины графа G.

S1. (Начало.) Положить LABEL $(s)=0$, PERM $(s)=1$ и PRED $(s)=s$. Для всех $v \neq s$ положить LABEL $(v)=\infty$, PERM $(v)=0$, PRED $(v)=v$.

S2. Пусть $i=0$ и $u=s$. (u — последняя из вершин с неизменной меткой. Теперь — это вершина s.)

S3. (Вычисление LABEL и изменение элементов массива PRED). Положить $i=i+1$. Выполнить для каждой вершины, кроме вершин с неизменной меткой, следующие действия: 1) Положить $M=\min\{\text{LABEL}(v), \text{LABEL}(u)+w(u, v)\}$. 2) Если $M<LABEL(v)$, то положить LABEL $(v)=M$ и $PRED(v)=u$.

S4. (Выделение вершины u_i.) Среди всех вершин, которые не помечены неизменной меткой, найти вершину w с наименьшей меткой. (Если таких вершин несколько, то выбор можно сделать произвольно.) Положить PERM $(w)=1$ и $u_i=w$ ($u_i=w$, и она является последней вершиной с неизменной меткой.)

S5. Если $i<n-1$, то идти к шагу S3. Иначе HALT. (Все кратчайшие пути найдены). Метки вершин представляют собой длины кратчайших путей. v, PRED (v), PRED (PRED (v)), . . ., s есть вершины кратчайшего ориентированного s—v-пути.

Отметим, что в программе для вычислительной машины ∞ представляется достаточно большим числом. Если конечная метка вершины v равна ∞, то это означает, что не существует ориентированного пути из s в v.

Чтобы проиллюстрировать работу алгоритма Дейкстры, рассмотрим граф на рис. 15.1, на котором длины ребер указаны рядом. На рис. 15.2 изображены элементы массивов LABEL и PRED.

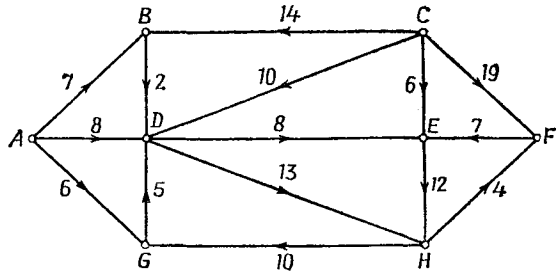


Рис. 15.1.

	Вершины							
i	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*
0	⓪*	∞	∞	∞	∞	∞	∞	∞
1	⓪	7	∞	8	∞	∞	⑥*	∞
2	⓪	⑦*	∞	8	∞	∞	⑥	∞
3	⓪	⑦	∞	⑧*	∞	∞	⑥	∞
4	⓪	⑦	∞	⑧	⑯*	∞	⑥	21
5	⓪	⑦	∞	⑧	⑯	∞	⑥	㉑*
6	⓪	⑦	∞	⑧	⑯	㉕*	⑥	㉑
7	⓪	⑦	∞*	⑧	⑯	㉕	⑥	㉑

а

PRED(*A*) = *A* PRED(*E*) = *D*
PRED(*B*) = *A* PRED(*F*) = *H*
PRED(*C*) = *C* PRED(*G*) = *A*
PRED(*D*) = *A* PRED(*H*) = *D*

O_T	K	*Кратчайший путь*
A	*B*	*A, B*
A	*C*	*Нет пути*
A	*D*	*A, D*
A	*E*	*A, D, E*
A	*F*	*A, D, H, F*
A	*G*	*A, G*
A	*H*	*A, D, H*

б

Рис. 15.2. Иллюстрация к алгоритму Дейкстры (массив LABEL изображен на рис. *а*).

Для любого i элементы массива LABEL, обведенные кружком, соответствуют вершинам, помеченным неизменной меткой, а именно вершинам из множества S_i. Элементы, помеченные звездочкой, являются метками последних вершин, помеченных неизменной меткой, а именно вершин u_i. Кратчайшие пути из s и соответствую-

щие расстояния получаются из конечных значений элементов массивов PRED и LABEL.

При обсуждении алгоритма мы предполагали, что все длины неотрицательны. Алгоритм Дейкстры неприменим, если какие-либо из длин отрицательны (почему?). Однако некоторая модификация этого алгоритма делает его применимым для более общего случая, когда в графе нет ориентированных циклов с отрицательной длиной. Джонсон [15.3] показал, что в худшем случае этот модифицированный алгоритм имеет сложность $O(n \cdot 2^n)$, а не $O(n^3)$, как было показано в работе [15.4].

Иногда может возникнуть необходимость получить не самые короткие пути. Такая и связанные с ней задачи обсуждаются в работах [15.5—15.10]. Алгоритмы, предназначенные для разреженных графов, приведены в работах [15.11, 15.12].

15.1.2. Кратчайшие пути между всеми парами вершин

Предположим, что необходимо найти кратчайшие пути между всеми $n(n-1)$ упорядоченными парами вершин в ориентированном графе на n вершинах. Простым решением этой задачи было бы n-кратное применение алгоритма Дейкстры. Однако существуют более эффективные алгоритмы по сравнению с этим решением. Они применимы даже тогда, когда в графе присутствуют ребра отрицательной длины, но не существует ориентированных циклов отрицательной длины. Рассмотрим один из этих алгоритмов. Он предложен Флойдом [15.13] и основан на процедуре Уоршолла (алгоритм 14.1) для вычисления транзитивного замыкания.

Пусть дан ориентированный граф G на n вершинах, ребра которого помечены длинами. Пусть вершины графа обозначаются числами $1, 2, \ldots, n$. Предположим, что в графе не существует ориентированного цикла с отрицательной длиной. Пусть $W=[w_{ij}]$ — $n \times n$-матрица длин ребер графа G, т. е. w_{ij} — длина ориентированного ребра (i, j) в графе G. Полагаем $w_{ij}=\infty$, если в графе нет ребра (i, j), ориентированного из i в j. Мы полагаем также, что $w_{ii}=0$ для всех i.

Начиная с матрицы $W^{(0)}$ алгоритм Флойда строит последовательность $W^{(1)}, W^{(2)}, \ldots, W^{(n)}$ таких $n \times n$-матриц, что элемент $w_{ij}^{(n)}$ матрицы $W^{(n)}$ есть расстояние между i и j в графе G. Матрица $W^{(k)}=[w_{ij}^{(k)}]$ определяется по матрице $W^{(k-1)}=[w_{ij}^{(k-1)}]$ в соответствии со следующим правилом:

$$w_{ij}^{(k)} = \min \{w_{ij}^{(k-1)}, \quad w_{ik}^{(k-1)} + w_{kj}^{(k-1)}\}. \tag{15.7}$$

Пусть $P_{ij}^{(k)}$ — путь минимальной длины среди всех ориентированных i —j-путей, которые используют в качестве внутренних вершин только вершины из множества $\{1,2, \ldots, k\}$. Следующая теорема доказывает корректность алгоритма Флойда.

Теорема 15.1.. Для $0 \leqslant k \leqslant n$ $w_{ij}^{(k)}$ есть длина пути $P_{ij}^{(k)}$.
Доказательство. Аналогично доказательству теоремы 14.1.

Обычно кроме кратчайших длин нам необходимо также получить сами пути с такими длинами. Напомним, что в алгоритме Дейкстры мы использовали массив PRED, чтобы хранить указание на вершины, встречающиеся в кратчайших путях. Это достигается в алгоритме Флойда следующим образом: по мере построения последовательности $W^{(0)}$, $W^{(1)}$, . . ., $W^{(n)}$ мы одновременно строим такую другую последовательность матриц $Z^{(0)}$, $Z^{(1)}$, . . ., $Z^{(n)}$, что элемент $z_{ij}^{(k)}$ матрицы $Z^{(k)}$ указывает на вершину, которая непосредственно следует за вершиной i в $P_{ij}^{(k)}$. Ясно, что в этом случае

$$z_{ij}^{(0)} = \begin{cases} j, & \text{если } w_{ij} \neq \infty; \\ 0, & \text{если } w_{ij} = \infty. \end{cases} \tag{15.8}$$

$Z^{(k)} = [z_{ij}^{(k)}]$ получается из $Z^{(k)} = [z_{ij}^{(k-1)}]$ в соответствии со следующим правилом: пусть $M = \min \{w_{ij}^{(k-1)}, w_{ik}^{(k-1)} + w_{kj}^{(k-1)}\}$, тогда

$$z_{ij}^{(k)} = \begin{cases} z_{ij}^{(k-1)}, & \text{если } M = w_{ij}^{(k-1)}, \\ z_{ik}^{(k-1)}, & \text{если } M < w_{ij}^{(k-1)}. \end{cases} \tag{15.9}$$

Это правило аналогично правилу, применяемому на шаге S3 алгоритма 15.1 для изменения массива PRED. Правомочность этого правила определяется следующим. Если $M = w_{ij}^{(k-1)}$, то Длина $P_{ij}^{(k)}$ равна длине $P_{ij}^{(k-1)}$. Поэтому $z_{ij}^{(k)}$ совпадает с $z_{ij}^{(k-1)}$.

С другой стороны, если $M < w_{ij}^{(k-1)}$, то $P_{ij}^{(k)}$ — конкатенация путей $P_{ik}^{(k-1)}$ и $P_{kj}^{(k-1)}$. Поэтому $z_{ij}^{(k)} = z_{ik}^{(k-1)}$. Очевидно, что кратчайший i—j-путь определяется последовательностью вершин i, i_1, i_2, ..., i_p, j, где

$$i_1 = z_{ij}^{(n)}, \quad i_2 = z_{i_1 j}^{(n)}, \quad i_3 = z_{i_2 j}^{(n)}, \ldots, \quad j = z_{i_p j}^{(n)}. \tag{15.10}$$

Отметим, что в формуле (15.7) $w_{ij}^{(k)} = w_{ij}^{(k-1)}$, если $w_{ik}^{(k-1)}$ или $w_{kj}^{(k-1)}$ равны ∞. Это простое замечание используется при дальнейшем описании алгоритма Флойда. Мы добавили также в этот алгоритм действия по проверке на присутствие в графе ориентированного цикла с отрицательной длиной.

Алгоритм 15.2. Кратчайшие пути между всеми парами вершин (Флойд).
S1. $W = [w_{ij}]$ — $n \times n$-матрица длин ребер в данном ориентированном графе G. Положить $w_{ii} = 0$ для всех $i = 1, 2, \ldots, n$. $Z = [z_{ij}]$ — $n \times n$-матрица, в которой

$$z_{ij} = \begin{cases} j, & \text{если } w_{ij} \neq \infty; \\ 0, & \text{если } w_{ij} = \infty. \end{cases}$$

S2. Пусть $k = 0$.
S3. Пусть $k = k + 1$. Для всех таких $i \neq k$, что $w_{ik} \neq \infty$, и всех таких $j \neq k$, что $w_{kj} = \infty$, выполнить следующие действия:
1) положить $M = \min \{w_{ij}, w_{ik} + w_{kj}\}$;
2) если $M < w_{ij}$, то положить $z_{ij} = z_{ik}$ и $w_{ij} = M$.

S4. 1. Если какой-либо элемент матрицы $w_{ii}<0$, то вершина i принадлежит некоторому ориентированному циклу с отрицательной длиной. HALT.
2. Если все $w_{ii}\geqslant 0$ и $k=n$, то $[w_{ij}]$ — длины всех кратчайших путей, а z_{ij} — первая вершина после i в кратчайшем i—j-пути. HALT.
3. Если все $w_{ii}\geqslant 0$, но $k<n$, то идти к шагу S3.

Алгоритм Флойда является наиболее эффективным из известных алгоритмов выделения всех кратчайших путей. Он применим к графам с отрицательными длинами. Для ознакомления с другим эффективным алгоритмом выделения всех кратчайших путей см. работу [15.14].

В работе [15.15] предлагается алгоритм выделения всех кратчайших путей. Позже в работе [15.16] было указано на наличие в этой работе ошибки и представлен исправленный вариант. В работе [15.17] приводится исчерпывающая библиография по алгоритмам выделения кратчайших путей.

15.2. Деревья с минимальной длиной взвешенных путей

m-деревом называется ориентированное дерево, в котором полустепень исхода каждой вершины не превышает m. Напомним, что в ориентированном дереве полустепень захода для корня равна 0, а для остальных вершин — единице. Обычно m-дерево изображают так, что корень находится вверху, а все ребра направлены вниз. Все вершины, находящиеся на одном и том же расстоянии от корня, лежат на одной горизонтальной линии. 3-дерево изображено на рис. 15.3.

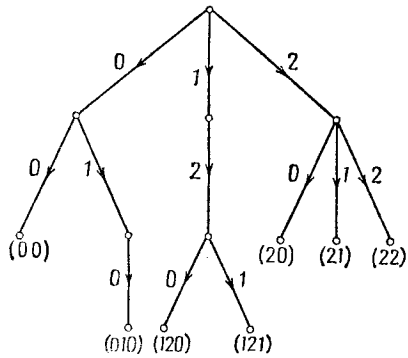


Рис. 15.3. 3-дерево.

Пусть T — m-дерево, а $M=\{0, 1, 2, \ldots, m-1\}$ — алфавит, состоящий из m букв. Предположим, что мы приписываем каждому ребру в дереве T букву из алфавита M так, что не может быть двух ребер, исходящих из одной и той же вершины и помеченных одной и той же буквой. Тогда мы можем приписать каждой вершине дерева T слово, образуемое конкатенацией букв, которыми помечены ребра, встречающиеся при движении из корня в данную вершину. Такие слова, приписанные листьям дерева T, называются *кодовыми словами*, и говорят, что они образуют *префиксный код*. Например, в случае тернарного дерева, изображенного на рис. 15.3, кодовыми словами являются 00, 010, 120, 121, 22, 20, 21.

Интересным и полезным свойством этих слов в префиксном коде является то, что никакое из слов не совпадает с началом другого. Предположим, что мы закодировали L сообщений M_1, M_2, . . ., M_L словами префиксного кода. Если мы передадим последовательность каких-либо из этих закодированных сообщений по каналам связи, то мы получим на принимающем конце канала последовательность букв, образуемую конкатенацией кодовых слов, соответствующих переданным сообщениям. Чтобы из этой последовательности выделить сообщения, необходимо разложить ее на слова из префиксного кода. Процесс разложения последовательности на кодовые слова называется *декодированием*, и его можно осуществить с помощью дерева, соответствующего префиксному коду. Например, рассмотрим последовательность 120202200, которая образована конкатенацией некоторых слов префиксного кода, соответствующего дереву, изображенному на рис. 15.3. Чтобы декодировать эту последовательность, рассмотрим составляющие ее буквы слева направо. В соответствии с этим просмотром мы двигаемся по дереву, начиная с корня вдоль ребер, которые соответствуют рассматриваемым буквам, до тех пор пока мы не дойдем до листа дерева. Кодовое слово, соответствующее этому листу, является первым словом в данной последовательности. Таким образом, мы получаем 120 в качестве первого слова в последовательности 120202200. Затем мы повторяем процесс декодирования для оставшейся последовательности 202200 и выделяем 20, 22, 00 в качестве второго, третьего и четвертого кодовых слов в последовательности 120202200.

Из процесса декодирования, описанного выше, ясно, что стоимость декодирования кодового слова пропорциональна числу букв в этом слове. Если w_i — частота появления сообщения M_i, то ожидаемая стоимость декодирования зависит от суммы $\sum_{i=1}^{L} w_i l_i$, где l_i — длина пути от корня до листа, соответствующего сообщению M_i. Сумма $\sum_{i=1}^{L} w_i l_i$ называется *длиной взвешенных путей* дерева. Таким образом, ожидаемую стоимость декодирования можно свести к минимуму таким выбором длин кодовых слов, что полученное дерево будет иметь минимальную длину взвешенных путей. Это рассуждение приводит нас к задаче построения m-дерева с минимальной длиной взвешенных путей по заданному множеству весов w_1, w_2, . . ., w_L. Прежде чем рассмотреть решение этой задачи, обсудим другой случай, в котором эта задача встречается.

Пусть нам дано L списков S_1, S_2, . . ., S_L. Предположим, что каждый список состоит из целых чисел, расположенных в неубывающем порядке. Мы хотели бы слить эти списки в единый список, элементы которого также располагались в неубывающем порядке. Чтобы выполнить это, мы можем слить, во-первых, любые два из этих списков, например S_1 и S_2, и получить новый список S'. Затем мы можем слить любые два списка из множества $\{S', S_3, S_4, \ldots, S_L\}$ и продолжить слияние до тех пор, пока не будет получен один

список. Для описания такого способа слияния можно использовать бинарное дерево.

Например, рассмотрим дерево, изображенное на рис. 15.4, которое описывает порядок слияния пяти списков S_1, S_2, S_3, S_4, S_5. Вначале мы сливаем списки S_1 и S_2, чтобы получить S_1'. Затем сливаются списки S_1' и S_3, чтобы получить S_2'. На следующем шаге сливаются списки S_4 и S_5, и мы получаем S_3'. Наконец сливаются списки S_2' и S_3', в результате чего получается один требуемый список.

Одним из путей слияния двух произвольных списков является следующий способ. Возьмем первые элементы, т. е. минимальные элементы из двух данных списков. Считаем наименьший из этих элементов первым элементом нового списка. Элемент, выбранный таким образом, удаляем из исходного списка. Повторяем эту операцию над теми же самыми двумя списками, один из которых стал короче, до тех пор, пока они не сольются в единый список.

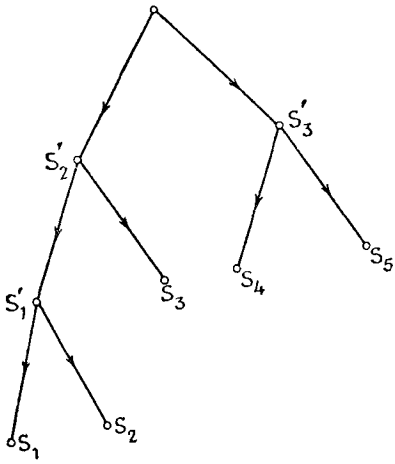


Рис. 15.4.

Ясно, что стоимость слияния любых двух списков пропорциональна общему числу элементов в списках. Поэтому стоимость слияния списков S_1, S_2, . . ., S_L равна $\Sigma_{i=1}|S_i| \cdot l_i$, где l_i — число процессов слияния, в которых участвуют элементы списка S_i.

Например, стоимость слияния списков S_1, S_2, . . ., S_5 в соответствии с деревом, изображенным на рис. 15.4, равна $3|S_1|+$ $+3|S_2|+2|S_3|+2|S_4|+2|S_5|$.

Из рис. 15.4 видно, что в этом случае l_i фактически равно длине пути из корня дерева в лист, соответствующий S_i. Таким образом, задача минимизации стоимости слияния списков S_1, S_2, . . ., S_L эквивалентна задаче построения бинарного дерева с минимальной длиной взвешенных путей для весов $|S_1|$, $|S_2|$, . . ., $|S_L|$.

В работе [15.18] приводится элегантное решение рассмотренной выше задачи. Остальная часть этого раздела посвящена алгоритму Хаффмана. Рассмотрим m-дерево T с L листьями. Пусть l_1, l_2, . . ., l_L — длины путей из корня в листы дерева T. Назовем l_i *длиной пути* соответствующего листа, а вектор $\lambda = (l_1, l_2, \ldots, l_L)$ — вектором длин путей дерева T. Определим характеристическую сумму $S(\lambda, m)$ вектора λ следующим образом:

$$S(\lambda, m) = \sum_{i=1}^{L} m^{-l_i}. \qquad (15.11)$$

Определим необходимые и достаточные условия для того, чтобы вектор λ был вектором длин путей m-дерева. При этом мы предполагаем, что каждое l_i — положительное число.

Лемма 15.1. Характеристическая сумма вектора длин путей $\lambda=(l_1, l_2, \ldots, l_L)$ m-дерева меньше или равна единице.

Доказательство проведем индукцией по максимальной длине в векторе λ. Если максимальная длина в λ равна единице, то все l_i равны единице. Очевидно, что в этом случае лемма справедлива. Пусть лемма верна для каждого вектора длин путей, в котором максимальная длина меньше k. Предположим, что максимальная длина в λ равна k. Пусть в соответствующем m-дереве T вершины v_1, $v_2, \ldots, v_{d^+(r)}$ являются сыновьями корня r. Отметим, что полустепень исхода $d^+(r)$ меньше или равна m. Пусть T_i — поддерево дерева T, имеющего v_i в качестве корня. Ясно, что длины путей в поддереве T_i короче не единицу соответствующих путей в T. Таким образом, если S_i — характеристическая сумма вектора длин путей дерева T_i, то $S(\lambda, m)=(1/m)(S_1+S_2+\ldots+S_{d^+(r)})$. Так как $d^+(r)\leqslant m$ и по предположению индукции $S_i\leqslant 1$ для всех i, то $S(\lambda, m)\leqslant 1$.

Лемма 15.2. Вектор $\lambda=(l_1\ l_2, \ldots, l_L)$, где каждое l_i — положительное число, является вектором длин путей m-дерева, если он удовлетворяет условию

$$\sum_{i=1}^{L} m^{-l_i}\leqslant 1. \tag{15.12}$$

Доказательство. Расположим длины l_i таким образом, что $l_1\leqslant l_2\leqslant\ldots<l_L$. Пусть μ — число длин, равных l_L. Тогда $l_{L-\mu}<l_{L-\mu+1}=l_{L-\mu+2}=\ldots=l_L$. Умножим выражение (15.12) на m^{l_L} и получим

$$\sum_{i=1}^{L-\mu} m^{l_L-l_i}+\mu\leqslant m^{l_L}. \tag{15.13}$$

Пусть $\mu=\sigma m+\rho$, где σ и ρ — неотрицательны, причем $0\leqslant\rho\leqslant m-1$, и определим π следующим образом:

$$\pi=\begin{cases} 0, & \text{если } \rho=0,\\ m-\rho, & \text{если } \rho>0.\end{cases}$$

Тогда мы можем добавить π к левой части выражения (15.13), не нарушая неравенства. Таким образом,

$$\sum_{i=1}^{L-\mu} m^{l_L-l_i}+\mu+\pi\leqslant m^{l_L}. \tag{15.14}$$

Отметим, что $\mu+\pi=\tau m$, где

$$\pi=\begin{cases} \sigma, & \text{если } \rho=0,\\ \sigma+1, & \text{если } \rho>0.\end{cases} \tag{15.15}$$

Тогда, разделив выражение (15.14) на m^{l_L}, получим

$$\sum_{i=1}^{L-\mu} m^{-l_i}+\tau m^{-(l_L-1)}\leqslant 1. \tag{15.16}$$

Значение левой части неравенства является характеристической суммой вектора длиной $L-\mu+\tau$, в которой максимальная длина короче на единицу максимальной длины в данном векторе λ. Докажем лемму индукцией по величине максимальной длины. Если $l_L=1$, то $l_i=1$ для всех i и из выражения (15.12) следует, что $L\leqslant m$.

Ясно, что лемма верна в этом случае, поскольку λ — вектор длин путей m-дерева, содержащего L ребер, каждое из которых выходит из корня.

Затем предположим, что лемма верна для всех векторов, в которых максимальная длина меньше k.

Пусть максимальная длина в λ равна k. Как мы уже упоминали ранее, мы можем построить вектор λ' длиной $L-\mu+\tau$, причем максимальная длина равна $k-1$. В λ' существует по меньшей мере τ длин, равных $k-1$, а остальные $L-\mu$ длин вектора λ' совпадают с первыми $L-\mu$ длинами вектора λ. Отметим, что $\tau=\sigma$ либо $\sigma+1$.

Так как λ' удовлетворяет условию (15.16), то по предположению индукции следует, что существует m-дерево T' с λ' в качестве вектора длин путей. Рассмотрим любые τ листьев дерева T' с длинами путей, равными $k-1$. Если мы присоединим m ребер к каждой из каких-либо σ выделенных вершин и ρ ребер к оставшейся вершине, если она существует, то мы получим новое m-дерево T, которое имеет $\mu=\sigma m+\rho$ листьев с длинами путей, равными k. Остальные длины путей совпадают с первыми $L-\mu$ длинами путей λ. Таким образом, λ — вектор длин путей дерева T.

Вектор длин путей $\lambda=(l_1, l_2, \ldots, l_L)$ называется оптимальным для вектора весов $W=(w_1, w_2, \ldots, w_L)$, если он минимизирует сумму $\sum_{i=1}^{L} w_i l_i$.

Если λ — оптимальный вектор для W, то из $l_i \geqslant l_j$ следует, что $w_i \leqslant w_j$. Предположим, что $l_1 \leqslant l_2 \leqslant \ldots \leqslant l_L$ и $w_1 \geqslant w_2 \geqslant \ldots \geqslant w_L$. Случай $m=1$ тривиален, поэтому мы предполагаем в дальнейшем, что $m \geqslant 2$. Следующий результат является основой алгоритма Хаффмана построения оптимального вектора длин путей для заданного вектора весов.

Теорема 15.2. Пусть $\lambda=(l_1, l_2, \ldots, l_L)$ — оптимальный вектор длин путей для вектора весов $W=(w_1, w_2, \ldots, w_L)$. Предположим, что $l_1 \leqslant l_2 \leqslant \ldots \leqslant l_L$ и $w_1 \geqslant w_2 \geqslant \ldots \geqslant w_L$. Тогда $l_{L-d+1}=l_{L-d+2}=\ldots=l_L$, где $d=2$, если $m=2$; в случае $m>2$ величина d определяется следующим образом:

$$d=\begin{cases} m, & \text{если } L \equiv 1 \pmod{m-1}, \\ m-1, & \text{если } L \equiv 0 \pmod{m-1}, \\ \rho, & \text{если } L \equiv \rho \pmod{m-1}. \end{cases} \tag{15.17}$$

Доказательство. По существу, в теореме утверждается, что вектор длин путей содержит по меньшей мере d длин, равных максимальной длине в том случае, если он оптимален. Доказательство теоремы для случая, когда $m=2$, проводится просто. Поэтому мы рассмотрим случай $m>2$.

Из условия (15.12), накладываемого на характеристическую сумму, следует, что

$$m^{l_L} - \sum_{i=1}^{L} m^{l_L - l_i} \geqslant 0. \tag{15.18}$$

Покажем, что

$$m-2 \geqslant m^{l_L} - \sum_{i}^{L} m^{l_L - l_i}. \tag{15.19}$$

Если условие (15.19) неверно, то $m^{l_L} - \sum_{i=1}^{i} m^{l_L - l_i} \geqslant m-1$. Таким образом, $\sum_{i=1}^{L-1} m^{l_L - l_i} + m \leqslant m^{l_L}$. Поделив это выражение на m^{l_L}, получим $\sum_{i=1}^{L-1} m^{-l_i} + m^{-(l_L-1)} \leqslant 1$. Однако это противоречит тому, что λ — оптимальный вектор. Поэтому соотношение (15.19) верно.

Из того, что $m \equiv 1 \pmod{m-1}$, следует, что $m^k \equiv 1 \pmod{m-1}$ для любого целого неотрицательного числа k. Поэтому

$$m^{l_L} - \sum_{i=1}^{L} m^{l_L - l_i} \equiv 1 - L \pmod{m-1} \equiv \equiv 1 - \rho \pmod{m-1}, \tag{15.20}$$

где $L \equiv \rho \pmod{m-1}$. Тогда

$$1 - \rho \pmod{m-1} = \begin{cases} 0, & \text{если } \rho = 1, \\ 1, & \text{если } \rho = 0, \\ m - \rho, & \text{если } 2 \leqslant \rho \leqslant m-2. \end{cases} \tag{15.21}$$

Из выражений (15.18) и (15.19) следует, что сумма $m^{l_L} - \sum_{i=1}^{L} m^{l_L - l_i}$ неотрицательна и меньше $m-1$.

Таким образом, из выражений (15.20) и (15.21) следует

$$m^{l_L} - \sum_{i=1}^{L} m^{l_L - l_i} = \begin{cases} 0, & \text{если } \rho = 1, \\ 1, & \text{если } \rho = 0, \\ m - \rho, & \text{если } 2 \leqslant \rho \leqslant m-2. \end{cases} \tag{15.22}$$

Предположим, что j — последний индекс, для которого верно, что $l_j < l_L$. Тогда верно, что $l_1 \leqslant l_2 \leqslant \ldots \leqslant l_j < l_{j+1} = l_{j+2} = \ldots = l_L$.

Чтобы показать, что $L - j \geqslant d$, используем (15.22). Тем самым теорема будет доказана. Мы можем переписать (15.22) следующим образом:

$$m^{l_L} - \sum_{i=1}^{j} m^{l_L - l_i} = \begin{cases} L - j, & \text{если } \rho = 1, \\ L - j + 1, & \text{если } \rho = 0, \\ L - j + m - \rho, & \text{если } 2 \leqslant \rho \leqslant m-2. \end{cases}$$

Левая часть вышеприведенного равенства делится на m, и мы получаем следующее:

Если $\rho = 1$, тогда $L - j = km$ для некоторого положительного целого числа k.

Если $\rho = 0$, тогда $L - j + 1 = km$ для некоторого положительного целого числа k.

Если $2 \leqslant \rho \leqslant m-2$, тогда $L - j + m - \rho = km$ для некоторого положительного целого числа k.

Таким образом, из выражения (15.17) следует, что если $\rho = 1$, то $L - j \geqslant m = d$, если $\rho = 0$, то $L - j \geqslant m - 1 = d$, если $2 \leqslant \rho \leqslant m-2$, то $L - j \geqslant \rho = d$.

Опишем алгоритм Хаффмана для построения оптимального вектора длин путей по заданному вектору весов W_0. Обозначим число весов в векторе весов W_i через L_i. Если $L_0 \leqslant m$, то соответствующим оптимальным вектором длин путей является вектор, состоящий из единиц. Поэтому мы предполагаем, что $L_0 > m$.

Начиная с W_0 алгоритм Хаффмана порождает такую последовательность $W_1, W_2, \ldots, W_s$ векторов весов, что $L_0 > L_1 > \ldots > L_s = m$. Для каждого L_i мы определим d_i в соответствии с выражением (15.17), т. е. $d_i = 0$, если $m = 0$.

Иначе

$$d_i = \begin{cases} m, & \text{если } L_i \equiv 1 \pmod{m-1}, \\ m-1, & \text{если } L_i \equiv 0 \pmod{m-1}, \\ \rho, & \text{если } L_i \equiv \rho \pmod{m-1} \text{ и } 2 \leqslant \rho \leqslant m-2. \end{cases} \tag{15.23}$$

Мы используем d_i при построении W_{i+1} из W_i. Алгоритм затем начинает с вектора $\lambda_s=(\overbrace{1, 1, \ldots, 1}^{m})$ и строит такую последовательность векторов длин путей $\lambda_s, \lambda_{s-1}, \ldots, \lambda_0$, что λ_i — оптимальный вектор длин путей для W_i. Мы предполагаем, что длины в каждом векторе λ_i расположены в том же порядке, что и веса в соответствующем векторе W_i.

Алгоритм 15.3. Оптимальный вектор длин путей (Хаффман).

S1. W_0— данный вектор весов. Пусть $i=0$.

S2. Пусть $W_i=(w_1, w_2, \ldots, w_{L_i}$, где $w_1 \geqslant w_2 \geqslant \ldots \geqslant w_{L_i}$. Построить W_{i+1} следующим образом:
1) Вычислить d_i в соответствии с выражением (15.23).
2) Вычислить $P_i=w_{L_i-d_i+1}+w_{L_i-d_i+2}+\ldots+w_{L_i}$, т. е. P_i— сумма последних d_i весов W_i.
3) W_{i+1} состоит из весов $w_1, w_2, \ldots, w_{L_i-d_i}$ и P_i, расположенных в невозрастающем порядке.

S3. Пусть $i=i+1$. Если $L_i=m$, то идти к шагу S4. Иначе идти к шагу S2.

S4. Пусть $\lambda_i=(\overbrace{1, 1, \ldots, 1}^{m})$. (Отметим, что теперь $i=s$). Положить $i=i-1$.

S5. Построить λ_i из λ_{i+1} следующим образом:
1) Каждому весу из первых L_i-d_i весов в векторе W_i приписывается длина, равная его длине в векторе W_{i+1}.
2) Каждому весу из последних d_i весов в векторе w_i приписывается длина, которая больше, чем длина для веса P_i в векторе W_i, на единицу.

S6. Если $i=0$, то HALT; λ_0— оптимальный вектор длин путей для вектора весов W_0. Иначе положить $i=i-1$ и идти к шагу S5.

Ясно, что λ_i — вектор длин путей. Из выражения (15.23) следует, что $d_i \leqslant m$ для всех i. Построение в алгоритме гарантирует, что λ_i — вектор длин путей для всех i.

Работа алгоритма Хаффмана проиллюстрирована на рис. 15.5 для $W_0=(9, 7, 6, 5, 5, 3, 3, 2, 1)$ и $m=3$. Деревья T_3, T_2, T_1, T_0, соответствующие векторам длин путей λ_3, λ_2, λ_1, λ_0, показаны на рис. 15.6. (Отметим, что T_i строится из T_{i+1} добавлением d_i ребер к

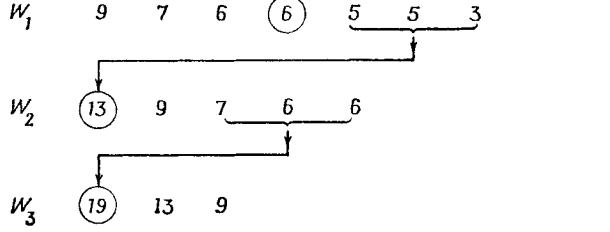


Рис. 15.5.
Иллюстрация алгоритма Хаффмана.

листу в T_{i+1}, представляющему вес P_i.) Покажем корректность алгоритма Хаффмана.

Теорема 15.3. Пусть $W_0, W_1, \ldots, W_s$ и $\lambda_0, \lambda_1, \ldots, \lambda_s$ — векторы весов и длин путей, построенные алгоритмом Хаффмана. Для любого i, $0 \leqslant i \leqslant s$, λ_i — оптимальный вектор длин путей для вектора весов W_i.

Доказательство. Ясно, что вектор длин путей $\lambda_s = (\underbrace{1, 1, \ldots, 1}_{m})$ оптимален для W_S. Покажем, что если λ_{i+1} оптимален для W_{i+1}, то вектор λ_i оптимален для W_i. Допустим противное, а именно что λ_{i+1} оптимален для W_{i+1}, а λ_i не оптимален для W_i. Тогда пусть λ_i^* — оптимальный вектор длин для вектора весов W_i. Пусть $W_i = (p_i, p_2, \ldots, p_{L_i})$, причем $p_1 \geqslant p_2 \geqslant \ldots > p_{L_i}$, $\lambda_i = (l_1, l_2, \ldots, l_{L_i})$, $W_{i+1} = (q_1, q_2, \ldots, q_{L_{i+1}})$, причем $q_1 \geqslant q_2 \geqslant \ldots \geqslant q_{L_{i+1}}$, $\lambda_{i+1} = (t_1, t_2, \ldots, t_{L_{i+1}})$, $\lambda_i^* = (l_1^*, l_2^*, \ldots, l_{L_i}^*)$. Так как λ_i не является оптимальным для W_i, то получим

$$\sum_{k=1}^{L_i} p_k l_k > \sum_{k=1}^{L_i} p_k l_k^*. \qquad (15.24)$$

Из шага S5 в построении Хаффмана следует, что для некоторого u $(1 \leqslant u \leqslant L_{i+1})$ следует $q_u = p_{L_i - d_i + 1} + p_{L_i - d_i + 2} + \ldots + p_{L_i}$. (Отметим, что q_u совпадает с P_i, а остальные q_i есть $p_1, p_2, \ldots, p_{L_i - d_i}$). Верно также, что $t_u = l_{L_i} - 1$. Таким образом,

$$\sum_{k=1}^{L_{i+1}} q_k t_k = \sum_{k=1}^{L_i} (p_k l_k) - q_u. \qquad (15.25)$$

Так как λ_i^* — оптимальный вектор, то мы получаем из теоремы 15.2, что $l^*_{L_i - d_i + 1} = l^*_{L_i - d_i + 2} = \ldots = l^*_{L_i}$. Если мы используем тот же самый тип преобразования, как и в случае перехода от λ_i к λ_{i+1}, то можем получить из λ_i^* вектор длин путей $\lambda_{i+1}^* = (t_1^*, t_2^*, \ldots, t^*_{L_{i+1}})$, который

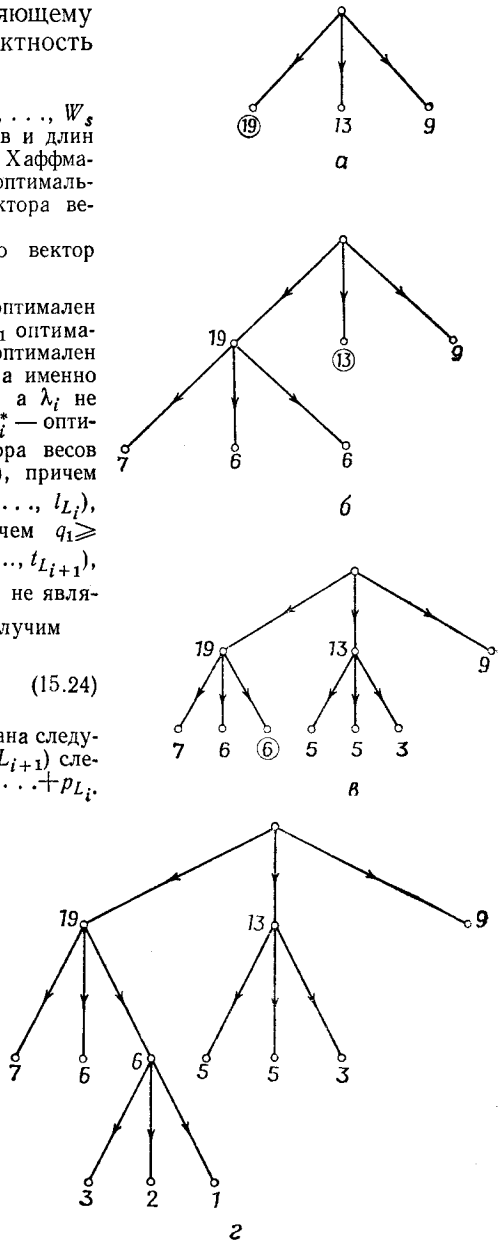


Рис. 15.6.
а — дерево T_3; *б* — дерево T_2; *в* — дерево T_1; *г* — дерево T_0.

удовлетворяет условию

$$\sum_{k=1}^{L_{i+1}} q_k t_k^* = \sum_{k=1}^{L_i} (p_k l_k^*) - q_u. \tag{15.26}$$

Теперь мы можем показать, используя выражение (15.24) — (15.26), что $\sum_{k=1}^{L_{i+1}} q_k t_k^* < \sum_{k=1}^{L_{i+1}} q_k t_k$, откуда следует, что λ_{i+1} не является оптимальным вектором длин путей. Пришли к противоречию.

15.3. Оптимальные деревья бинарного поиска

Упорядоченным деревом называется ориентированное дерево, для которого определен порядок сыновей любой вершины дерева. Обычно мы изображаем упорядоченное дерево так, что корень размещается вверху, а сыновья каждой вершины расположены в соответствии с порядком слева направо. Для бинарного упорядоченного дерева поддерево, корень которого является левым сыном вершины v, называется *левым поддеревом* вершины v. Аналогично определяется *правое поддерево* вершины v. Пусть $A=\{a_1, a_2, \ldots, a_n\}$ — множество элементов, упорядоченных следующим образом: $a_1 < a_2 < \ldots < a_n$. Бинарным деревом поиска для множества A является упорядоченное бинарное дерево, в котором каждая вершина v так помечена элементом $l(v)$ множества A, что 1) для каждого u в левом поддереве вершины v, $l(u)<l(v)$; 2) для каждого u в правом поддереве вершины v, $l(u)>l(v)$; 3) для каждого элемента x множества A существует точно одна такая вершина v, что $l(v)=x$. Предположим, A является подмножеством универсума S. Тогда пусть $B=\{b_0, b_1, \ldots, b_n\}$ — такое множество, что 1) b_i ($1 \leqslant i \leqslant n-1$) представляет множество всех элементов $x \in S - A$, для которых $a_i < x < a_{i+1}$; 2) b_0 представляет собой множество всех элементов $x \in S-A$, для которых $x < a_1$

3) b_n представляет множество всех элементов $x \in S-A$, для которых $x > a_n$.

Расширенным деревом бинарного поиска для A называется дерево бинарного поиска для A с $n+1$ листьями, представляющими элементы множества B. Отметим, что в расширенном дереве бинарного поиска полустепень исхода каждой внутренней вершины равна двум.

Например, деревья, представленные на рис. 15.7, являются двумя различными расширенными деревьями бинарного поиска над множеством $A=\{a_1, a_2, a_3, a_4\}$. Отметим, что в расширенном дереве бинарного поиска листья появляются слева направо в порядке $b_0, b_1, \ldots, b_n$. В дальнейшем мы будем называть расширенное дерево бинарного поиска для множества A просто *деревом бинарного поиска для* A.

Пусть дано подмножество A универсума S и дерево T бинарного поиска для A. Множество S может быть множеством всех слов над английским алфавитом, а упорядочение может быть лексикографическим порядком. Предположим, что нам необходимо определить, принадлежит ли элемент $x \in S$ множеству A. Сравним x с элементом,

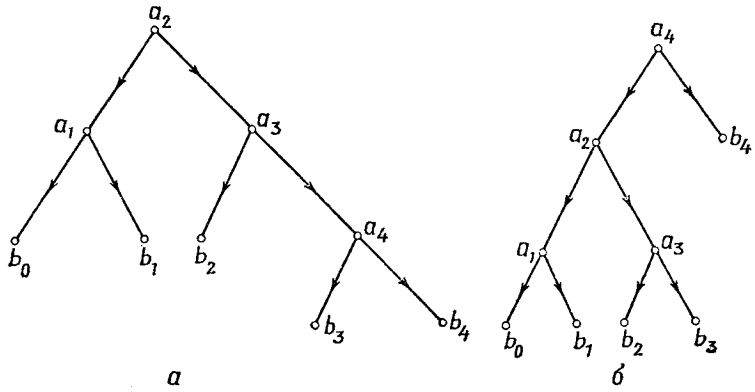


Рис. 15.7. Примеры деревьев бинарного поиска.

который соответствует корню дерева T. При этом могут возникнуть четыре случая, в соответствии с которыми мы продолжаем:

Случай 1. Корень отсутствует (дерево T бинарного поиска пусто). Следовательно, x не принадлежит A, и поиск завершается безуспешно.

Случай 2. x равен элементу, соответствующему корню. Следовательно, поиск завершается успешно.

Случай 3. x меньше элемента, соответствующего корню. Поиск продолжается ниже в левом поддереве корня.

Случай 4. x больше элемента, соответствующего корню. Поиск продолжается ниже в правом поддереве корня. Ясно, что успешный поиск завершается во внутренних вершинах, а безуспешный — в листьях дерева T.

Определим глубину вершины v как длину пути из корня до v. Тогда можно отметить, что число сравнений, выполняемых до того, как поиск завершается успешно во внутренней вершине v дерева T, на единицу больше глубины v. С другой стороны, если поиск завершается безуспешно в листе, то число выполненных сравнений равно глубине листа.

Пусть $p_1, p_2, \ldots, p_n$ — частоты обращения к элементам $a_1, a_2, \ldots, a_n$ соответственно, и пусть $q_0, q_1, \ldots, q_n$ — частоты, с которыми поиск завершается в листьях T, представляющих $b_0, b_i, \ldots, b_n$ соответственно. Тогда среднее время поиска в дереве T (по всем возможным ситуациям) пропорционально стоимости дерева T, определяемой в виде $\sum_{i=1}^{n} p_i$ (глубина $a_i + 1) + \sum_{i=0}^{n} q_i$ (глубина b_i).

Например, если $p_1=0{,}2$, $p_2=0{,}2$, $p_3=0{,}1$, $p_4=0{,}1$, $q_0=0{,}1$, $q_1=0{,}1$, $q_2=0{,}05$, $q_3=0{,}05$; $q_4=0{,}1$, то стоимость дерева, изображенного на рис. 15.7, *а*, составит $\{p_1(2)+p_2(1)+p_3(2)+p_4(3)\}+\{q_0(2)+q_1(2)+q_2(2)+q_3(3)+q_4(3)\}=2{,}05$. Стоимость дерева, изображенного на рис. 15.7, *б*, составит $\{p_1(3)+p_2(2)+p_3(3)+p_4(1)\}+\{q_0(3)+q_1(3)+q_2(3)+q_3(3)+q_4(1)\}=2{,}4$.

Этот пример дает представление о следующей проблеме. Даны неотрицательные веса p_i и q_i. Необходимо построить дерево бинарного поиска минимальной стоимости для множества $A=\{a_1, a_2, \ldots, a_n\}$, элементы которого упорядочены следующим образом: $a_1<a_2<\ldots<a_n$. Нам часто будет удобно ссылаться на такое дерево как на *оптимальное* дерево бинарного поиска для весов q_0, p_1, q_1, p_2, $\ldots$, q_{n-1}, p_n, q_n. Алгоритм построения оптимального дерева бинарного поиска, который мы сейчас рассмотрим, основан на следующем свойстве таких деревьев:

Пусть T — оптимальное дерево бинарного поиска для весов q_0, p_1, q_1, p_2, $\ldots$, p_n, q_n. Если a_k — элемент, соответствующий корню дерева T, то левое поддерево корня дерева T является оптимальным деревом бинарного поиска для весов q_0, p_1, q_1, p_2, $\ldots$, p_{k-1}, q_{k-1}, а правое поддерево корня дерева T — оптимальным деревом бинарного поиска для весов q_0, p_1, $\ldots$, p_{k-1}, q_{k-1}. Пусть T_{ij} для $0\leqslant i<j\leqslant n$ является оптимальным деревом бинарного поиска для множества $\{q_i, p_{i+1}, q_{i+1}, \ldots, p_j, q_j\}$. Пусть r_{ij} — корень дерева T_{ij}, а c_{ij} — стоимость дерева T_{ij}. Вес w_{ij} дерева T_{ij} определяется как $q_i+(p_{i+1}+q_{i+1})+\ldots+(p_j+q_j)$. T_{ii} будет деревом, состоящим из листа b_i. Поэтому $c_{ii}=0$, а $w_{ii}=q_i$.

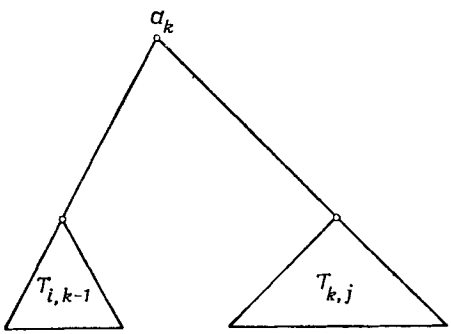


Рис. 15.8. Поддерево T_{ij} с корнем a_k.

Предположим, что a_k — корень дерева $T_{ij}(i<j)$. Тогда, как мы уже видели, левое поддерево с корнем a_k является деревом $T_{i,k-1}$, которое в свою очередь является оптимальным деревом бинарного поиска для множества $\{q_i, p_{i+1}, \ldots, p_{k-1}, q_{k-1}\}$, а правое поддерево с корнем a_k является деревом T_{kj}, которое в свою очередь является оптимальным деревом бинарного поиска для множества $\{q_k, p_{k+1}, \ldots, p_j, q_j\}$ (рис. 15.8). Так как глубина вершины в дереве T_{ij} на единицу больше ее глубины в дереве $T_{i,k-1}$ или T_{kj}, то отсюда следует, что

$$c_{ij}=c_{i,k-1}+w_{i,k-1}+c_{kj}+w_{kj}+p_k=w_{ij}+c_{i,k-1}+c_{kj}. \tag{15.27}$$

Ясно, что мы можем найти корень r_{ij} дерева T_{ij}, определив величину $k\,(i<k\leqslant j)$, которая минимизирует сумму (15.27). Это рассуждение образует основу алгоритма 15.4, который позволяет рассчитать r_{ij} и c_{ij} в порядке возрастания величин $j-i$. Этот алгоритм предложен Гильбертом и Муром [15.19].

Алгоритм 15.4. Оптимальное дерево бинарного поиска (Гильберт и Мур).
S1. Даны неотрицательные веса $p_1, p_2, \ldots, p_n$ и $q_0, q_1, \ldots, q_n$. Для $i=0, 1, 2, \ldots, n$ положить $w_{ii}=q_i$, $c_{ii}=0$, $r_{ii}=b_i$.
S2. Для $l=1, 2, \ldots, n$ выполнить шаг S3.
S3. Для $i=0, 1, 2, \ldots, n-l$ выполнить шаг S4.
S4. Положить $j=i+l$, $w_{ij}=w_{i,\,j-1}+p_l+q_j$.
Пусть m — значение $k\,(i<k\leqslant j)$, для которого сумма $c_{i,\,k-1}+c_{kj}$ минимальна.
Положить $c_{ij}=w_{ij}+c_{i,\,m-1}+c_{mj}$, $r_{ij}=a_m$.
S5. HALT.

Вычислив r_{ij} с помощью приведенного выше алгоритма, мы можем легко построить дерево T_{0n}, используя следующую процедуру:

1. r_{0n} — корень дерева T_{0n}. Если $r_{0n}=a_k$, то a_k имеет в качестве левого сына $r_{0,\,k-1}$, а в качестве правого — r_{kn}.

2. Рассмотрим внутреннюю вершину a_m. Если $a_m=r_{ij}$, то a_m имеет $r_{i,\,m-1}$ в качестве левого сына, а r_{mj} — в качестве правого.

Проиллюстрируем алгоритмы 15.4 и приведенную выше процедуру для построения оптимального дерева бинарного поиска.

Таблица 15.1

$l=j-i$ ↓	$i\rightarrow$ 0	1	2	3	4
0	$w_{00}=1$ $c_{00}=0$ $r_{00}=b_0$	$w_{11}=2$ $c_{11}=0$ $r_{11}=b_1$	$w_{22}=3$ $c_{22}=0$ $r_{22}=b_2$	$w_{33}=2$ $c_{33}=0$ $r_{33}=b_3$	$w_{44}=1$ $c_{44}=0$ $r_{44}=b_4$
1	$w_{01}=5$ $c_{01}=5$ $r_{01}=a_1$	$w_{12}=6$ $c_{12}=6$ $r_{12}=a_2$	$w_{23}=8$ $c_{23}=8$ $r_{23}=a_3$	$w_{34}=4$ $c_{34}=4$ $r_{34}=a_4$	
2	$w_{02}=9$ $c_{02}=14$ $r_{02}=a_2$	$w_{13}=11$ $c_{13}=17$ $r_{13}=a_3$	$w_{24}=10$ $c_{24}=14$ $r_{24}=a_3$		
3	$w_{03}=14$ $c_{03}=27$ $r_{03}=a_2$	$w_{14}=13$ $c_{14}=23$ $r_{14}=a_3$			
4	$w_{04}=16$ $c_{04}=34$ $r_{04}=a_3$				

Рассмотрим четыре элемента $a_1<a_2<a_3<a_4$ с весами $p_1=2$, $p_2=1$, $p_3=3$ и $p_4=1$, а также $q_0=1$, $q_1=2$, $q_2=3$, $q_3=2$ и $q_4=1$. В табл. 15.1 указаны значения w_{ij}, r_{ij} и c_{ij}, вычисленные на основе алгоритма 15.4. В этой таблице элементы в строке вычисляются слева направо. Вычисления в какой-либо строке начинаются только после вычислений всех элементов предыдущей строки. На рис. 15.9 представлено соответствующее оптимальное дерево бинарного поиска.

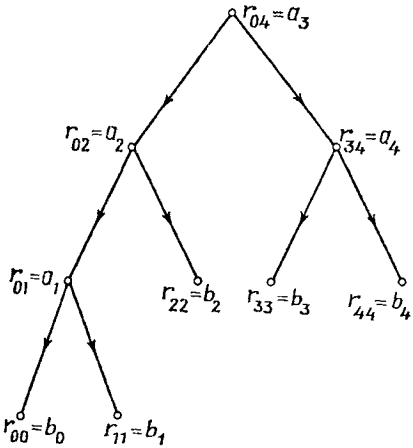


Рис. 15.9. Построение оптимального дерева бинарного поиска.

Легко показать, что алгоритм 15.4 имеет сложность, равную $O(n^3)$, а процедура построения дерева по таблице r_{ij} — сложность $O(n)$. Таким образом, оптимальное дерево бинарного поиска можно построить за время $O(n^3)$.

Кнут [15.20] показал, что корень дерева T_{ij} лежит между корнями $r_{i,\ j-1}$ и $r_{i+1,\ j}$. Другими словами, $r_{i,j-1}\leqslant r_{ij}\leqslant r_{i+1,\ j}$. Поэтому на шаге S_4 поиск величины m можно производить в пределах $r_{i,\ j-1}$,—$r_{i+1,\ j}$. Сложность алгоритма 15.4 с такой модификацией становится равной $O(n^2)$. Несколько обобщений алгоритма Кнута обсуждаются в работе [15.21].

Авторы работы [15.22] предложили $O(n^2)$-алгоритм для построения оптимального дерева бинарного поиска в частном случае, когда $p_i=0$ [15.23]. Кнут [15.24] показал, что этот алгоритм можно выполнить за время $O(n \log n)$. Другой алгоритм, тесно связанный с алгоритмом Ху — Такера со сложностью $O(n \log n)$ для этой же задачи, предложен в работе [15.25]. Этот алгоритм легче проверить: он основан на использовании вариационных методов.

Проведены обширные исследования методов поиска с рассмотрением некоторых вопросов, касающихся оптимальных деревьев бинарного поиска. Детальное обсуждение этих тем и связанная с ними библиография приведены в работах [15.24, 15.26, 15.27].

15.4. Максимальные паросочетания в графе

В этом разделе мы рассмотрим задачу построения максимального паросочетания в графе. Сначала мы обсудим основной подход к построению такого паросочетания, предложенный в работе [15.28].

Затем опишем алгоритм Габова [15.29], который является эффективной реализацией подхода Эдмондса.

Более эффективный алгоритм для случая двудольных графов обсуждается в разд. 15.5. Приложения, связанные с задачей об оптимальном назначении и задачей о расписании, рассматриваются в разд. 15.6.

15.4.1. Подход Эдмондса

Алгоритм Эдмондса основан на теореме Бержа (8.20), в которой устанавливается, что паросочетание максимально тогда и только тогда, когда не существует добавляющего пути по отношению к этому паросочетанию. Поэтому по данному графу и начальному паросочетанию M мы можем получить максимальное паросочетание, действуя следующим образом.

Найдем добавляющий путь P по отношению к M. Получим паросочетание $M \oplus P$, которое имеет на одно ребро больше, чем M. Находим добавляющий путь по отношению к этому новому паросочетанию и т. д. Повторяем эту операцию до тех пор, пока мы не получим паросочетания, по отношению к которому не существует добавляющих путей. Тогда по теореме Бержа такое паросочетание максимально. Таким образом, задача в основном сводится к эффективному нахождению добавляющего пути по отношению к данному паросочетанию. Наиболее важной идеей в данном контексте является понятие «цветок», введенное Эдмондсом, которое описывается ниже.

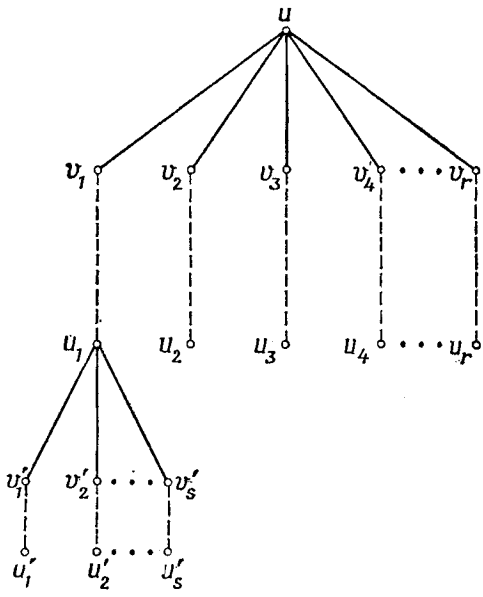


Рис. 15.10.

Чтобы найти добавляющий путь по отношению к паросочетанию M, мы обязательно должны начать наш поиск с ненасыщенной вершины, например u. Если существует добавляющий путь P из u к u' (отметим, что u' является также ненасыщенной вершиной), то в P вершина u' смежна с вершиной u либо с насыщенной вершиной v. Такая вершина v будет находиться на четном расстоянии от u в пути P, т. е. существует чередующийся путь четной длины из

вершины u в вершину v. Отсюда следует, что поиск добавляющего пути должен проводиться только в ограниченной группе вершин, а именно тех вершин, до которых существует чередующийся путь четной длины из u.

Например, пусть $v_1, v_2, \ldots, v_r$ — вершины, смежные с u (рис. 15.10). Если какая-либо из этих вершин не насыщена, то мы нашли добавляющий путь. Иначе, пусть $u_1, u_2, \ldots, u_r$ — соответствующие им напарники в паросочетании M. На этом этапе выбранная группа состоит из вершин $u_1, u_2, \ldots, u_r$. Тогда мы выбираем вершину, например u_1, из выделенной группы, которая еще не рассматривалась. Если u_1 имеет смежную с ней вершину, которая не насыщенна, то мы нашли добавляющий путь.

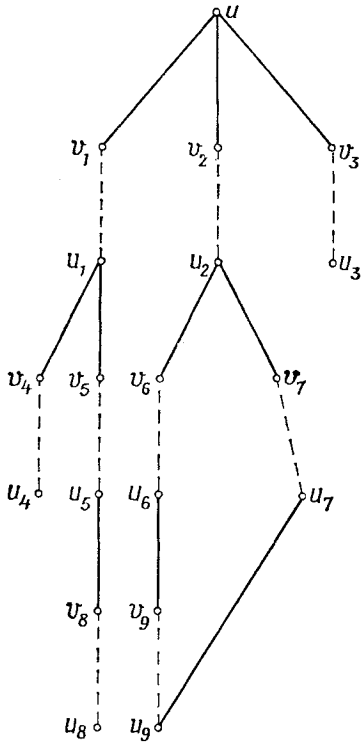


Рис. 15.11. Образование «цветка».

Другими словами, предположим, что u_1 не смежна ни с одной из выбранных вершин. Если $v_1', v_2', \ldots, v_s'$ — такие вершины, смежные с u_1, что $v_i' \neq v_j$ для любых i и j, то их напарники $u_1', u_2', \ldots, u_s'$ также присоединяются к выбранной группе вершин. Если мы при рассмотрении вершины в выбранной группе обнаружим, что она смежна с другой вершиной, уже принадлежащей выбранной группе, то это означает, что получен нечетный цикл, т. е. цикл нечетной длины. Этот цикл, являющийся замкнутым чередующимся путем нечетной длины, называется «цветком». Например, на рис. 15.11 добавление ребра (u_9, u_7) создает «цветок» $\{u_2, v_6, u_6, v_9, u_9, u_7, v_7, u_2\}$. До добавления этого ребра выбранная группа состоит из вершин $u, u_1, u_2, \ldots, u_9$. Если образуется «цветок», то вершины v_6, v_7 и v_9 также присоединяются к выбранной группе, поскольку мы можем найти чередующийся путь четной длины из вершины u к этим вершинам. Например, на рис. 15.11 $(u, v_2, u_2, v_7, u_7, u_9, v_9)$ — чередующийся путь четной длины к вершине v_9. Как только образовался «цветок», мы заключаем, что все вершины этого «цветка» присоединяются к выбранной группе.

Если в алгоритме Эдмондса образуется «цветок», то все его вершины заменяются одной. Новая вершина называется *псевдоверши-*

ной. Все вершины, которые были смежны с одной или более вершинами «цветка», будут смежны с этой новой вершиной. Таким образом, мы получаем сокращенный граф. Эта процедура повторяется всякий раз, когда обнаруживается «цветок». Если в результате мы закончим, получив некоторый сокращенный граф, но не получив добавляющего пути, то это означает, что не существует добавляющего пути по отношению к текущему паросочетанию и в исходном графе. Следовательно, текущее паросочетание максимально. С другой стороны, если мы обнаруживаем добавляющий путь в какомлибо сокращенном графе, то, очевидно, следует, что существует добавляющий путь P и в исходном графе. Чтобы найти этот путь P, мы должны вернуться по пройденному пути, восстанавливая «цветки», полученные ранее.

Сжатие и восстановление «цветков», требуемые в алгоритме Эдмондса, могут привести к сложности выполнения алгоритма $O(n^4)$, где n — число вершин в графе.

Габов устранил операции расширения и сжатия, записывая структуры «цветков» с помощью эффективной процедуры расстановки меток и соответствующих массивов. Это позволило достичь сложности $O(n^3)$. Метод расстановки меток, используемый Габовым, аналогичен методам, используемым в алгоритмах паросочетаний в работах [15.30—15.32].

15.4.2. Метод Габова

Вначале мы обсудим главную стратегию и определим различные массивы, используемые в алгоритме Габова.

Пусть дан граф, имеющий n вершин и m ребер. Алгоритм начинает свою работу с определения нумерации вершин и ребер графа. Вершины нумеруются числами от 1 до n, а ребра — числами $n+2$, $n+4$, . . ., $n+2m$. Номер ребра (x, y) обозначается через $N(x, y)$. В алгоритме используется номинальная вершина, отмеченная числом 0. В массиве END элементы пронумерованы от $n+1$ до $n+2m$. Для каждого ребра массива END имеется два элемента, расположенные последовательно и содержащие номера вершин, инцидентных этому ребру. Таким образом, если ребро (v, w) имеет номер k (где $k=n+2i$ для некоторого $1 \leqslant i \leqslant m$), то END $(k-1)=v$, а END $(k)=w$, т. е. по данному номеру ребра с помощью этого массива можно легко определить вершины, инцидентные этому ребру.

Алгоритм Габова строит множество паросочетаний, начиная с начального, которое может быть пустым. Он завершается получением максимального паросочетания. Паросочетание хранится в массиве MATE. Этот массив имеет по одному элементу для каждой вершины. Ребро (v, w) входит в паросочетание, если MATE $(v)=w$ и MATE $(w)=v$.

Вершина v называется *внешней* по отношению к фиксированной ненасыщенной вершине u и тогда и только тогда, когда существует чередующийся путь четной длины от u к v. Ясно, что этот путь $P(v)$, когда он прослеживается от v к u, начинается с ребра, вошедшего в паросочетание. Таким образом, $P(v)=(v, v_1, \ldots, u$, где (v, v_1) — ребро, вошедшее в паросочетание. Если рассматривается ребро, соединяющее внешнюю вершину v с ненасыщенной вершиной $u' \neq u$, то алгоритм при этом выделяет добавляющий путь вида $(u')*P(v)=(u', v, v_1, \ldots, u)$, где $*$ — обозначение конкатенации. Если такое ребро никогда не будет обнаружено, то вершина u не войдет ни в один добавляющий путь.

В массиве LABEL каждой вершине соответствует элемент массива. При этом элемент массива, соответствующий внешней вершине v, используется для определения чередующегося пути $P(v)$. Элемент массива LABEL интерпретируется как начальная метка, или как метка вершины, или как метка ребра.

Начальная метка. Начальная вершина u имеет начальную метку. В этом случае LABEL (u) устанавливается равным 0. Чередующийся путь $P(u)=(u)$.

Метка вершины. Если LABEL $(v)=i$, где $1 \leqslant i \leqslant n$, то говорят, что v имеет метку вершины. В этом случае v является внешней вершиной и элемент LABEL (v) равен номеру другой внешней вершины. Путь $P(v)$ определяется как $(v$, MATE $(v))*P$ (LABEL $(v))$.

Метка ребра. Если LABEL $(v)=n+2i$, $1 \leqslant i \leqslant m$, то говорят, что v имеет метку ребра. Пусть v — внешняя вершина и LABEL (v) содержит номер ребра, соединяющего две внешние вершины, например x и y. Поэтому LABEL $(v)=N(x, y)$. Метка ребра $N(x, y)$ вершины v указывает, что существует чередующийся путь $P(v)$ четной длины из v в начальную вершину u, который проходит через ребро (x, y). Путь $P(v)$ можно определить через пути $P(x)$ и $P(y)$. Пусть v принадлежит пути $P(x)$, а $P(x, v)$ — часть пути $P(x)$ из x в v. Тогда $P(v)=\mathrm{rev}P(x, v)*P(y)$, где первый член выражения обозначает обращение пути из x в v.

LABEL $(v)<0$, когда v не является внешней вершиной. Вначале, когда все вершины являются невнешними, мы полагаем, что значения LABEL для всех вершин равны -1.

Алгоритм использует также массив FIRST. Если v — внешняя вершина, то FIRST (v) — первая невнешняя вершина в $P(v)$. Если путь $P(v)$ не содержит невнешних вершин, то FIRST (v) полагается равным 0. FIRST $(v)=0$, если v — невнешняя вершина.

Массив, называемый OUTER, используется для хранения внешних вершин, встречающихся при поиске добавляющего пути. Граф поиска растет во внешних вершинах в порядке их появления. В этих внешних вершинах производится поиск в ширину.

Алгоритм Габова (как он представлен ниже) состоит из трех процедур: PROC-EDMONDS, PROC-LABEL и PROC-REMATCH.

PROC-EDMONDS является главной процедурой. Она начинает поиск добавляющего пути из каждой ненасыщенной вершины, просматривает ребра графа, решая приписать метки или расширить паросочетание.

При обнаружении добавляющего пути (шаг E3 в алгоритме 15.5) вызывается PROC-REMATCH. Эта процедура вычисляет новое паросочетание, которое имеет на одно ребро больше текущего паросочетания.

Если при рассмотрении ребра (x, y) образуется «цветок» (шаг E4), то вызывается PROC-LABEL. В этом случае вершины x и y являются внешними. PROC-LABEL выполняет следующие операции:

1. Значением переменной JOIN является первая невнешняя вершина, которая принадлежит как $P(x)$, так и $P(y)$.
2. Все невнешние вершины, предшествующие JOIN в $P(x)$ или $P(y)$, становятся внешними вершинами. Они помечаются меткой ребра $N(x, y)$. Эта метка ребра указывает, что к каждой из этих вершин существует чередующийся путь четной длины из начальной вершины, который проходит через ребро (x, y).
3. После этого JOIN — первая невнешняя вершина как в $P(x)$, так и в $P(y)$. Поэтому элементы массива FIRST, соответствующие вершинам, предшествующим JOIN в $P(x)$ или $P(y)$, полагаются равными JOIN.

Опишем алгоритм Габова. Комментарии и объяснения, соответствующие каждому шагу, приводятся в скобках.

Алгоритм 15.5. Максимальное паросочетание (Габов).
PROC-EDMONDS.

E0. (Инициализация.) Пусть G — данный граф. Пронумеровать вершины графа G числами от 1 до n, а ребра — числами $n+2, n+4, \ldots, n+2m$. Создать номинальную вершину 0. Для $0 \leqslant i \leqslant n$ положить LABEL $(i)=-1$, FIRST $(i)=0$ и MATE $(i)=0$. (Вначале все вершины являются невнешними и ненасыщенными.) Положить $u=0$.

E1. (Нахождение ненасыщенной вершины.) Положить $u=u+1$. Если $u>n$, то HALT и MATE содержит максимальное паросочетание. Иначе если u — насыщенная вершина, то повторить шаг E1. Если u — ненасыщенная вершина, то добавить u в массив OUTER. Положить LABEL $(u)=0$. (Помечаем вершину u начальной меткой и начинаем новый поиск.)

E2. (Выбор ребра.) Выбрать ребро (x, y) (где x — внешняя вершина), которое еще не рассматривалось в x.
Если такого ребра нет, то идти к шагу E7. (Ребра (x, y) можно выбирать в произвольном порядке. Мы применяем поиск в ширину: выбирается внешняя вершина $x=x_1$, а ребра (x_1, y) просматриваются при последующих выполнениях шага E2. Когда все такие ребра уже выбраны, то выбирается вершина x_2, которая была помечена непосредственно после x_1, и процесс повторяется для случая $x=x_2$. Такой поиск в ширину требует поддержания списка внешних вершин $x_1, x_2, \ldots$. Это осуществляется с помощью массива OUTER.)

E3. (Обнаружить добавляющий путь.) Если y не входит в паросочетание и $y \neq u$, то выполнить PROC-REMATCH и идти к шагу E7.

E4. (Образовать «цветок».) Если y — внешняя вершина, то выполнить PROC-LABEL и идти к шагу E2.

E5. (Пометить вершину.) Положить v=MATE (y). Если v — внешняя вершина, то идти к шагу E6. Если v — невнешняя вершина, то положить LABEL (v)=x, FIRST (v)=y и добавить вершину v к массиву OUTER. (Вершина y встречается в этом поиске впервые; ее напарник v является новой внешней вершиной. Этот факт отмечается в массиве OUTER.) Идти к шагу E6.

E6. (Выбрать новое ребро.) Идти к шагу E2. (Получается замкнутый добавляющий путь четной длины, и ребро (x, y) ничего не добавляет.)

E7. (Прекратить поиск.) Положить LABEL (i)=—1 для $0 \leqslant i \leqslant n$. Затем идти к шагу E1. (Все вершины полагаются не внешними для следующего шага.)

PROC-LABEL.

L0. (Инициализация. Положить r=FIRST (x) и s=FIRST (y). Если r=s, то идти к шагу L6. (Нет ни одной невнешней вершины в «цветке».) Иначе пометить вершины r и s. (Шаги L1 и L2 определяют JOIN при попеременном движении вдоль путей $P(x)$ и $P(y)$. Метки приписываются невнешним вершинам этих путей. Это выполняется посредством помещения в LABEL (r) отрицательного номера ребра, т. е. LABEL (r)=—N (x, y). Таким образом, при каждом вызове процедуры PROC-LABEL используются различные значения меток.)

L1. (Переключение путей.) Если $s \neq 0$, то поменять местами r и s (r — не внешняя помеченная вершина в $P(x)$ и $P(y)$ попеременно.)

L2. (Взять не внешнюю вершину.) Положить r=FIRST (LABEL (MATE (r))). (r полагается равной следующей не внешней вершине в $P(x)$ или $P(y)$. Если r не помечена, то пометить r и идти к шагу L1. Иначе положить JOIN=r. (Мы определили JOIN.) Идти к шагу L3.

L3. (Пометить вершины в $P(x)$, $P(y)$, т. е. всем невнешним вершинам между x и JOIN или y и JOIN будут приписаны метки ребра, а именно $N(x, y)$.) Положить v=FIRST (x) и выполнить шаг L4. Затем положить v=FIRST (y) и выполнить шаг L4. Затем идти к шагу L5.

L4. (Пометить не внешнюю вершину v.) Если $v \neq$JOIN, то положить LABEL (v)=$N(x, y)$, FIRST (v)=JOIN и добавить v в массив OUTER. Затем положить v=FIRST (LABEL (MATE (v))). (Получаем следующую не внешнюю вершину.) Повторить шаг L4. Иначе (т. е. v=JOIN, и, следовательно, мы должны приписать метки ребер всем невнешним вершинам в рассматриваемом пути) продолжить действия, определенные в шаге L3 (т. е. возвратиться к шагу L3.)

L5. (Изменить FIRST.) Для каждой внешней вершины i, если FIRST (i) — внешняя вершина, положить FIRST (i)=JOIN (т. е. JOIN — новая первая не внешняя вершина в $P(i)$).

L6. (Расстановка реберных меток завершается.) Конец процедуры.

PROC-REMATCH.

R0. (Получить добавляющий путь.) Вычислить $P(x)$, как описано ниже:
1. Если x имеет реберную метку $N(v, w)$, то вычислить $P(v)$ и $P(w)$. Если x принадлежит $P(v)$, то $P(x)$=(rev $P(v, x)$)*$P(w)$.
Иначе $P(x)$=(rev $P(w, x)$)*$P(v)$.
2. Если x имеет вершинную метку, то $P(x)$=(x, MATE (x))*P(LABEL (x)). Добавляющий путь p_a определяется выражением P_a=(y)*$P(x)$.

R1. (Дополнить текущее паросочетание.) Получить новое паросочетание, удаляя из текущего все ребра, входящие в P_a, и добавляя к нему все ребра, входящие в P_a и не входившие в исходное паросочетание (т. е. если M — текущее паросочетание, то $M \oplus P_a$ — новое паросочетание). Изменить соответствующим образом элементы массива MATE и завершить процедуру.

Следует отметить, что в вышеприведенном алгоритме поиск добавляющего пути из какой-либо вершины выполняется только однажды. Предположим, что поиск из ненасыщенной вершины u завершается без выделения добавляющего пути. Обозначим этот поиск через S_u. *Венгерским подграфом* H для вершины u называется подграф, который состоит из всех ребер, инцидентных внешним вершинам из

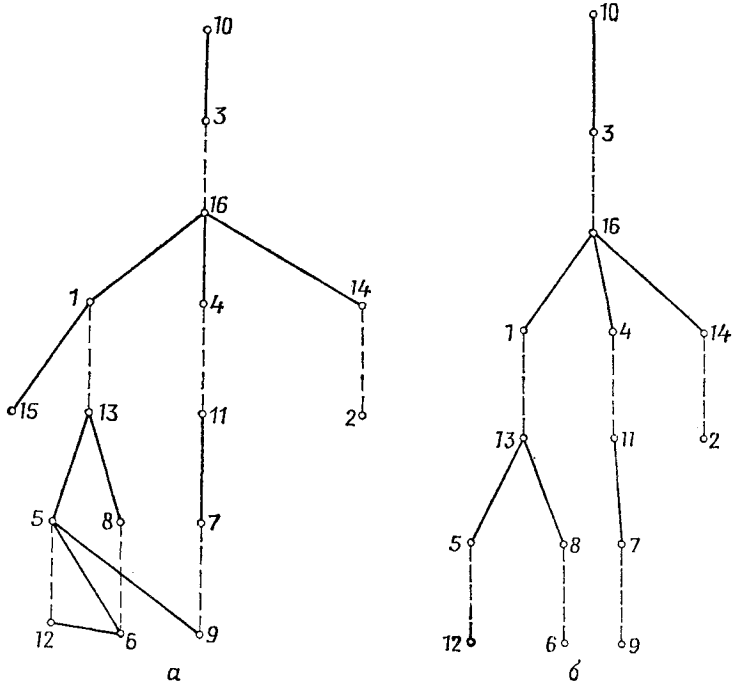


Рис. 15.12.

S_u, и всех вершин, инцидентных этим ребрам. Эдмондс [15.28] показал, что мы можем игнорировать венгерский подграф H после выделения S_u и модифицировать алгоритм 15.5, изменяя шаг E2 следующим образом:

E2′. (Выбрать ребро.) Выбрать ребро... . Если такое ребро не существует, то идти к шагу E1.

Шаг E2′ вызовет выполнение шага E7, который удаляет метки у вершин, пропускаемых после обнаружения S_u. Эта модификация ускоряет алгоритм, если граф не имеет совершенного паросочетания. Однако это не изменяет сложность алгоритма в худшем случае $O(n^3)$.

Вопросы, касающиеся сложности и корректности алгоритма 15.5

обсуждаются в работе [15.29]. Проиллюстрируем работу алгоритма Габова.

Для графа, представленного на рис. 15.12, *а*, начальное паросочетание в котором показано пунктирными линиями, алгоритм Габова работает следующим образом.

Начальной вершиной является вершина 10. После завершения поиска во внешних вершинах 10, 16, 13, 11 и 2 граф поиска будет таким, как показано на рис. 15.12, *б*. Элементы массивов LABEL и FIRST на этом этапе приведены в табл. 15.2. Массив OUTER содержит вершины 10, 16, 13, 11, 2, 12, 6 и 9 в указанном порядке.

Таблица 15.2

Номер вершины	LABEL	FIRST
1	−1	0
2	16	14
3	−1	0
4	−1	0
5	−1	0
6	13	8
7	−1	0
8	−1	0
9	11	7
10	0	0
11	16	4
12	13	5
13	16	1
14	−1	0
15	−1	0
16	10	3

Таблица 15.3

Номер вершины	LABEL	FIRST
1	$N(9,5)$	3
2	16	14
3	−1	0
4	$N(9,5)$	3
5	$N(12,6)$	3
6	13	3
7	$N(9,5)$	3
8	$N(12,6)$	3
9	11	3
10	0	0
11	16	3
12	13	3
13	16	3
14	−1	0
15	−1	0
16	10	3

Когда мы осуществляем поиск в вершине 12, то рассматриваем ребро (12.6) и получаем «цветок» (13, 5, 12, 6, 8, 13). Элементы массивов LABEL и FIRST изменяются следующим образом: LABEL (5)=LABEL (8)=N (12, 6); FIRST (i)=1, i=5, 12, 6, 8. Вершины 5 и 8 помещаются в массив OUTER. Затем, когда мы осуществляем поиск в вершине 6, мы рассматриваем ребро (6, 5), которое образует другой «цветок» (6, 5, 12, 6). Но все вершины этого «цветка» внешние, и поэтому никаких изменений не производится.

Затем мы осуществляем поиск в вершине 9. При рассмотрении ребра (9, 5) образуется «цветок» (9, 7, 11, 4, 16, 1, 13, 8, 6, 12, 5, 9). Вновь изменяем значения элементов массивов LABEL и FIRST для некоторых вершин. Полученные значения приведены в табл. 15.3.

В этот момент массив OUTER содержит вершины 10, 16, 13, 11, 2, 12, 6, 9, 5, 8, 1, 7 и 4 в указанном порядке.

Во всех вершинах массива OUTER вплоть до 9 поиск уже был произведен. Поэтому он продолжается в оставшихся вершинах. Поиск в вершинах 5 и 8 не добавляет каких-либо новых вершин в массив OUTER. При поиске в вершине 1 мы рассматриваем ребро (1, 15) и обнаруживаем, что вершина 15 не насыщена. Получен добавляющий путь. Этим добавляющим путем является путь (15)**P* (1).

Используем процедуру, составляющую шаг R0 для вычисления *P* (1). Вершина 1 имеет метку ребра *N* (9, 5). Поэтому для вычисления *P* (1) нам необходимо иметь пути *P* (9) и *P* (5). Далее, так как вершина 5 имеет метку ребра *N* (12, 6), то нам необходимо иметь пути *P* (12) и *P* (6) для вычисления пути *P* (5). Вершина 12 имеет метку вершины. Поэтому

$$P(12) = (12, \text{MATE}(12)) * P(\text{LABEL}(12)) = (12, 5) * P(13) =$$
$$= (12, 5) * (13, \text{MATE}(13)) * P(\text{LABEL}(13)) = (12, 5, 13, 1) * P(16) =$$
$$= (12, 5, 13, 1) * (16, \text{MATE}(16)) * P(\text{LABEL}(16)) =$$
$$= (12, 5, 13, 1, 16, 3, 10).$$

Аналогично *P* (6)=(6, 8, 13, 1, 16, 3, 10) и *P* (9)=(9, 7, 11, 4, 16, 3, 10).

Так как вершина 5 лежит на пути *P* (12), то *P* (5)=(rev *P* (12, 5))* **P* (6)=(5, 12)*(6, 8, 13, 1, 16, 3, 10)=(5, 12, 6, 8, 13, 1, 16, 3, 10). Мы обнаруживаем, что вершина 1 принадлежит пути *P* (5). Поэтому *P* (1)=(rev *P* (5, 1)**P* (9)=(1, 13, 8, 6, 12, 5)*(9, 7, 11, 4, 16, 3, 10)=(1, 13, 8, 6, 12, 5, 9, 7, 11, 4, 16, 3, 10).

Таким образом, добавляющий путь имеет вид (15)**P* (1)=(15, 1, 13, 8, 6, 12, 5, 9, 7, 11, 4, 16, 3, 10).

После дополнения мы получаем новое паросочетание, состоящее из ребер (15, 1), (13, 8), (6, 12), (5, 9), (7, 11), (4, 16), (3, 10) и (14,2). Так как все вершины графа насыщены в этом паросочетании, то это — максимальное паросочетание. (Фактически это совершенное паросочетание.)

Эдмондс [15.33] и Габов [15.34] рассмотрели эффективный алгоритм для задачи выделения взвешенного паросочетания. Лоулер [15.9] обсуждает детально паросочетание и связанные с ним задачи.

15.5. Максимальные паросочетания в двудольном графе

Задача нахождения максимального паросочетания в двудольном графе имеет широкий спектр приложений. Например, она появляется как подзадача при решении транспортной задачи Хичкока [15.1]. Далее, составление расписаний в определенных случаях включает разбиение множества ребер двудольного графа на непересекающие-

ся паросочетания этого графа. Определение элемента разбиения в свою очередь требует определения максимального паросочетания в двудольном графе. Это показано, например, в работе [15.35].

Ввиду такого разнообразия приложений интересна вычислительная сложность этой проблемы. Хопкрофт и Карп [15.36] показали, как построить максимальное паросочетание в двудольном графе за число шагов, пропорциональное $n^{5/2}$, где n — число вершин в графе. Методология их подхода основывается на некоторых интересных положениях (сформулированных ими) в теории паросочетаний. Эти положения и их алгоритм двудольного паросочетания обсуждаются в этом разделе.

15.5.1. Методология подхода Хопкрофта и Карпа

Все разработанные до сих пор алгоритмы максимального паросочетания начинают с паросочетания (которое может быть не максимальным) и получают, если оно существует, паросочетание большей мощности с помощью выделения добавляющего пути. Выбор добавляющего пути можно выполнить произвольным образом. Сложность этих алгоритмов составляет $O(n^3)$. Хопкрофт и Карп показали, что если дополнение осуществляется вдоль кратчайшего пути, то максимального паросочетания можно достичь за $O(n^{5/2})$ фаз, где каждая фаза включает в себя поиск максимального множества не пересекающихся по вершинам кратчайших, добавляющих по отношению к паросочетанию путей. Докажем этот результат.

Пусть M — паросочетание. Добавляющий путь P называется *кратчайшим путем по отношению к* M, если P имеет наименьшую длину среди всех добавляющих по отношению к M путей.

Лемма 15.3. Пусть M и N — два паросочетания в графе G. Если $|M|=s$ и $|N|=r$, где $r>s$, то $M \oplus N$ содержит по меньшей мере $r-s$ не пересекающихся по вершинам добавляющих по отношению к M путей.

Доказательство. Рассмотрим порожденный подграф G' графа G на множестве ребер $M \oplus N$. Согласно теореме 8.19, каждая (связная) компонента подграфа G' является 1) циклом четной длины, ребра которого попеременно входят в $M-N$ и $N-M$, либо 2) путем, ребра которого попеременно входят в $M-N$ и $N-M$.

Пусть компонентами G' будут $C_1, C_2, \ldots, C_k$, где C_i имеет множества вершин V_i и ребер E_i.

Пусть $\delta(C_i)=|E_i \cap N|-|E_i \cap M|$. Тогда $\delta(C_i)$ есть -1, либо 0, либо 1 для любого i, и $\delta(C_i)=1$ тогда и только тогда, когда C_i—добавляющий по отношению к M путь. Теперь $\sum_{i=1}^{k}\delta(C_i)=|N-M|-|M-N|=|N|-|M|=r-s$. Следовательно, существует не менее $r-s$ таких компонент подграфа G', что $\delta(C_i)=1$. Эти компоненты не пересекаются по вершинам, и каждая из них является добавляющим по отношению к M путем.

Лемма 15.4. Пусть M — паросочетание. Пусть $|M|=r$, и предположим, что мощность максимального паросочетания есть s. Тогда существует добавляющий по отношению к M путь длины не более

$$2\left\lfloor \frac{r}{s-r} \right\rfloor + 1.$$

Доказательство. Пусть s — максимальное паросочетание. Тогда по предыдущей лемме $M \oplus S$ содержит не менее $s-r$ не пересекающихся по вершинам (а следовательно, и ребрам) добавляющих по отношению к M путей. Вместе с тем эти пути содержат не более r ребер из M. Поэтому эти пути будут содержать не более $\lfloor r/(s-r) \rfloor$ ребер из M каждый и, следовательно, всего не более $2\lfloor r/(s-r) \rfloor + 1$ ребер.

Лемма 15.5. Пусть M — паросочетание, P — кратчайший добавляющий по отношению к M путь, а P' — добавляющий по отношению к $M \oplus P$ путь. Тогда $|P'| \geqslant |P| + |P \cap P'|$.

Доказательство. Пусть $N = M \oplus P \oplus P'$. Тогда N — паросочетание и $|N| = |M| + 2$. Поэтому $M \oplus N$ содержит два не пересекающихся по вершинам добавляющих пути P_1 и P_2 по отношению к M.

Поскольку $M \oplus N = P \oplus P'$, то $|P \oplus P'| \geqslant |P_1| + |P_2|$. Но $|P_1| \geqslant |P|$ и $|P_2| \geqslant |P|$, так как P — кратчайший добавляющий путь. Поэтому $|P \oplus P'| \geqslant |P_1| + |P_2| \geqslant 2|P|$. Тогда из тождества $|P \oplus P'| = |P| + |P'| - |P \cap P'|$ мы получаем $|P'| \geqslant |P| + |P \cap P'|$. Предположим, что мы вычислили, начиная с паросочетания $M_0 = \varnothing$, последовательность паросочетаний $M_1, M_2, \ldots, M_i, \ldots$, где, $M_{i+1} = M_i \oplus P_i$, а P_i — кратчайший добавляющий по отношению к M_i путь. Тогда из леммы 15.5 $|P_{i+1}| \geqslant |P_i| + |P_i \cap P_{i+1}|$. Следовательно, мы имеем следующее:

Лемма 15.6. $|P_i| \leqslant |P_{i+1}|$.

Теорема 15.4. Для всех таких i и j, что $|P_i| = |P_j|$, P_i и P_j не пересекаются по вершинам.

Доказательство. Предположим, что $|P_i| = |P_j|$, $i < j$, и P_i и P_j пересекаются по вершинам. Тогда существуют такие k и l, что $i \leqslant k < l \leqslant j$, P_k и P_l пересекаются по вершинам, а для любого r ($k < r < l$) P_r не пересекается по вершинам с P_k и P_l. Тогда P_l — добавляющий путь по отношению к $M_k \oplus P_k$, поэтому $|P_l| \geqslant |P_k| + |P_k \cap P_l|$. Но $|P_l| = |P_k|$. Следовательно, $|P_k \cap P_l| = 0$. Таким образом, P_k и P_l не имеют общих ребер. Но если P_k и P_l имеют общую вершину v, то они имеют и общее ребро, которое инцидентно v и входит в $M_k \oplus P_k$. Следовательно, P_k и P_l не пересекаются по вершинам, и мы приходим к противоречию. Главный результат этого раздела формулируется в следующей теореме:

Теорема 15.5. Пусть s — мощность максимального паросочетания. Число различных целых чисел в последовательности $|P_0|, |P_1|, \ldots, |P_i|, \ldots$ меньше или равно $2\lfloor \sqrt{s} \rfloor + 2$.

Доказательство. Пусть $r = \lfloor s - \sqrt{s} \rfloor$. Тогда $|M_r| = r$, и по лемме 15.4

$$|P_r| \leqslant 2\frac{\lfloor s - \sqrt{s} \rfloor}{(s - \lfloor s - \sqrt{s} \rfloor)} + 1 \leqslant 2\lfloor \sqrt{s} \rfloor + 1.$$

Таким образом, для каждого $i < r$ $|P_i|$ есть одно из $\lfloor \sqrt{s} \rfloor + 1$ положительных нечетных чисел, меньших или равных $2\lfloor \sqrt{s} \rfloor + 1$. $|P_{r+1}|, |P_{r+2}|, \ldots, |P_s|$ привносят не более $s - r = \lceil \sqrt{s} \rceil$ различных целых чисел, а поэтому общее число целых чисел в последовательности $|P_0|, |P_1|, \ldots$ меньше или равно $\lfloor \sqrt{s} \rfloor + 1 + \lceil \sqrt{s} \rceil \leqslant 2\lfloor \sqrt{s} \rfloor + 2$. Результаты, сформулированные в лемме 15.6 и теоремах 15.4 и 15.5, позволяют рассматривать вычисление последовательности $M_0, M_1, \ldots$ как состоящее не более чем из $2\lfloor \sqrt{s} \rfloor + 2$ таких фаз, что добавляющие пути, находимые на каждой фазе, имеют одинаковую длину и не пересекаются по вершинам. Так как все добавляющие пути на какой-либо фазе не пересекаются по вершинам, то они являются добавляющими путями по отношению к паросочетанию, с которого началась эта фаза. Это привело Хопкрофта и Карпа к предложению следующего альтернативного пути для описания порядка вычисления максимального паросочетания.

Шаг **0**. Начать с нулевого паросочетания M, т. е. $M = \varnothing$.

Шаг **1**. Пусть $l(M)$ — длина кратчайшего добавляющего пути по отношению к M. Найти максимальное множество путей $\{Q_1, Q_2, \ldots, Q_t\}$ со следующими свойствами:

1) для каждого i Q_i — добавляющий путь по отношению к M и $|Q_i|=l(M)$,
2) Q_i не пересекаются по вершинам. HALT, если таких путей не существует.

Шаг 2. Положить $M=M\oplus Q_1\oplus Q_2\oplus\ldots\oplus Q_t$, идти к шагу 1. Из предыдущего обсуждения ясно, что шаги 1 и 2 приведенного выше вычисления будут выполняться не более чем $2\lfloor\sqrt{s}\rfloor+2$ раз, т. е. $O(n^{1/2})$ раз. Далее сложность вычисления в значительной степени зависит от сложности реализации шага 1. Для графа общего вида реализация этого шага довольно сложна, так как требует порождения всех добавляющих путей по отношению к данному паросочетанию и последующего выбора из них максимального множества кратчайших путей, не пересекающихся по вершинам.

Однако в случае двудольного графа возможна реализация шага 1 со сложностью $O(n^2)$, так что сложность вычисления максимальных паросочетаний для таких графов равна $O(n^{5/2})$. В следующем подразделе мы обсудим этот результат, также полученный Хопкрофтом и Карпом.

15.5.2. Алгоритм Хопкрофта и Карпа для нахождения максимального двудольного паросочетания

Пусть G — связный двудольный граф, множество вершин которого разбито на подмножества X и Y. Мы будем называть вершины из X верхними, а вершины из Y — нижними.

Способ выполнения шага 1, предложенный Хопкрофтом и Карпом и упомянутый ранее, состоит из двух стадий. Если паросочетание M не максимально, то на первой стадии выполнения шага 1 строится такой ориентированный граф G^*, что кратчайшие добавляющие пути по отношению к M взаимно-однозначно соответствуют ориентированным путям в графе G^*, которые начинаются в свободной нижней вершине, а заканчиваются в свободной верхней вершине. На второй стадии строится максимальное множество таких ориентированных путей в графе G^* с тем свойством, что они не пересекаются по вершинам. Эти пути дают требуемые кратчайшие добавляющие пути в графе G по отношению к M.

Граф G^* строится следующим образом. Прежде всего мы ориентируем ребра G таким образом, что добавляющие пути по отношению к M становятся ориентированными. В этой ориентации каждое ребро, не вошедшее в паросочетание, направлено из нижней вершины в верхнюю, а каждое ребро, вошедшее в паросочетание, — из верхней вершины в нижнюю. Пусть $\vec{G}$ — полученный ориентированный граф.

Пусть L_0 — множество свободных верхних вершин, для которых верно, что все ребра, инцидентные этим вершинам, не входят в паросочетание, т. е. ориентированы из нижних вершин в верхние.

Рассмотрим такое множество нижних вершин L_1, что существует ориентированное ребро из каждой из этих вершин по меньшей мере в одну верхнюю вершину из L_0.

Если L_1 содержит хотя бы одну свободную нижнюю вершину, то мы выделили добавляющий путь, показывающий, что длина кратчайшего добавляющего пути по отношению к M равна 1. Пусть

L_1' — множество свободных нижних вершин в L_1, а E_0' — множество ориентированных ребер, соединяющих L_1' с L_0. Тогда требуемый граф G^* имеет множество вершин $L_0 \cup L_1'$ и множество ребер E_0'.

Иначе (т. е. если L_1 не содержит свободных нижних вершин) пусть E_0 — множество ребер, связывающих L_1, с L_0, а L_2 — множество напарников нижних вершин из L_1. Пусть E_1 — множество соответствующих ребер, вошедших в паросочетание. Ясно, что L_2 — множество верхних вершин и ребра в E_1 направлены из L_2 в L_1.

Предположим, что мы построили множество вершин L_0, L_1, . . ., L_{2k} и множества ребер E_0, E_1, . . ., E_{2k}. Рассмотрим множество таких нижних вершин L_{2k+1}, которые не встречались в L_0, L_1, . . ., L_{2k}, что каждая из них смежна по меньшей мере с одной из верхних вершин множества L_{2k}.

Если L_{2k+1} содержит по меньшей мере одну свободную нижнюю вершину, то это означает, что мы нашли добавляющий путь, показывающий, что длина кратчайшего добавляющего пути по отношению к M равна $2k+1$. Пусть L_{2k+1}' — множество свободных нижних вершин в L_{2k+1}, а E_{2k}' — множество ориентированных ребер, соединяющих L_{2k+1} с L_{2k}.

Иначе (если L_{2k+1} не содержит свободных нижних вершин) пусть E_{2k} — множество ребер, соединяющих L_{2k+1}, с L_{2k}, и пусть L_{2k+2} — множество напарников нижних вершин в L_{2k+1}. Пусть также E_{2k+1} — множество соответствующих ребер, вошедших в паросочетание. Ясно, что процедура, описанная выше, закончится одним из двух следующих способов:

1. Для некоторого k множество L_{2k+1} содержит по крайней мере одну свободную нижнюю вершину. В этом случае мы останавливаем здесь процедуру и требуемый граф G^* имеет множество вершин $(L_0 \cup L_1 \cup \ldots \cup L_{2k} \cup L_{2k+1}')$ и множество ребер $(E_0 \cup E_1 \cup \ldots \ldots \cup E_{2k-1} \cup E_{2k}')$.
2. Множество L_{2k+1} пусто для некоторого k, т. е. мы не можем получить никаких новых нижних вершин, которые смежны с верхними вершинами из L_{2k}. Это свидетельствует о том, что данный граф не содержит добавляющих путей по отношению к M. Следовательно, паросочетание M максимально.

Таким образом, граф G^* обладает следующими свойствами:

1. Для нечетного i множество L_i состоит из нижних вершин. В противном случае оно состоит из верхних вершин.
2. Каждое ребро графа G^* ориентировано из вершины множества L_{i+1} в вершину множества L_i для некоторого i.
3. Граф G^* не имеет ориентированных циклов.
4. Кратчайший добавляющий путь в графе G по отношению к M взаимно-однозначно соответствует ориентированному пути в графе G^*, который начинается в свободной нижней и заканчивается в свободной верхней вершине.

Ясно, что сложность построения графа G^* составляет $O(n^2)$. Из свойства 4 следует, что порождение максимального множества не пересекающихся по вершинам добавляющих путей в графе G по отношению к M эквивалентно порождению максимального множества не пересекающихся по вершинам ориентированных путей в графе G^*, которые начинаются в свободных нижних и заканчиваются в свободных верхних вершинах. Такие ориентированные пути в графе G^* могут быть порождены за $O(n^2)$ шагов поиском в глубину в графе G^*, как мы увидим в алгоритме 15.6.

Представим алгоритм Хопкрофта и Карпа для максимального двудольного паросочетания. Не нарушая общности, предположим, что данный граф связен. Алгоритм состоит из двух процедур: PROC-HOPKARP и PROC-AUGMENT.

PROC-HOPKARP — главная процедура. По данному паросочетанию M она проверяет, является ли M максимальным паросочетанием. Если нет, то строит граф G^* и вызывает PROC-AUGMENT. PROC-AUGMENT порождает максимальное множество не пересекающихся по вершинам ориентированных добавляющих путей в графе G^* и выполняет необходимые расширения. Добавляющие пути, построенные при каком-либо выполнении этой процедуры, соответствуют путям в отдельной фазе последовательности P_0, P_1, ...

В PROC-AUGMENT мы используем массив PATH, в котором запоминаются вершины пути по мере его порождения. Переменная POINT указывает на самый верхний элемент в массиве PATH.

Алгоритм 15.6. Максимальное двудольное паросочетание (Хопкрофт и Карп).

PROC-HOPKARP.

H1. G — данный связный двудольный граф. Пусть M_0 — нулевое паросочетание, т. е. $M_0=\varnothing$. Положить $i=0$.

H2. Построить из графа G ориентированный граф $\overline{G}$, ориентируя каждое не вошедшее в паросочетание ребро (по отношению к M_i) графа G из нижней вершины в верхнюю и каждое вошедшее в паросочетание ребро (по отношению к M_i) из верхней вершины в нижнюю.

H3. Пусть L_0 — множество свободных верхних вершин по отношению к M_i. Если L_0 не пусто, то идти к шагу H4. Иначе HALT. (Паросочетание M_i максимально.)

H4. Построить последовательность $L_0, L_1, \ldots$ подмножеств множества вершин V графа $\overline{G}$ и последовательность $E_0, E_1, \ldots$ подмножеств множества таких ребер E графа $\overline{G}$, что

1) $E_i=\{(u, v) \mid (u, v)\in\overline{E}, v\in L_i, u\notin L_0\cup L_1\cup\ldots\cup L_i\}$, $i=0, 1, 2, \ldots$,
2) $L_{i+1}=\{u \mid$ для некоторого v, $(u, v)\in E_i\}$, $i=0, 1, 2, \ldots$,
3) $m^*=\min\{i \mid L_i\cap\{\text{свободные нижние вершины}\}\neq\varnothing\}$.

H5. Если m^* определено, то идти к шагу H6, иначе HALT. (Не существует добавляющего пути по отношению к M_i, и, следовательно, M_i — максимальное паросочетание.)

H6. Построить подграф $G^*=(V^*, E^*)$ графа $\overline{G}$, где $V^*=L_0\cup L_1\cup\ldots\cup L_{m^*-1}\cup\{(L_{m^*}\cap\{\text{свободные нижние вершины}\})\}$ и $E^*=E_0\cup E_1\cup\ldots\cup E_{m^*-2}\cup\{(u, v) \mid v\in L_{m^*-1}$ и $u\in\{\text{свободные нижние вершины}\}\}$.

H7. Выполнить процедуру PROC-AUGMENT и идти к шагу H2.

PROC-AUGMENT.

A1. Положить POINT=0. (Содержание массива PATH очевидно.) Выбрать свободную нижнюю вершину y графа G^*, которая еще не помечена как «рассмотренная», и идти к шагу A2. Если такой свободной нижней вершины не существует, то идти к шагу A8.

A2. Пометить y как «рассмотренную». Положить POINT=POINT+1 и PATH (POINT)=y. (y помещается сверху в массив PATH.)

A3. Выбрать ребро (y, x), которое еще не помечено как «рассмотренное», и идти к шагу A5. Если такого ребра не существует (мы не можем получить добавляющий путь, который включает вершину y), то выполнить следующие действия:
1) если POINT=1, то идти к шагу A1;
2) если POINT>1, то идти к шагу A4.

A4. Положить POINT=POINT—2 и y=PATH (POINT). (Удаляем сверху из массива PATH два элемента.) Идти к шагу A3.

A5. Пометить (y, x) как «рассмотренное». Если x уже помечена как «рассмотренная» (x уже встречалась в некотором кратчайшем добавляющем пути, либо такого пути, содержащего x, не существует), то идти к шагу A3. Иначе пометить x как «рассмотренную» и положить POINT= =POINT+1 и PATH (POINT)=x (x помещается сверху в массив PATH).

A6. Если x — свободная вершина по отношению к M_i, то идти к шагу A7. Иначе положить y-напарник вершины x и идти к шагу A2.

A7. Добавляющий путь P по отношению к M_i найден. Вершинами P являются вершины PATH (1), PATH (2), . . ., PATH (POINT). Положить $M_{i+1}=M_i \oplus P$. (M_{i+1} — новое паросочетание после расширения.) Положить $i=i+1$ и идти к шагу A1.

A8. (Все кратчайшие не пересекающиеся по вершинам пути в графе G^* выделены и соответствующие расширения осуществлены.) Конец процедуры.

Можно показать, что сложность выполнения PROC-AUGMENT равна $O(n^2)$. Так как в любой фазе построения графа G^* выполнение PROC-AUGMENT производится только один раз и по теореме 15.5 существует $O(n^{1/2})$ фаз, то отсюда следует, что сложность алгоритма 15.6 равна $O(n^{5/2})$.

В качестве примера рассмотрим граф G, изображенный на рис. 15.13, *а*, где пунктирные линии обозначают ребра паросочетания M. В этом графе нижние вершины пронумерованы числами 1—12, а верхние вершины — числами —1— —10.

Вначале мы ориентируем ребра графа G так, что ребра, не вошедшие в паросочетание, ориентируются из нижних вершин в верхние, а ребра, вошедшие в паросочетание, ориентируются из верхних вершин в нижние. Получающаяся ориентация ребер изображена на рис. 15.13, *а*.

Множества L_0, L_1, L_2, L_3, получаемые в результате работы алгоритма, есть множества $L_0=\{-5, -6\}$, $L_1=\{1, 4, 5\}$, $L_2=\{-7, -8, -9\}$, $L_3=\{6, 8, 9, 11\}$.

Тогда $m^*=3$. Так как вершины 6 и 8 множества L_3 не являются свободными нижними вершинами, то они не входят в граф G^*. Граф G^* для паросочетания M изображен на рис. 15.13, *б*. Отметим, что существует только два добавляющих пути в графе G^*. Однако они не являются не пересекающимися по вершинам. Поэтому PROC-AUGMENT получит только один добавляющий путь для фазы, которая начинается с паросочетания M. Если ребро {—6,5; —9,9}—

полученный добавляющий путь, то после расширения мы получим новое паросочетание, в котором вновь введенными ребрами являются (—6,5) и (—9,9). Ребро (5, —9), входящее в паросочетание, удаляется из него.

Для дальнейшего изучения мы рекомендуем работу [15.37], в которой показано, как частный случай более общего результата, что максимальное паросочетание в двудольном графе может быть построено за $O(n^{5/2})$ шагов.

Авторы работы [15.38] разработали $O(n^{5/2})$-алгоритм для максимального паросочетания в графе общего вида. Этот алгоритм использует метод, основанный на поисках в ширину и в глубину. При разработке этого алгоритма также использовались некоторые идеи, выдвинутые ранее [15.29, 15.36].

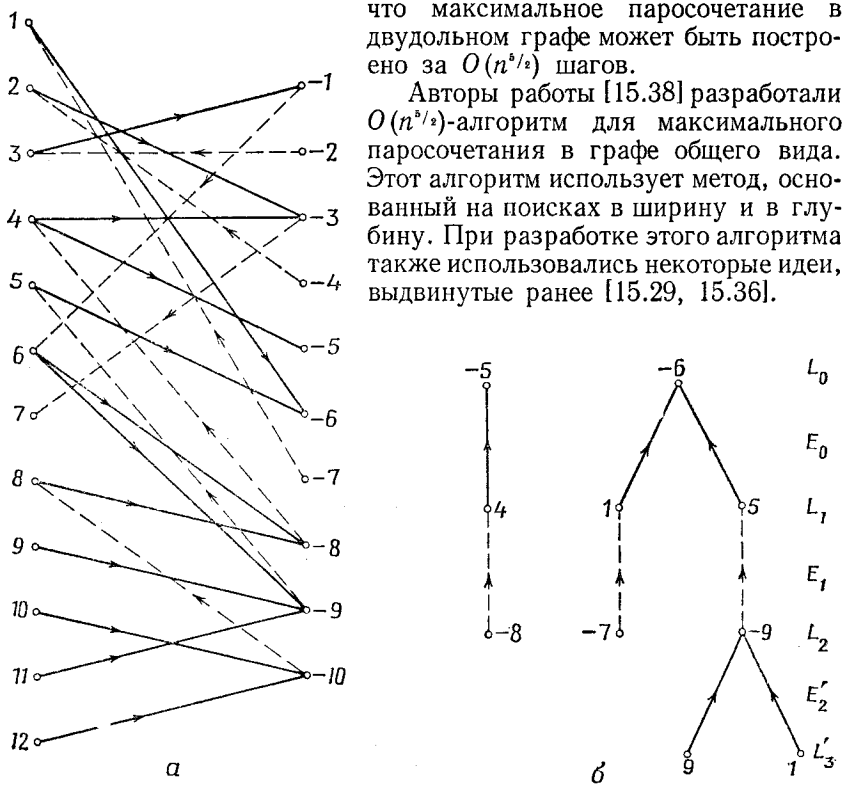


Рис. 15.13.

15.6. Совершенное паросочетание, оптимальное назначение и составление расписаний

В этом разделе обсуждаются задачи оптимального назначения и составления расписаний, изучение которых включает и теорию паросочетаний. Получение оптимального назначения требует в качестве первого шага построения совершенного паросочетания в со-

ответствующем двудольном графе. Имея это в виду, вначале обсудим алгоритм построения совершенного паросочетания в двудольном графе.

15.6.1. Совершенное паросочетание

Рассмотрим задачу о назначении, в которой n рабочих могут выполнить один или несколько из n видов работ; необходимо определить, можем ли мы назначить рабочих на все виды работ по одному рабочему на каждый вид работы, который он может выполнить. Если представим рабочих множеством вершин $X=\{x_1, x_2, \ldots, x_n\}$, а виды работ — множеством вершин $Y=\{y_1, y_2, \ldots, y_n\}$ двудольного графа G, в котором x_i смежна с y_j тогда и только тогда, когда рабочий x_i может выполнить вид работы y_j, то ясно, что задача о назначении сводится к задаче определения, имеет ли граф G совершенное паросочетание.

Одним из методов решения этой задачи было бы применение алгоритма 15.6 и выделение максимального паросочетания. Если это паросочетание состоит из n ребер, то граф имеет совершенное паросочетание, а полученное максимальное паросочетание есть не что иное, как совершенное паросочетание.

Главным недостатком такого метода является то, что если граф не имеет совершенного паросочетания, то мы узнаем об этом только в конце процедуры. Рассмотрим алгоритм, который находит совершенное паросочетание в графе G либо останавливается, когда находит такое подмножество S множества X, что $|\Gamma(S)|<|S|$, где $\Gamma(S)$ — множество вершин, смежных с вершинами из подмножества S. Из теоремы Холла 8.13 ясно, что в последнем случае в графе не существует совершенного паросочетания.

Основная идея, лежащая в основе этого алгоритма, очень проста. Как обычно, мы начинаем с некоторого начального паросочетания M. Если M насыщает все вершины множества X, то это как раз то паросочетание, которое мы ищем. Иначе в общем случае мы выбираем ненасыщенную вершину u множества X и систематически ищем добавляющий путь P, начинающийся в u. При поиске такого пути мы запоминаем количество вершин, выбранных из множества X, число смежных с ними вершин и число вершин, выбранных из множества Y.

Двудольность графа гарантирует нам то, что мы не получим при поиске цикла нечетной длины и, следовательно, «цветок» не может образоваться. Как мы видели в разд. 15.4.1, замкнутый чередующийся путь четной длины не дает возможности расширить имеющееся паросочетание M. Следовательно, граф поиска, который мы строим, всегда будет деревом. Это дерево называется *венгерским деревом*. На любом этапе, если мы находим добавляющий путь, мы выполняем дополнение и получаем новое паросочетание, которое насыщает во множестве X на одну вершину больше, чем было до этого. Если

такого пути не существует, что мы получаем множество $S \subseteq X$, нарушающее необходимое и достаточное условие существования совершенного паросочетания.

Пусть M — паросочетание в графе G, а u — ненасыщенная вершина в X. Дерево H графа G называется *M-чередующимся деревом* с корнем u, если 1) u принадлежит множеству вершин H и 2) для каждой вершины v из H единственный путь из u в v в H есть M-чередующийся путь (т. е. чередующийся путь по отношению к M).

Пусть S — подмножество множества вершин X, а T — подмножество множества вершин Y, которые встречаются в H.

Чередующееся дерево «растет» следующим образом. Вначале H состоит только из вершин u. Затем оно растет таким образом, что на каждом этапе существуют две возможности:

1. Все вершины в H, кроме u, насыщены (рис. 15.14, *а*).

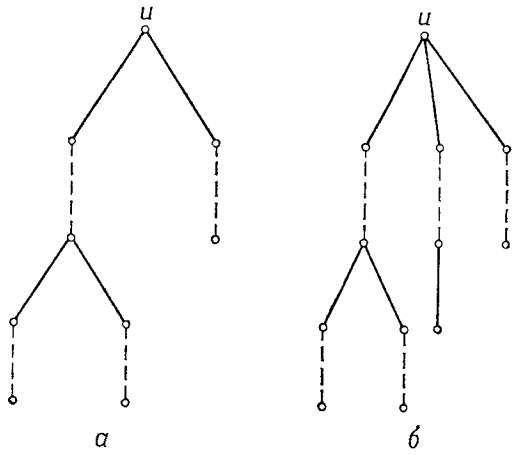


Рис. 15.14. Примеры чередующихся деревьев.

2. H содержит ненасыщенную вершину, отличную от u (рис. 15.14, *б*). В этом случае мы получаем добавляющий путь и, следовательно, получаем новое паросочетание.

В первом случае $\Gamma(S)=T$ либо $T \subset \Gamma(S)$.

1а. $\Gamma(S)=T$. Так как $|S|=|T|+1$ в дереве H, то мы получаем в этом случае $|\Gamma(S)|=|S|-1$ и множество S не удовлетворяет необходимому и достаточному условию теоремы Холла. Следовательно, можно сделать вывод, что в графе G нет совершенного паросочетания.

1б. $T \subset \Gamma(S)$. Тогда существует вершина y из Y, которая не встречается в T, но входит в $\Gamma(S)$. Путь эта вершина y будет смежна с вершиной x из S. Если y насыщена и z — ее напарник, то мы наращиваем H, добавляя вершины y и z, а также ребра (x, y) и (y, z).

Тогда мы вновь возвращаемся к первому случаю. Если y не насыщена, то мы наращиваем H, добавляя вершину y и ребро (x, y) и приходя ко второму случаю. Путь из u в y в H является добавляющим путем по отношению к M. Метод, описанный выше, реализуется следующим алгоритмом:

Алгоритм 15.7. Совершенное паросочетание.

S1. Пусть G — двудольный граф с разбиением множества вершин (X, Y), где $|X|=|Y|$. Пусть M_0— пустое паросочетание, т. е. $M_0=\varnothing$. Положить $i=0$.

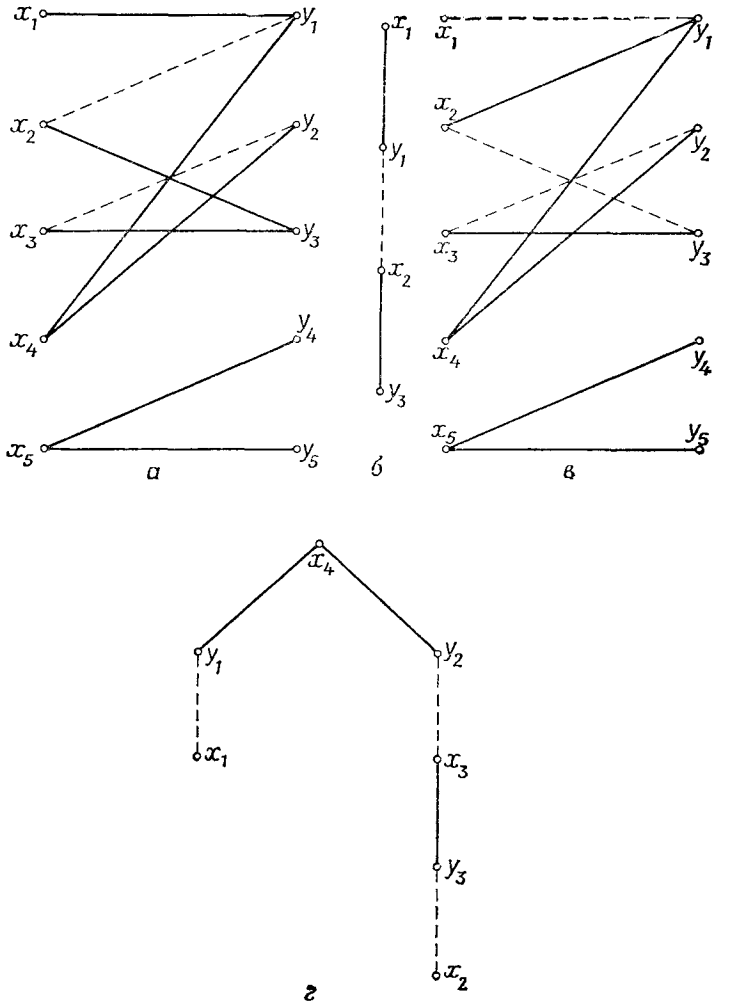


Рис. 15.15.

S2. Если все вершины из X насыщены в паросочетании M_i, то HALT. (M_i— совершенное паросочетание в графе G.) Иначе выбрать ненасыщенную вершину u в X и положить $S=\{u\}$, $T=\varnothing$.

S3. Если $\Gamma(S)=T$, то HALT. (Тогда $|\Gamma(S)|<|S|$ и, следовательно, в графе G нет совершенного паросочетания.) Иначе выбрать вершину y из $\Gamma(S)-T$.

S4. Если y не насыщена в M_i, то идти к шагу S5. Иначе положить z — напарник y, $S=S\cup\{z\}$ и $T=T\cup\{y\}$, после чего идти к шагу S3.

S5. (Найден добавляющий путь P). Положить $M_{i+1}=M_i\oplus P$ и $i=i+1$. Идти к шагу S2.

В качестве примера рассмотрим двудольный граф G, представленный на рис. 15.15, *а*. В этом графе ребра начального паросочетания M изображены штриховыми линиями. Вершина x_1 в M не насыщена. Строим M — чередующееся дерево с корнем x_1. Мы завершаем наращивание дерева в момент, показанный на рис. 15.15, *б*, где мы разместили добавляющий путь x_1, y_1, x_2, y_3. Затем мы расширяем M и получаем новое паросочетание, показанное на рис. 15.15, *в*.

Вершина x_4 в этом паросочетании не насыщена. Поэтому мы переходим к построению чередующегося дерева с корнем x_4 по отношению к новому паросочетанию. Построение этого дерева завершается на этапе, показанном на рис. 15.15, *г*. Дальнейший рост этого дерева невозможен, так как на этом этапе $\Gamma(S)=T$, где $S=\{x_1, x_2, x_3, x_4\}$, а $T=\{y_1, y_2, y_3\}$. Следовательно, граф с рис. 15.15, *а* не имеет совершенного паросочетания.

Иногда нас может заинтересовать получение совершенного паросочетания, обладающего определенным свойством. В работе [15.39] рассматривается алгоритм, применимый для некоторого класса таких задач.

15.6.2. Оптимальное назначение

Рассмотрим задачу о назначении, в которой каждый рабочий может выполнить любой вид работы. Очевидно, что в таком случае мы можем каждому рабочему назначить работу (конечно, мы, как и раньше, предполагаем, что имеется n рабочих и n работ). Фактически любое максимальное паросочетание является совершенным, и мы получим $n!$ таких паросочетаний. Задача, интересующая нас в этом случае, состоит в учете эффективности назначения рабочих на различные работы и получении такого назначения, которое максимизирует общую эффективность. Задача нахождения такого назначения известна как *задача об оптимальном назначении*.

Двудольный граф для этой задачи является полным, т. е. если $X=\{x_1, x_2, \ldots, x_n\}$ — множество вершин, представляющих рабочих, а $Y=\{y_1, y_2, \ldots, y_n\}$ — множество вершин, представляющих работы, то для всех i и j x_i смежна с y_j. Припишем каждому ребру (x_i, y_j) вес $w_{ij}=w(x_i, y_j)$, который показывает эффективность работы рабочего x_i при выполнении задания y_j (измеряемую в каких-либо единицах). Тогда задача об оптимальном назначении соответствует задаче определения в таком взвешенном графе совершенного паросочетания с максимальным весом. Такое паросочетание называется *оптимальным*.

Рассмотрим алгоритм сложности $O(n^4)$, предложенный Каном и Мункресом [15.40, 15.41] для задачи об оптимальном назначении. Мы будем придерживаться изложения в работе [15.35].

Пусть *допустимой вершинной разметкой* является такая функция f со значениями из множества действительных чисел, определенная на множестве $X \cup Y$, что $f(x)+f(y) \geqslant w(x, y)$ для всех $x \in X$ и $y \in Y$, тогда $f(x)$ называется *меткой* вершины x.

Например, следующая разметка является допустимой вершинной разметкой: $f(x)=\max\{w(x, y)\}$, если $x \in X$; $f(y)$, если $y \in Y$. Из этого следует, что допустимая вершинная разметка всегда существует независимо от значений весов ребер.

Пусть для данной допустимой вершинной разметки f величина E_f — множество всех ребер (x, y) графа G, для которых $f(x)+f(y)=w(x, y)$. Остовный подграф графа G со множеством ребер E_f называется *подграфом равенств*, соответствующим f. Мы будем обозначать этот граф G_f.

Следующая теорема, связывающая подграф равенств и оптимальные паросочетания, образует основу алгоритма Кана — Мункреса.

Теорема 15.6. Пусть f — допустимая вершинная разметка графа $G=(V, E)$. Если G_f содержит совершенное паросочетание M^*, то M^*— оптимальное паросочетание в графе G.

Доказательство. Предположим, что G_f содержит совершенное паросочетание M^*. Так как G_f — остовный подграф графа G, то M^*— совершенное паросочетание в графе G. Пусть $w(M^*)$ — вес M^*, т. е. $w(M^*)=\sum_{e \in M^*} w(e)$. Поскольку каждое ребро $e \in M^*$ принадлежит подграфу равенств и каждая вершина графа G инцидентна точно одному ребру из M^*, то мы получаем

$$w(M^*)=\sum_{e \in M^*} w(e)=\sum_{v \in V} f(v). \tag{15.28}$$

С другой стороны, если M — произвольное совершенное паросочетание в графе G, то

$$w(M)=\sum_{e \in M} w(e) \leqslant \sum_{v \in V} f(v). \tag{15.29}$$

Объединяя выражения (15.28) и (15.29), получим $w(M^*) \geqslant w(M)$. Таким образом, M^*— оптимальное паросочетание в графе G.

В алгоритме Кана — Мункреса мы начинаем с произвольной допустимой вершинной разметки f и определяем соответствующий граф G_f. Затем выбираем начальное паросочетание M в G_f и применяем алгоритм 15.7. Если получается совершенное паросочетание в G_f, то, согласно теореме 15.6, это паросочетание оптимально. В противном случае алгоритм 15.7 заканчивается после построения паросочетания M', которое несовершенно, получая M' — чередующееся дерево H, не содержащее добавляющего по отношению к M' пути и не может быть наращено далее в G_f. Тогда мы изменяем f, получая допустимую вершинную разметку f' с тем свойством, что и M', и H содержится в $G_{f'}$ и H можно расширить в $G_{f'}$. Мы осуществляем такие изменения допустимой вершинной разметки всякий раз, когда это необходимо до тех пор, пока не обнаружим совершенное паросочетание в каком-либо из подграфов равенств. Детали алгоритма Кана — Мункреса представлены ниже.

Алгоритм 15.8. Оптимальное назначение (Кан и Мункрес).

S1. G — данный полный двудольный граф с разбиением множества вершин (X, Y), где $|X|=|Y|$. $W=[w_{ij}]$ — заданная матрица весов. Положить $i=0$.

S2. Начать с произвольной допустимой вершинной разметки f в графе G. Найти подграф равенств G_f и выбрать начальное паросочетание M_i в G_f.

S3. Если все вершины в X насыщены в M_i, то M_i — совершенное паросочетание и, следовательно, по теореме 15.6 это — оптимальное паросочетание. HALT. Иначе пусть u — ненасыщенная вершина в X. Положить $S=\{u\}$ и $T=\varnothing$.

S4. Пусть $\Gamma_f(S)$ — множество вершин, которые смежны в графе G_f с вершинами из множества S. Если $\Gamma_f(S)\supset T$, то идти к шагу S5. Иначе (т. е. если $\Gamma_f(S)=T$) вычислить

$$d_f = \min_{\substack{x\in S \\ y\notin T}} \{f(x)+f(y)-w(x, y)\} \tag{15.30}$$

и получить новую допустимую вершинную разметку f', определяемую выражением

$$f'(v)=\begin{cases} f(v)-d_f, & \text{если } v\in S; \\ f(v)+d_f, & \text{если } v\in T; \\ f(v) & \text{в остальных случаях.} \end{cases} \tag{15.31}$$

Отметим, что $d_f>0$ и $\Gamma_f(S)=T$. Заменить f на f' и G_f на $G_{f'}$.

S5. Выбрать вершину y из $\Gamma_f(S)-T$. Если y не насыщена в M_i, то идти к шагу S6. Иначе положить z-напарник y в M_i, $S=S\cup\{z\}$, $T=T\cup\{y\}$ и идти к шагу S4.

S6. (Найден добавляющий путь P). Положить $M_{i+1}=M_i\oplus P$ и $i=i+1$. Идти к шагу S3.

Чтобы проиллюстрировать алгоритм Кана — Мункреса, рассмотрим полный двудольный граф G, имеющий следующую матрицу весов:

$$W=\begin{bmatrix} 4 & 4 & 1 & 3 \\ 3 & 2 & 2 & 1 \\ 5 & 4 & 4 & 3 \\ 1 & 1 & 2 & 2 \end{bmatrix}.$$

Начальную допустимую вершинную разметку f для графа G можно выбрать следующим образом: $f(x_1)=4$, $f(x_2)=3$, $f(x_3)=5$, $f(x_4)=2$, $f(y_1)=f(y_2)=f(y_3)=f(y_4)=0$.

Подграф равенств G_f, представлен на рис. 15.16, *а*. Применяя алгоритм 15.6, мы определяем, что G_f не имеет совершенного паросочетания, поскольку для множества $S=\{x_1, x_2, x_3\}$, $T=\Gamma(S)=\{y_1, y_2\}$. Используя выражение (15.30), определяем $d_f=1$. Тогда, используя выражение (15.31), получаем следующую новую разметку: $f'(x_1)=3$, $f'(x_2)=2$, $f'(x_3)=4$, $f'(x_4)=2$; $f'(y_1)=1$, $f'(y_2)=1$, $f'(y_3)=0$, $f'(y_4)=0$.

Подграф равенств $G_{f'}$ представлен на рис. 15.16, *б*. С помощью алгоритма 15.6 мы получаем совершенное паросочетание M, состоящее

из ребер (x_1, y_2), (x_2, y_1), (x_3, y_3), (x_4, y_4). Это паросочетание является оптимальным.

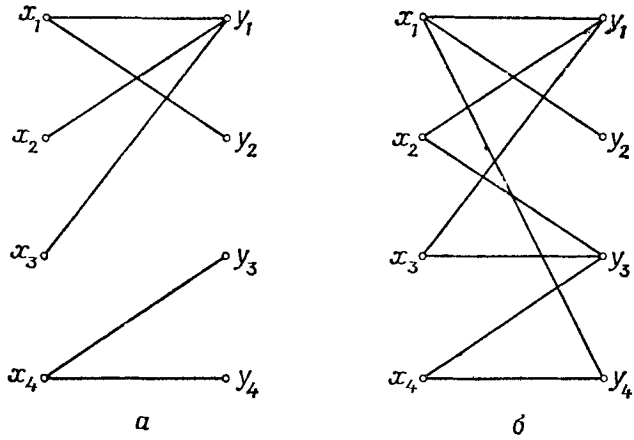


Рис. 15.16.

В работе [15.42] рассмотрен алгоритм для некоторого класса задач о взвешенных паросочетаниях, которые возникают в определенных приложениях, связанных с составлением расписаний и оптимальным назначением.

15.6.3. Составление расписаний

В школе работают p учителей $x_1, x_2, \ldots, x_p$ и имеется q классов $y_1, y_2, \ldots, y_q$. Известно, что учитель x_i должен провести в классе y_j p_{ij} занятий; необходимо составить расписание таким образом, чтобы время проведения занятия было наименьшим из возможных. Такая задача является частным случаем задачи о *составлении расписаний.*

Предположим, что мы построили двудольный граф $G=(X, Y)^*$, в котором вершины из X представляют учителей, вершины из Y — классы, а вершина $x_i \in X$ соединена с вершиной $y_j \in Y$ p_{ij} параллельными ребрами. Так как в любое время каждый учитель может преподавать не более чем в одном классе и каждый класс может находиться не более чем на одном занятии, то расписание занятий для одного урока соответствует паросочетанию в графе G и наоборот, каждое паросочетание соответствует возможному назначению учителей по классам для проведения одного урока. Таким образом, задача о составлении расписаний, упомянутая выше, есть задача разбиения ребер графа G на возможно меньшее число паросочетаний.

Из следствия 8.22.1 вытекает, что минимальное число паросочетаний в любом разбиении множества ребер двудольного графа G

равно максимальной степени этого графа G. Доказательство этого утверждения включает следующую процедуру определения разбиения на наименьшее число паросочетаний множества ребер графа:
Шаг 1. Пусть G — данный двудольный граф. Положить $i=0$ и $G_0=G$.
Шаг 2. Построить паросочетание M_i графа G_i, которое насыщает все вершины с максимальной степенью в графе G_i.
Шаг 3. Удалить M_i из графа G_i. Пусть G_{i+1} — получающийся в результате граф. Если G_{i+1} не имеет ребер, то $M_0, M_1, \ldots, M_i$ — требуемое разбиение множества ребер графа G. Иначе положить $i=i+1$ и идти к шагу 1.
Ясно, что сложность приведенной выше процедуры зависит от сложности реализации шага 2, который требует определения паросочетания, насыщающего все вершины с максимальной степенью в двудольном графе $G=(X, Y)$. Такое паросочетание можно найти следующим образом (теорема 8.21):

Пусть X_a — множество вершин с максимальной степенью в X, а Y_b — множество вершин с максимальной степенью в Y. Пусть G_a — подграф графа G, образуемый ребрами, инцидентными вершинам из X_a. Аналогично G_b — подграф, образуемый ребрами, инцидентными вершинам из Y_b.

Согласно теореме 8.22, существует паросочетание M_a, которое насыщает все вершины из X_a. Паросочетание M_a — максимальное паросочетание в G_a. Аналогично в G_b существует максимальное паросочетание M_b, которое насыщает все вершины в Y_b. Следуя процедуре, используемой в доказательстве теоремы 8.21, мы можем получить из M_a и M_b паросочетание M, которое насыщает вершины в X_a и Y_b. Тогда M — требуемое паросочетание, насыщающее все вершины максимальной степени в графе G.

Сложность выделения M_a и M_b совпадает со сложностью выделения максимального паросочетания, т. е. $O(n^{5/2})$, где n — число вершин в графе G. Легко показать, что сложность построения M по M_a и M_b есть $O(n^2)$. Таким образом, сложность выполнения шага 2 равна $O(n^{5/2})$. Так как шаг 2 будет повторен Δ раз, где Δ — максимальная степень вершин в графе G и $\Delta \leqslant n$, то сложность построения требуемого расписания равна $O(n^{7/2})$.

Анализ задачи о составлении расписаний в общем виде и ссылки на литературу по этой проблеме имеются в работе [15.43].

15.7. Потоки в транспортной сети

Транспортная сеть N есть связный ориентированный граф, который не имеет петель и удовлетворяет следующим условиям:
1. Существует только одна вершина с нулевой полустепенью захода; эта вершина называется *источником* и обозначается через s.

2. Существует только одна вершина с нулевой полустепенью исхода; эта вершина называется *стоком* и обозначается через t.
3. Каждому ориентированному ребру $e=(i, j)$ в сети N сопоставлено неотрицательное вещественное число, называемое *пропускной способностью* ребра; оно обозначается через $c(e)$ или $c(i, j)$. Если не существует ребра e, ориентированного из i в j, то мы доопределяем $c(e)=0$.

Транспортная сеть представляет собой модель сети перевозки продукции из центра производства к потребителю через связывающие их пути. *Пропускная способность* ребра может рассматриваться как максимальная скорость, с которой продукция транспортируется вдоль этого ребра.

Потоком f в транспортной сети N является функция, сопоставляющая каждому ребру $e=(i, j)$ неотрицательное вещественное число $f(e)=f(i, j)$ так, что удовлетворяются следующие условия:

1. $f(i, j)\leqslant c(i, j)$ для любого ребра (i, j) в N. (15.32)
2. $\Sigma_{\text{по всем } j} f(i, j)=\Sigma_{\text{по всем } j} f(j, i)$ для всех $i\neq s, t$. (15.33)

Значение потока $f(e)$ в ребре e можно рассматривать как скорость, с которой материал перемещается вдоль e при потоке f. Условие

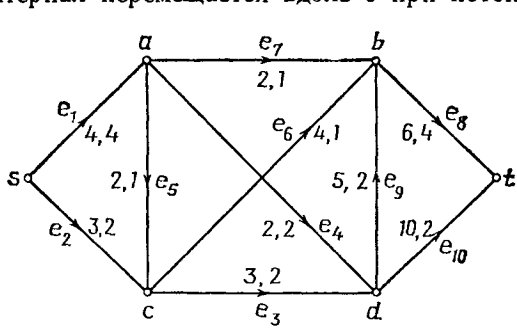


Рис. 15.17. Транспортная сеть.

(15.32), называемое *ограничением по пропускной способности*, требует, чтобы скорость потока вдоль ребра не превосходила пропускной способности ребра. Условие (15.33), называемое *условием сохранения*, требует, чтобы для каждой вершины i, исключая источник и сток, скорость, с которой материал доставляется в i, была бы равна скорости, с которой он удаляется из i.

Пример транспортной сети N с потоком f представлен на рис. 15.17. На нем рядом с каждым ребром e приведены пропускная способность $c(e)$ и поток $f(e)$ в указанном порядке.

Величина потока f, обозначаемая через $\mathrm{val}(f)$, определяется выражением

$$\mathrm{val}(f)=\sum_{\text{по всем } j} f(s, j). \tag{15.34}$$

Далее мы покажем, используя условие сохранения, что

$$\mathrm{val}(f) = \sum_{\text{по всем } j} f(s, j) = \sum_{\text{по всем } j} f(j, t). \tag{15.35}$$

Это будет только подтверждением интуитивно очевидного факта, что ввиду условия сохранения общее количество материала, выходящего из источника, равно общему количеству материала, входящего в сток.

Говорят, что поток f^* в транспортной сети N максимален, если не существует такого потока f в N, что $\mathrm{val}(f) > \mathrm{val}(f^*)$.

В следующем подразделе мы докажем теорему Форда и Фалкерсона о максимальном потоке и минимальном разрезе, после чего рассмотрим помечивающий алгоритм Форда и Фалкерсона определения максимального потока в транспортной сети. В заключение мы докажем теоремы Менгера, используя теорему о максимальном потоке и минимальном разрезе.

15.7.1. Теорема о максимальном потоке и минимальном разрезе

Будем говорить, что разрез $\langle S, \bar{S}\rangle$ в транспортной сети N разделяет источник s и сток t, если $s \in S$, а $t \in \bar{S}$. Такой разрез будем называть s—t-разрезом. Пропускная способность $c(K) = c(S, \bar{S})$ разреза $K = \langle S, \bar{S}\rangle$ определяется выражением

$$c(K) = \sum_{\substack{i \in S \\ j \in \bar{S}}} c(i, j). \tag{15.36}$$

Отметим, что пропускная способность ребер, которые ориентированы из $\bar{S}$ в S, не увеличивает пропускную способность разреза $K = \langle S, \bar{S}\rangle$. Мы обозначим через $f(S, \bar{S})$ сумму потоков в ребрах, ориентированных из S в $\bar{S}$. Величина $f(\bar{S}, S)$ определяется аналогично.

Например, рассмотрим разрез $K = \langle S, \bar{S}\rangle$ в транспортной сети, изображенной на рис. 15.17, где $S = \{a, b, c, s\}$ и $\bar{S} = \{d, t\}$.

Ребрами, ориентированными из S в $\bar{S}$, являются ребра e_3, e_4 и e_8. Следовательно, $c(S, \bar{S}) = c(e_3) + c(e_4) + c(e_8) = 11$, $f(S, \bar{S}) = f(e_3) + f(e_4) + f(e_8) = 8$, $f(\bar{S}, S) = f(e_9) = 2$.

Теорема 15.7. Для любого потока f и любого s—t-разреза $\langle S, \bar{S}\rangle$ в транспортной сети N

$$\mathrm{val}(f) = f(S, \bar{S}) - f(\bar{S}, S). \tag{15.37}$$

Доказательство. Из определений потока и величины потока имеем

$$\sum_{\text{по всем } j} f(i, j) - \sum_{\text{по всем } j} f(j, i) = \begin{cases} \mathrm{val}(f), & \text{если } i = s; \\ 0, & \text{если } i \in S - \{s\}. \end{cases}$$

Суммируя эти равенства по всем вершинам в S, получим

$$\sum_{i\in S}\sum_{\text{по всем } j} f(i, j)-\sum_{i\in S}\sum_{\text{по всем } j} f(j, i)=\operatorname{val}(f). \tag{15.38}$$

В левой части этого равенства $f(i, j)$ и $-f(i, j)$ для $i\in S$ и $j\in S$ появляются точно по одному разу и поэтому взаимно погашаются. Поэтому выражение (15.38) упрощается до $\sum_{i\in S}\sum_{j\in\bar{S}} f(i, j)-\sum_{i\in S}\sum_{i\in\bar{S}} f(j, i)=\operatorname{val}(f)$.

Таким образом, $f(S, \bar{S})-f(\bar{S}, S)=\operatorname{val}(f)$.

Отметим, что выражение (15.34) является частным случаем выражения (15.37).

Следствие 15.7.1. Для любого потока f и любого s—t-разреза $K=\langle S, \bar{S}\rangle$ в транспортной сети N:

$$\operatorname{val}(f)\leqslant c(S, \bar{S}). \tag{15.39}$$

Доказательство. Так как любая величина $f(i, j)$ неотрицательна, то из выражения (15.37) получим $\operatorname{val}(f)=f(S, \bar{S})-f(S, \bar{S})\leqslant f(S, \bar{S})\leqslant c(S, \bar{S})$.

Ребро (i, j) будем называть *f-насыщенным*, если $f(i, j)=c(i, j)$, и *f-ненасыщенным* в противном случае; оно будет *f-положительным*, если $f(i, j)>0$, и *f-нулевым*, если $f(i, j)=0$.

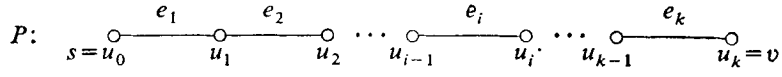
Отметим, что знак равенства в выражении (15.39) достигается тогда и только тогда, когда $f(\bar{S}, S)=0$ и $f(S, \bar{S})=c(S, \bar{S})$. Другими словами, $\operatorname{val}(f)=c(S, \bar{S})$ тогда и только тогда, когда все ребра, ориентированные из S в $\bar{S}$, являются f-насыщенными, а ребра, ориентированные из $\bar{S}$ в S,— f-нулевыми. s—t-разрез K в транспортной сети N является *минимальным*, если не существует такого s—t-разреза K' в N, что $c(K')<c(K)$.

Следствие 15.7.2. Пусть f — поток, а K — такой s—t-разрез, что $\operatorname{val}(f)=c(K)$. Тогда f — максимальный поток, а K — минимальный s—t-разрез.

Доказательство. Пусть f^*— максимальный поток, а K^*— минимальный s—t-разрез. Из следствия 15.7.1. имеем $\operatorname{val}(f^*)\leqslant c(K^*)$. Поэтому мы получаем $\operatorname{val}(f)\leqslant\operatorname{val}(f^*)\leqslant c(K^*)\leqslant c(K)$. Так как по предположению следствия $\operatorname{val}(f)=c(K)$, то отсюда следует, что $\operatorname{val}(f)=\operatorname{val}(f^*)=c(K^*)=c(K)$. Таким образом, f — максимальный поток, а K — минимальный s—t-разрез.

Перейдем к доказательству того, что величина максимального потока фактически равна пропускной способности минимального разреза.

Рассмотрим транспортную сеть N с потоком f. Пусть цепочка



является путем в N из источника в какую-либо вершину v. Отметим, что P — необязательно ориентированный путь.

Ребро e_i пути P называется *прямым ребром* пути P, если оно ориентировано из u_{i-1} в u_i. Иначе оно называется *обратным ребром* пути P. Пусть для каждого ребра e_i пути P

$$\varepsilon_i(P)=\begin{cases} c(e_i)-f(e_i), & \text{если } e_i\text{—прямое ребро;}\\ f(e_i), & \text{если } e_i\text{—обратное ребро.}\end{cases} \tag{15.40}$$

Сопоставим пути P неотрицательное число $\varepsilon(P)$, определяемое выражением

$$\varepsilon(P)=\min_i\{\varepsilon_i(P)\}. \tag{15.41}$$

Мы назовем путь *f-ненасыщенным*, если все прямые ребра пути являются *f-ненасыщенными*, а все обратные ребра пути — *f-положительными*.

s—t-путь P называется f-дополняющим путем, если P является f-ненасыщенным. Из выражений (15.40) и (15.41) следует, что для s—t-пути P неравенство $\varepsilon(P)>0$ справедливо тогда и только тогда, когда P является f-дополняющим. Для данного s—t-пути P в сети N мы можем определить новый поток $\hat{f}$ следующим образом:

$$\hat{f}(e)=\begin{cases} f(e)+\varepsilon(P), & \text{если } e\text{—прямое ребро;}\\ f(e)-\varepsilon(P), & \text{если } e\text{ — обратное ребро;}\\ f(e) & \text{в противном случае.}\end{cases}$$

Очевидно, что $\mathrm{val}(\hat{f})=\mathrm{val}(f)+\varepsilon(P)$.

Таким образом, $\mathrm{val}(\hat{f})>\mathrm{val}(f)$ тогда и только тогда, когда P является f-дополняющим путем. Другими словами, поток f не максимален, если существует f-дополняющий путь.

В качестве примера рассмотрим сеть N, изображенную на рис. 15.17. Пусть поток f будет таким, каким он показан на этом рисунке. В пути P, состоящем из e_2, e_5, e_7 и e_8, ребра e_2, e_8 и e_7 — прямые, а e_5 — единственное обратное ребро. По отношению к потоку f имеем равенство $\varepsilon_2(P)=1$, $\varepsilon_5(P)=1$, $\varepsilon_7(P)=1$ и $\varepsilon_8(P)=2$. Следовательно, $\varepsilon(P)=\min\{\varepsilon_2(P),\ \varepsilon_5(P),\ \varepsilon_7(P),\ \varepsilon_8(P)\}=1$.

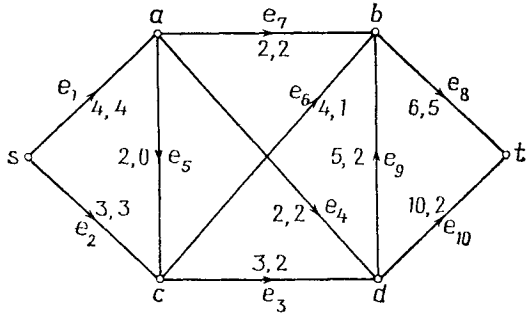


Рис. 15.18.

Так как $\varepsilon(P)>0$, то P является дополняющим путем. Измененный на пути P поток $\hat{f}$ представлен на рис. 15.18. Отметим, что $\hat{f}$ получен увеличением потока на всех прямых ребрах пути P на величину $\varepsilon(P)$ и его уменьшением на всех обратных ребрах пути P на величину $\varepsilon(P)$. Потоки в остальных ребрах остаются неизменными.

Теорема 15.8. Поток f в транспортной сети N максимален тогда и только тогда, когда в сети нет f-дополняющих путей.

Доказательство. Необходимость. Если в N существует f-дополняющий путь P, то ясно, что f — не максимальный поток, так как измененный на основе пути P поток $\hat{f}$ имеет большую величину, чем f.

Достаточность. Предположим, что N не содержит f-дополняющего пути. Пусть S — множество всех вершин в N, до которых существуют f-ненасыщенные пути из источника. Ясно, что $s \in S$. Далее $t \in \overline{S}$, так как в N не существует f-дополняющего пути.

Покажем, что $\mathrm{val}(f)=c(S, \overline{S})$, доказывая (согласно следствию 15.7.2), что f — максимальный поток, а $\langle S, \overline{S}\rangle$ — минимальный разрез.

Рассмотрим ориентированное ребро (v, w), для которого $v \in S$, а $w \in \overline{S}$. Так как $v \in S$, то существует f-ненасыщенный s—v-путь Q. Ребро (v, w) должно быть f-насыщенным, так как иначе Q можно расширить и получить f-ненасыщенный s—w-путь, что невозможно, так как $w \in \overline{S}$. Аналогично мы можем показать, что каждое ребро (v, w), ориентированное из $\overline{S}$ в S, должно быть f-нулевым.

Таким образом, $f(S, \overline{S})=c(S, \overline{S})$ и $f(\overline{S}, S)=0$. Следовательно, $\mathrm{val}(f)= =f(S, \overline{S})-f(\overline{S}, S)=c(S, \overline{S})$. Тогда из следствия 15.7.2 вытекает, что f — максимальный поток, а $\langle S, \overline{S}\rangle$ — минимальный разрез.

В ходе доказательства приведенной выше теоремы мы установили следующий хорошо известный результат Форда и Фалкерсона [15.44], а также Элиаса, Файнштайна и Шеннона [15.45].

Теорема 15.9. (Теорема о максимальном потоке и минимальном разрезе.) В транспортной сети величина максимального потока равна пропускной способности минимального разреза.

Теорему о максимальном потоке и минимальном разрезе можно использовать для доказательства нескольких комбинаторных результатов. Мы рассмотрим один из этих результатов в подразделе 15.7.4. Другие приведены в работах [15.1, 15.46].

15.7.2. Помечивающий алгоритм Форда и Фалкерсона

Рассмотрим теперь алгоритм Форда и Фалкерсона [15.44] для определения максимального потока в транспортной сети. Этот алгоритм, основанный на теореме 15.8, состоит из двух фаз.

На первой фазе по заданному потоку мы, используя помечивающую процедуру, проверяем, существует ли f-дополняющий путь. Если такого пути не существует, то, согласно теореме 15.8, данный поток f максимален. Иначе мы переходим ко второй фазе, в которой, используя метки, полученные в первой фазе, определим f — дополняющий путь P — и получим измененный на основе пути P поток $\hat{f}$. Затем мы повторим фазу *1* для нового потока $\hat{f}$. Отметим, что $\mathrm{val}(\hat{f})> \mathrm{val}(f)$.

В первой фазе метка вида (d_v, Δ_v) приписывается вершине v. Первый символ d_v в метке указывает на вершину, из которой v получила эту метку. Она также указывает и направление помечивания — прямое или обратное. Вскоре станет ясно, что если вершина v получает метку, то это значит, что существует f-ненасыщенный s—t-путь P и что для этого пути $\varepsilon(P)=\Delta_v$.

Первая фаза начинается с пометки источника парой $(-, \infty)$. Здесь значение d_s несущественно. Помечивание остальных вершин происходит в соответствии со следующими правилами:

Предположим, что вершина u помечена, а вершина v нет. Пусть e — ребро, связывающее u и v.

Прямое помечивание. Если $e=(u, v)$, то прямое помечивание v из u вдоль ребра e возможно, если $c(e)>f(e)$. Если такое помечивание осуществлено, то v получает метку (u^+, Δ_v), где $\Delta_v=\min\{\Delta_u, c(e)-f(e)\}$.

Обратное помечивание. Если $e=(v, u)$, то обратное помечивание v из u вдоль ребра e возможно, если $f(e)>0$. Если такое помечивание осуществлено, то v получает метку (u^-, Δ_v), где $\Delta_v=\min\{\Delta_u, f(e)\}$.

На первой фазе вершины помечаются только однажды. Эта фаза завершается, когда 1) вершина t получает метку либо 2) вершина t не помечена и ни одну из вершин больше нельзя пометить.

Если t получает метку в первой фазе, то из правил помечивания следует, что существует f-дополняющий путь P и $\varepsilon(P)=\Delta_t$. Во второй фазе путь P прослеживается в обратном направлении с помощью символов d_v и определяется измененный на основе P поток $\hat{f}$. Затем фаза 1 повторяется применительно к новому потоку $\hat{f}$. Если фаза 1 завершается, не приписав метки вершине t, то это означает, что f-дополняющего пути не существует и, следовательно, имеющийся поток максимален. Описание алгоритма Форда — Фалкерсона представлено ниже.

Алгоритм 15.9. Максимальный поток в транспортной сети (Форд и Фалкерсон).

S1. Выбрать какой-либо поток f в данной транспортной сети. Мы можем положить $f(e)=0$ для каждого ребра e в N.

S2. (Начинается фаза 1.) Пометить s парой $(-, \infty)$.

S3. Если существует непомеченная вершина, которую можно пометить с помощью прямого или обратного помечивания, то выбрать одну такую вершину v, пометить ее и перейти к шагу S4. Иначе идти к шагу S7.

S4. Если $v=t$, то идти к шагу S5. (Фаза 1 завершена.) Иначе идти к шагу S3.

S5. (Начинается фаза 2.) Пусть метка вершины v есть (d_v, Δ_v). Тогда выполнить следующие действия:
1) если $d_v=u^+$, то положить $f(u, v)=f(u, v)+\Delta_t$,
2) если $d_v=u^-$, то положить $f(v, u)=f(v, u)-\Delta_t$.

S6. Если $u=S$, то удалить все метки (фаза 2 завершена) и идти к шагу S2. Иначе положить $v=u$ и идти к шагу S5.

S7. (Полученный поток f — максимален.) HALT.

Проиллюстрируем работу вышеприведенного алгоритма.

Рассмотрим транспортную сеть N, представленную на рис. 15.19. На нем рядом с каждым ребром e показаны значения $c(e)$ и $f(e)$ соответственно. В качестве начального потока принимаем $f(e)=0$ для всех ребер в сети N. Начиная с метки $(-, \infty)$ для источника s, мы помечаем (шаг S3 алгоритма 15.9) вершины a, b, c, d и t в указанном порядке. Получаем метки a : $(s^+, 3)$, b : $(a^+, 3)$, c : $(a^+, 3)$, d : $(c^+, 2)$,

t : (b^+, 2). Фаза 1 завершается, так как вершина t уже помечена. В фазе 2 мы определяем дополняющий путь P и измененный поток f_1 (шаги S3 и S6).

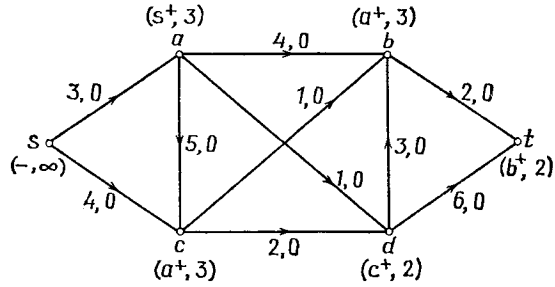
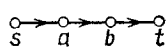


Рис. 15.19. Сеть для иллюстрации помечивающего алгоритма Форда — Фалкерсона.

Первый символ в метке t есть b^+. Это означает, что в пути P вершина b предшествует вершине t. Первый символ в метке b указывает, что вершина a предшествует вершине b в пути P. Аналогично мы видим, что s предшествует a пути P. Таким образом,



Все ребра в P являются прямыми. Поэтому для получения измененного потока f_1 мы увеличиваем потоки во всех ребрах пути P на $\Delta_t = 2$. Поток f_1 имеет величину, равную 2, и изображен на рис. 15.20, *а*.

Мы стираем метки всех вершин. Имея поток f_1, мы получаем затем новое множество меток, показанных на рис. 15.20, *а*. Добавляющий путь по отношению к f_1 состоит из вершин s, c, d, t в указанном порядке. Все ребра в этом пути P являются прямыми. Потоки в этих ребрах увеличиваются на 2. Получающийся поток f_2 показан на рис. 15.20, *б*.

Новое множество меток, основанное на f_2, показано на рис. 15.20, *б*. Дополняющий путь по отношению к f_2 состоит из вершин s, c, b, a, d, t. Ребро, соединяющее вершины a и b, является обратным ребром в этом пути. Все другие ребра пути являются прямыми. Увеличим потоки в прямых ребрах на 1 и настолько же уменьшим поток в обратном ребре. Получающийся в результате поток f_3 представлен на рис. 15.20, *в*.

Имея f_3, мы переходим к помечиванию вершин. Этот процесс завершается в состоянии, показанном на рис. 15.20, *в*, когда вершина t еще не помечена. Таким образом, не существует дополняющего пути по отношению к потоку f_3. Поэтому f_3 — максимальный поток.

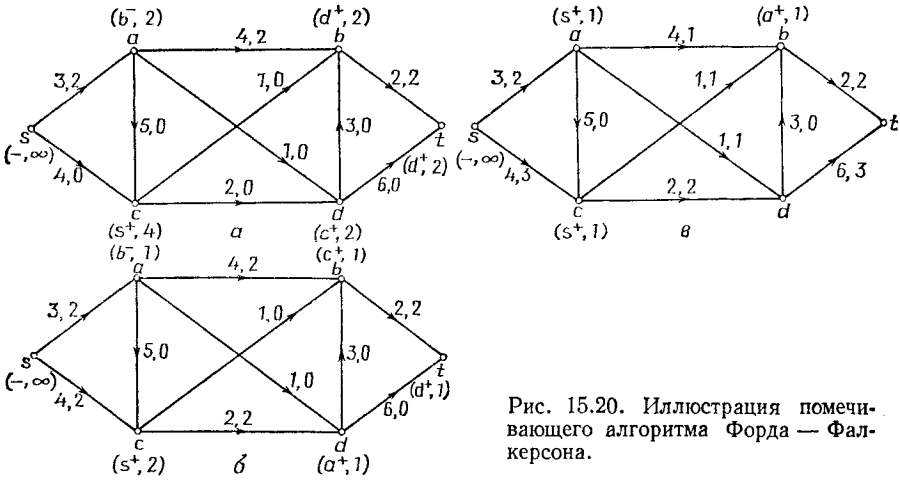


Рис. 15.20. Иллюстрация помечивающего алгоритма Форда — Фалкерсона.

Пусть S — множество вершин, помеченных на рис. 15.20, *в*. Тогда $S=\{s, a, b, c\}$. Поэтому из доказательства теоремы 15.8 ясно, что разрез $\langle S, \bar{S}\rangle$ является минимальным разрезом, а $\mathrm{val}(f_3)=$ =Пропускная способность $\langle S, \bar{S}\rangle$.

15.7.3. Модификация Эдмондса и Карпа алгоритма помечивания

В алгоритме Форда — Фалкерсона, который мы описали в предыдущем подразделе, вершины можно помечать в произвольном порядке. Другими словами, выбор дополняющего пути (когда он существует) делается произвольным образом. Проиллюстрируем на примере проблему, к которой может привести произвольность выбора.

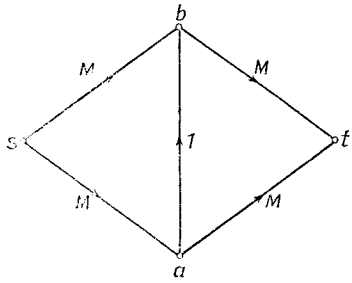


Рис. 15.21.

Рассмотрим транспортную сеть N, изображенную на рис. 15.21. Предположим, что мы начинаем выполнять помечивающий алгоритм с нулевого потока и поочередно используем пути $P_1: s, a, b, t$ и $P_2: s, b, a, t$ в качестве дополняющих путей. На каждом шаге величина потока увеличивается точно на 1, а максимальный поток величиной $2M$ достигается после $2M$ дополняющих шагов. Таким образом, число вычислительных шагов, выполняемых в этом случае, не ограничено функцией числа вершин и ребер сети N. Это число есть фактически функция от пропускной способности M, которая может быть произвольно большой.

Более того, Форд и Фалкерсон [15.1] показали, что их алгоритм может не найти решения, если пропускные способности ребер являются иррациональными числами. Они привели пример, в котором величина потока стремится к $^1/_4$ величины максимального потока при бесконечном увеличении числа шагов.

Для устранения этого недостатка Эдмондс и Карп [15.4] предложили усовершенствование помечивающего алгоритма: на каждом шаге поток увеличивается вдоль кратчайшего пути. Под кратчайшим путем здесь понимается путь, имеющий наименьшее число ребер. Очевидно, что кратчайший дополняющий путь будет выбран, если в процедуре помечивания мы рассматриваем вершины по правилу «первым помечен — первым рассмотрен», т. е. если вершина v помечена до вершины u, то мы рассматриваем вершину v до вершины u. Рассмотрение помеченной вершины v означает помечивание (когда это возможно) всех непомеченных вершин, смежных v.

Эдмондс и Карп показали, что эта модификация гарантирует независимость числа вычислительных шагов, требуемых для реализации помечивающего алгоритма, от пропускных способностей ребер. Перейдем к доказательству этого результата.

Рассмотрим поток f в транспортной сети N. Пусть

$$P:\quad \underset{s=u_0}{\circ}\overset{e_1}{\text{———}}\underset{u_1}{\circ}\overset{e_2}{\text{———}}\underset{u_2}{\circ}\ \cdots\ \underset{u_{i-1}}{\circ}\overset{e_i}{\text{———}}\underset{u_i}{\circ}\ \cdots\ \underset{u_{k-1}}{\circ}\overset{e_k}{\text{———}}\underset{u_k=t}{\circ}$$

дополняющий путь. Напомним, что

$$\varepsilon_i(P)=\begin{cases} c(e_i)-f(e_i), & \text{если } e_i\text{—прямое ребро;}\\ f(e_i), & \text{если } e_i\text{—обратное ребро}\end{cases}$$

$$\text{и } \varepsilon(P)=\min_i\{\varepsilon_i(P)\}.$$

Таким образом, $\varepsilon(P)=\varepsilon_i(P)$ для некоторого i. Тогда соответствующее ребро e_i называется *узким местом*.

Предположим, что помечивающий алгоритм начинает с начального потока f_0 и строит последовательность потоков $f_1, f_2, f_3, \ldots$.

Отметим, что, когда ребро e является узким местом в прямом направлении в дополняющем пути, оно становится насыщенным при увеличении потока. Если e является узким местом в обратном направлении, то при таком возрастании поток через e уменьшается до нуля. Это замечание приводит к следующему результату:

Лемма 15.7. Если $k<p$ и e — узкое место в прямом (обратном) направлении в дополняющем пути, который меняет как f_k на f_{k+1}, так и f_p на f_{p+1}, то существует такое $l\,(k<l<p)$, что e используется как обратное (прямое) ребро в дополняющем пути, который меняет f_l на f_{l+1}.

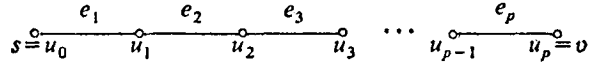
Путь $\lambda^i(u, v)$ — длина кратчайшего f_i-ненасыщенного пути из u в v. Этот путь, конечно, не обязательно будет ориентированным. Далее заметим, что ребро e используется как прямое ребро в пути, только если оно не насыщено, т. е. $f_i(e)<c(e)$, и ребро e используется как обратное ребро, только если $f_i(e)>0$.

Лемма 15.8. Для каждой вершины v и каждого $k=0, 1, 2, \ldots$,

$$\lambda^k(s, v) \leqslant \lambda^{k+1}(s, v), \tag{15.42}$$
$$\lambda^k(v, t) \leqslant \lambda^{k+1}(v, t). \tag{15.43}$$

Доказательство. Мы докажем соотношение (15.42). Соотношение (15.43) доказывается аналогично.

Предположим, что не существует f_{k+1}-ненасыщенного пути из s в v. Тогда $\lambda^{k+1}(s, v)$ будем считать бесконечной и неравенство (15.42) выполняется тривиально. Поэтому предположим, что



кратчайший f_{k+1}-ненасыщенный путь из s в v. Предположим, что e_i используется как прямое ребро в пути P. Тогда $f_{k+1}(e_i)<c(e_i)$. Поэтому 1) $f_k(e_i)<c(e_i)$ либо 2) $f_k(e_i)=c(e_i)$, и поэтому e_i использовано как обратное ребро в дополняющем пути, который меняет f_k на f_{k+1}. В первом случае

$$\lambda^k(s, u_i) \leqslant \lambda^k(s, u_{i-1}) + 1, \tag{15.44}$$

так как кратчайший f_k-ненасыщенный путь из s в u_{i-1}, за которым следует e_i в прямом направлении, является f_k-ненасыщенным путем из s в u_i. Очевидно, что в последнем случае

$$\lambda^k(s, u_{i-1}) = \lambda^k(s, u_i) + 1. \tag{15.45}$$

В любом случае соотношение (15.44) выполняется.

Аналогично мы можем показать, что (15.44) верно, когда e_i используется как обратное ребро в пути P.

Суммируя (15.44) для $i=1, 2, \ldots, p$ и замечая, что $\lambda^k(s, u_0)=0$, получим $\lambda^k(s, u_p) \leqslant p = \lambda^{k+1}(s, v)$. Таким образом, соотношение (15.42) доказано.

Лемма 15.9. Если используется принцип «первый помечен — первый рассмотрен», а $k<l$ и e используется как прямое (обратное) ребро в дополняющем пути, который меняет f_k на f_{k+1}, и как обратное (прямое) ребро в дополняющем пути, который меняет f_l на f_{l+1}, то $\lambda^l(s, t) \geqslant \lambda^k(s, t) + 2$.

Доказательство. Предположим, что e ориентировано из u в v. Тогда

$$\lambda^k(s, v) = \lambda^k(s, u) + 1, \tag{15.46}$$

так как e используется как обратное ребро при увеличении f_k. Далее,

$$\lambda^l(s, t) = \lambda^l(s, v) + 1 + \lambda^l(u, t), \tag{15.47}$$

так как e используется как обратное ребро при увеличении f_l. Но $\lambda^l(s, v) \geqslant \lambda^k(s, v)$ и

$$\lambda^l(u, t) \geqslant \lambda^k(u, t). \tag{15.48}$$

Тогда из выражений (15.46)—(15.48) получим $\lambda^l(s, t) \geqslant \lambda^k(s, u) + \lambda^k(u, t) + 2 = \lambda^k(s, t) + 2$.

Теорема 15.10. (Эдмондс и Карп.) Если в помечивающем алгоритме Форда — Фалкерсона каждое увеличение потока выполняется вдоль кратчайшего дополняющего пути, то максимальный поток можно получить не более чем после $m(n+2)/2$ приращений величины потока, где m — число ребер, а n — число вершин в транспортной сети.

Доказательство. Рассмотрим какое-либо ребро e, ориентированное из u в v. Рассмотрим далее такую последовательность потоков $f_{k_1}, f_{k_2}, \ldots$, где $k_1 < k_2 < \ldots$, что e используется как прямое ребро при увеличении f_{k_1} и является узким местом в этот момент. Согласно лемме 15.7, существует такая последова-

тельность $l_1, l_2, \ldots$, что $k_1<l_1<k_2<l_2<k_3<\ldots$, и e используется как обратное ребро при увеличении f_{l_i}.

По лемме 15.8 $\lambda^{k_i}(s, t)+2\leqslant\lambda^{l_i}(s, t)$ и $\lambda^{l_i}(s, t)+2\leqslant\lambda^{k_i+1}(s, t)$. Поэтому $\lambda^{k_1}(s, t)+4(j-1)\leqslant\lambda^{k_j}(s, t)$. Так как $\lambda^{k_j}(s, t)\leqslant n-1$, а $\lambda^{k_1}(s, t)\geqslant 1$, то мы получаем $1+4(j-1)\leqslant n-1$, или $j\leqslant(n+2)/4$.

Таким образом, e можно использовать как узкое место в прямом направлении не более чем $(n+2)/4$ раз. Аналогично его можно использовать как узкое место в обратном направлении не более чем $(n+2)/4$ раз. Поэтому каждое ребро может служить узким местом не более чем $(n+2)/2$ раз. Поэтому общее число увеличений потока не превышает $m(n+2)/2$.

В доказательстве приведенной выше теоремы не делается никаких предположений о природе пропускных способностей ребер, кроме их неотрицательности. Таким образом, из теоремы ясно, что если используется принцип «первый помечен — первый рассмотрен», то алгоритм 15.9 заканчивается для любых вещественных неотрицательных пропускных способностей за конечное число увеличений потока. Так как дополняющий путь можно найти за $O(m)$ шагов, то из теоремы 15.10 следует, что алгоритм 15.9 имеет сложность $O(m^2 n)$.

В работе [15.47] описывается класс сетей, для которых необходимо $O(n^3)$ увеличений потока при условии выполнения каждого увеличения вдоль кратчайшего дополняющего пути. Поэтому верхнюю границу в теореме 15.10 нельзя улучшить, если не рассматривать линейный коэффициент.

В работе [15.48] показано, как улучшить эффективность алгоритма 15.9, выделяя в одном из применений помечивающей процедуры все кратчайшие пути для увеличения потока. Эта идея подобна идее, использованной Хопкрофтом и Карпом [15.36] при построении максимального паросочетания в двудольном графе. В данном случае алгоритм имеет сложность $O(mn^2)$.

Недавно был представлен [15.49] алгоритм сложности $O(n^3)$. Впоследствии авторы работы [15.50] дали более простой алгоритм со сложностью $O(n^3)$. Этот и подобные алгоритмы описаны в работе [15.51].

Была изучена [15.37] сложность алгоритма Диница для особого класса транспортных сетей. Как мы упоминали ранее в разд. 15.5, этот результат был использован, чтобы показать, что максимальное паросочетание в двудольном графе можно построить за время, пропорциональное $O(n^{5/2})$.

15.7.4. Теоремы Менгера

В этом разделе мы докажем теоремы Менгера (15.12—15.15) для ориентированных и неориентированных графов, используя теорему о максимальном потоке и минимальном разрезе. Напомним, что мы сформулировали ранее теоремы Менгера (8.9 и 8.12) для неориентированных графов без доказательства. В последующем изложении

мы для целей общности будем допускать возможность того, что источник имеет ненулевую полустепень захода, а сток — ненулевую полустепень исхода. Наше доказательство теорем Менгера основано на следующем результате:

Теорема 15.11. Пусть N — транспортная сеть с источником s и стоком t, в которой каждое ребро имеет единичную пропускную способность. Тогда 1) величина максимального потока в сети N равна максимальному числу r не пересекающихся по ребрам ориентированных s—t-путей в сети N; 2) пропускная способность минимального разреза в сети N равна минимальному числу q ребер, удаление которых разрушает все ориентированные s—t-пути в сети N.

Доказательство. 1. Пусть f^*— максимальный поток в N, а N^*— ориентированный граф, полученный из N после удаления всех f^*-нулевых ребер. Так как пропускная способность каждого ребра равна единице, то ясно, что $f(e)=1$ для каждого ребра e в N^*. Таким образом, а) $d^+_{N^*}(s)-d^-_{N^*}(s)=\mathrm{val}\,(f^*)=d^-_{N^*}(t)-d_{N^*}(t)$. б) $d^+_{N^*}(v)=d^-_{N^*}(v)$ для всех $v\neq s, t$. Здесь $d^+_{N^*}(x)$ и $d^-_{N^*}(x)$ обозначают полустепени исхода и полустепени захода соответственно вершины x в графе N^*. Тогда (упражнение 5.8) существует $\mathrm{val}\,(f^*)$ не пересекающихся по ребрам ориентированных s—t-путей в графе N^*, а следовательно, и в графе N. Таким образом,

$$\mathrm{val}\,(f^*)\leqslant r. \tag{15.49}$$

Пусть $\{P_1, P_2, \ldots, P_r\}$ — набор r не пересекающихся по ребрам s—t-путей в графе N. Определим такой поток f, что

$$f(e)=\begin{cases} 1, & \text{если } e \text{ принадлежит какому-либо пути } P_i; \\ 0 & \text{в противном случае.}\end{cases}$$

Ясно, что $\mathrm{val}\,(f)=r$. Так как f^*— максимальный поток, то мы имеем

$$\mathrm{val}\,(f^*)\geqslant r. \tag{15.50}$$

Объединяя выражения (15.49) и (15.50), получим $\mathrm{val}\,(f^*)=r$.

2. Пусть $K^*=\langle S, \bar{S}\rangle$ — минимальный s—t-разрез в графе N. Удалим из N множество ребер $(S, \bar{S})$. Тогда в получающемся ориентированном графе не будет ни одного ориентированного s—t-пути. Поэтому

$$\mathrm{cap}\,(K^*)=|(S, \bar{S})|\geqslant q. \tag{15.51}$$

Пусть Z — множество из q ребер, удаление которых разрушает все ориентированные s—t-пути в N, а S — множество всех вершин, достижимых из s с помощью ориентированного пути, не содержащего ребра из Z. Ясно, что $K=\langle S, \bar{S}\rangle$—$s$—$t$-разрез в N. $(S, \bar{S})\subseteq Z$.
Поэтому

$$\mathrm{cap}\,(K^*)\leqslant \mathrm{cap}\,(K)=|(S, \bar{S})|\leqslant |Z|=q. \tag{15.52}$$

Объединяя выражения (15.51) и (15.52), получим $\mathrm{cap}\,(K^*)=q$.

Теорема 15.12. Пусть s и t — две вершины в ориентированном графе G. Тогда максимальное число не пересекающихся по ребрам ориентированных s—t-путей в G равно минимальному числу ребер, удаление которых разрушает все ориентированные s—t-пути в графе G.

Доказательство. Построим из графа G транспортную сеть N с источником s и стоком t, приписывая каждому ребру графа G единичную пропускную способность. Тогда доказываемая теорема является следствием теоремы 15.11 и теоремы о максимальном потоке и минимальном разрезе.

Для неориентированного графа G пусть $D(G)$ — ориентированный граф, полученный заменой каждого ребра e графа G парой противоположно ориентированных ребер, инцидентных тем же вершинам, что и e. Можно легко показать, что 1) существует взаимно-однозначное соответствие между путями в графе G и ориентированными путями в графе $D(G)$ и 2) для любых двух вершин s и t минимальное число ребер, удаление которых из графа разрушает все s—t-пути в графе G, равно минимальному числу ребер, удаление которых разрушает все ориентированные пути в графе $D(G)$. Из этих замечаний вытекает справедливость неориентированного варианта теоремы 15.12.

Теорема 15.13. Пусть s и t — две вершины неориентированного графа G. Максимальное число не пересекающихся по ребрам s—t-путей в графе G равно минимальному числу ребер, удаление которых разрушает все s—t-пути в графе G.

Вершинные аналоги теорем 15.12 и 15.13 доказываются следующим образом:

Пусть s и t — две несмежные вершины в ориентированном графе $G=(V, E)$. Построим из графа G ориентированный граф G' следующим образом:

1. Расщепим вершину $v \in V-\{s, t\}$ на две новые вершины v' и v'' и соединим их ориентированным ребром (v', v'').
2. Заменим каждое ребро графа G с конечной вершиной $v \in V-\{s, t\}$ на новое ребро, имеющее v' в качестве конечной вершины.
3. Заменим каждое ребро графа G с начальной вершиной $v \in V-\{s, t\}$ на новое ребро, имеющее v'' в качестве начальной вершины.

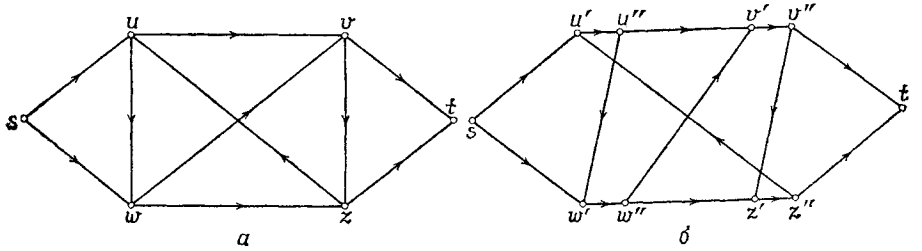


Рис. 15.22.
а — граф G; *б* — граф G'.

Граф G и соответствующий ему граф G' представлены на рис. 15.22. Нетрудно доказать, что:

1. Каждый ориентированный s—t-путь в графе G' соответствует ориентированному s—t-пути в графе G, который получается стягиванием всех ребер вида (v', v''), и, наоборот, каждый ориентированный s—t-путь в графе G соответствует s—t-пути в графе G', полученному расщеплением всех вершин пути, отличных от s и t.
2. Два ориентированных s—t-пути в графе G' не пересекаются по ребрам тогда и только тогда, когда соответствующие им пути в графе G не пересекаются по вершинам.
3. Максимальное число не пересекающихся по ребрам ориентированных s—t-путей в графе G' равно максимальному числу не пересекающихся по вершинам ориентированных s—t-путей в графе G.
4. Минимальное число ребер, удаление которых разрушает все ориентированные s—t-пути в графе G', равно минимальному числу вершин, удаление которых разрушает все ориентированные s—t-пути в графе G.

Из этих замечаний следует справедливость вершинного аналога теоремы 15.12.

Теорема 15.14. Пусть s и t — две несмежные вершины в ориентированном графе G. Тогда максимальное число не пересекающихся по вершинам ориентированных s—t-путей в графе G равно минимальному числу вершин, удаление которых разрушает все ориентированные s—t-пути в графе G.

Теореме 15.13 соответствует следующая теорема:

Теорема 15.15. Пусть s и t — две несмежные вершины в неориентированном графе G. Тогда максимальное число не пересекающихся по вершинам s—t-путей в графе G равно минимальному числу вершин, удаление которых разрушает все s—t-пути в графе G.

Доказательство. Достаточно применить теорему 15.14 к графу $D(G)$.

Стандартной ссылкой на теорию потоков в сетях является ссылка на работу [15.1], в которой рассматривается также задача поиска потока минимальной стоимости. Эти вопросы излагаются также в работах [15.7—15.9, 15.46, 15.52, 15.53].

15.8. Оптимальное ветвление

Рассмотрим взвешенный ориентированный граф $G=(V, E)$. Пусть $w(e)$ — вес ребра e. Вес подграфа графа G есть сумма весов всех ребер этого подграфа.

Подграф G_s графа G есть *ветвление* в графе G, если G_s не имеет ориентированных циклов и полустепень захода каждой вершины G_s не превышает 1. Ясно, что каждая компонента G_s является ориентированным деревом. Ветвление с максимальным весом называется *оптимальным ветвлением*.

В этом разделе мы обсуждаем алгоритм Эдмондса [15.54] для вычисления оптимального ветвления графа G. Наше обсуждение основывается на работе [15.55].

Ребро $e=(i, j)$, направленное из вершины i в вершину j, является критическим, если 1) $w(e)>0$ и 2) $w(e)\geqslant w(e')$ для каждого ребра $e'=(k, j)$, заходящего в j. Остовный подграф H графа G является критическим подграфом G, если 1) каждое ребро подграфа H критическое и 2) полустепень захода каждой вершины H не превышает 1.

Ориентированный граф G и критический подграф H графа G представлены на рис. 15.23.

Очевидно, что 1) каждая компонента критического графа содержит не более одного цикла — и такой цикл будет ориентированным циклом — и что 2) критический подграф без циклов является оптимальным ветвлением графа G.

Рассмотрим ветвление B. Пусть $e=(i,j)$ — ребро, не входящее в B, а e' — ребро B, заходящее в вершину j. Тогда e является *подходящим* ребром по отношению к B, если $B'=(B\cup e)-e'$ является ветвлением.

Например, ребра $\{e_1, e_2, e_3, e_4, e_7, e_8\}$ образуют ветвление графа, изображенного на рис. 15.23. Ребро e_6, не принадлежащее B, является подходящим по отношению к B, так как $(B\cup e_6)-e_7$ является ветвлением рассматриваемого графа.

Следующие две леммы легко доказываются и приводят к теореме 15.16, которая образует основу доказательства Карпа корректности алгоритма Эдмондса. Позже множество ребер подграфа H будет также обозначаться символом H.

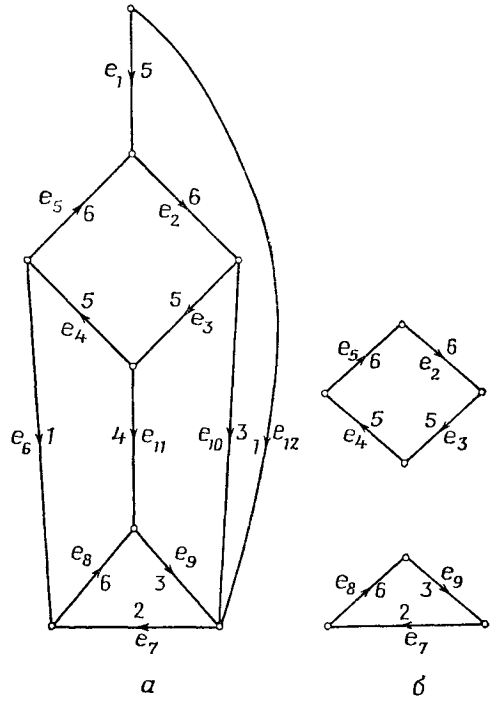


Рис. 15.23.

а — ориентированный граф G; *б* — критический подграф графа G

Лемма 15.10. Пусть B — ветвление, а $e=(i, j)$ — ребро, не принадлежащее B. Ребро e является подходящим по отношению к B тогда и только тогда, когда не существует ориентированного пути из j в i.

Лемма 15.11. Пусть B — ветвление, а C — такой ориентированный цикл, что не существует ребра из $C—B$, являющегося подходящим по отношению к B. Тогда $|C—B|=1$.

Теорема 15.16. Пусть H — критический подграф. Тогда существует такое оптимальное ветвление B, что для каждого ориентированного цикла C в H верно, что $|C—B|=1$.

Доказательство. Пусть B — оптимальное ветвление, которое содержит максимальное (среди всех оптимальных ветвлений) число ребер критического подграфа. Рассмотрим какое-либо ребро $e \in H—B$. Пусть e заходит в вершину j, а e' — ребро B, заходящее в вершину j. Если бы e было подходящим ребром, то $(B \cup e)—e'$ было бы оптимальным ветвлением, содержащим большее число ребер подграфа H, чем их содержит ветвление B, что противоречит допущению. Таким образом, ни одно ребро из $H—B$ не является подходящим по отношению к B. Поэтому, согласно лемме 15.11, для каждого цикла C в H имеем $|C—B|=1$.

Пусть $C_1, C_2, \ldots, C_k$— ориентированные циклы в подграфе H. Отметим, что никакие два цикла из подграфа H не могут иметь общее ребро. Другими словами, эти циклы не пересекаются по ребрам. Для каждого C_i пусть e_i^0 — ребро с минимальным весом в C_i.

Следствие 15.16.1. Существует такое оптимальное ветвление B, что 1) $|C_i - B| = 1$, $i = 1, 2, \ldots, k$ и 2) если ни одно ребро из $B - C_i$ не входит в вершину из C_i, $i = 1, 2, \ldots, k$, то

$$C_i - B = e_i^0. \tag{15.53}$$

Доказательство. Пусть среди всех оптимальных ветвлений, удовлетворяющих условию 1, B будет ветвлением, содержащим минимальное число ребер из множества $\{e_1^0, e_2^0, \ldots, e_k^0\}$. Тогда покажем, что B удовлетворяет условию 2.

Предположим, что это неверно для какого-либо i, т. е. $e_i^0 \in B$. Тогда ни одно ребро из $B - C_i$ не входит в вершину из C_i. Пусть $e = C_i - B$. Тогда $(B - e_i^0) \cup e$ является оптимальным ветвлением, которое удовлетворяет условию 1, но имеет меньше ребер из множества $\{e_1^0, e_2^0, \ldots, e_k^0\}$, чем B, что противоречит допущению.

Этот результат очень важен для разработки алгоритма Эдмондса. Он определяет, что мы можем ограничить поиск оптимального ветвления лишь теми ветвлениями, которые удовлетворяют условию (15.53).

Построим из данного графа G более простой граф G'. Покажем также, как построить из оптимального ветвления графа G' оптимальное ветвление графа G, которое удовлетворяет условию (15.53).

Как и ранее, пусть H — критический подграф графа G, а C_1, C_2, ... , C_k — ориентированные циклы в H. Пусть G' — граф, получающийся после стягивания всех ребер в каждом C_i ($i = 1, 2, \ldots, k$). В G' вершины каждого цикла C_i представлены единственной вершиной a_i, называемой *псевдовершиной*. Веса ребер графа G' совпадают с весами в графе G, кроме весов ребер, заходящих в псевдовершины. Эти веса модифицируются следующим образом:

Пусть $e = (i, j)$ — ребро такого графа G, что j — вершина некоторого цикла C_r, а i не принадлежит C_r. Тогда в графе G' ребро e заходит в псевдовершину a_r. Определим $\tilde{e}$ как единственное ребро в C_r, которое заходит в вершину j. Тогда в графе G' вес ребра e, обозначаемый $w'(e)$, определяется выражением

$$w'(e) = w(e) - w(\tilde{e}) + w(e_r^0). \tag{15.54}$$

Рассмотрим, например, ребро, заходящее в вершину ориентированного цикла $\{e_2, e_3, e_4, e_5\}$ критического подграфа графа G, изображенного на рис. 15.23. Тогда $\tilde{e}_1 = e_5$, а вес e_1 в графе G' определяется следующим образом: $w'(e_1) = w(e_1) - w(e_5) + w(e_4) = 5 - 6 + 5 = 4$.

Отметим, что e_4 — ребро в цикле $\{e_2, e_3, e_4, e_5\}$ с минимальным весом.

Пусть E и E' — множества ребер графов G и G' соответственно. Покажем, как построить ветвление B графа G, удовлетворяющее условию (15.53), используя ветвление B' графа G', и наоборот. Легко

показать для любого ветвления B графа G, удовлетворяющего условию (15.53), что

$$B'=B\cap E' \tag{15.55}$$

есть ветвление графа G'. Ветвление B', как оно определено выше, является единственным ветвлением, соответствующим данному ветвлению B.

Рассмотрим ветвление B' графа G'. Для каждого C_i определим C_i' следующим образом:

1. Если полустепень захода в B' псевдовершины a_i равна нулю, то $C_i'=C_i-e_i^0$.
2. Если полустепень захода в B' псевдовершины a_i не равна нулю и e — ребро B', входящее в псевдовершину a_i, то $C_i'=C_i-\tilde{e}$.

Тогда легко показать, что

$$B=B'\bigcup_{i=1}^{k} C_i' \tag{15.56}$$

есть ветвление графа G, которое удовлетворяет условию (15.53). Ветвление B, как оно определено выше, является единственным ветвлением для данного ветвления B'.

Таким образом, можно заключить, что существует взаимно-однозначное соответствие между множествами ветвлений графа G, которые удовлетворяют условию (15.53), и графа G'. Более того, для весов соответствующих ветвлений B и B' выполняется соотношение

$$w(B)-w(B')=\sum_{i=1}^{k} w(C_i)-\sum_{i=1}^{k} w(e_i^0). \tag{15.57}$$

Из этого свойства ветвлений B и B' следует, что если B — оптимальное ветвление графа G, удовлетворяющее условию (15.53), то B' — оптимальное ветвление графа G' и наоборот. Таким образом, нами доказана следующая теорема:

Теорема 15.17. Существует взаимно-однозначное соответствие между множеством всех оптимальных ветвлений в графе G, которые удовлетворяют условию (15.53), и множеством всех оптимальных ветвлений в графе G'. Алгоритм Эдмондса для построения оптимального ветвления основан на этой теореме и включает следующие шаги:

Алгоритм 15.10. Оптимальное ветвление (Эдмондс).

S1. Для данного графа $G=G_0$ построить последовательность графов G_0, G_1, G_2, . . ., G_k, где
1) G_k — первый граф в последовательности, критический подграф которого является ациклическим и
2) G_i $(1\leqslant i\leqslant k)$ получается из G_{i-1} стягиванием циклов в критическом подграфе H_{i-1} графа G_{i-1} и изменением весов в соответствии с выражением (15.54).

S2. Так как H_k — ациклический граф, то он является оптимальным ветвлением в G_k. Пусть $B_k=H_k$. Построить последовательность B_{k-1}, B_{k-2}, ..., B_0, где 1) B_i $(0\leqslant i\leqslant k-1)$ — оптимальное ветвление графа G_i и 2) B_i для $i\geqslant 0$ строится с помощью расширения псевдовершин, как это определяется выражением (15.56), в ветвлении B_{i+1}.

Например, пусть G_0 есть граф, изображенный на рис. 15.23, *а*, а H_0 — граф, изображенный на рис. 15.23, *б*. H_0 — критический подграф графа G_0. После стягивания ребер циклов из H_0 и изменения весов ребер мы получаем граф G_1, показанный на рис. 15.24, *а*. Критический подграф H_1 графа G_1 показан на рис. 15.24, *б*. H_1 — ациклический граф. Поэтому он является оптимальным ветвлением графа G_1. Оптимальное ветвление графа G_0 получается из H_1 расширением псевдовершин a_1 и a_2 (которые соответствуют двум ориентированным циклам в H_0), и оно изображено на рис. 15.24, *в*.

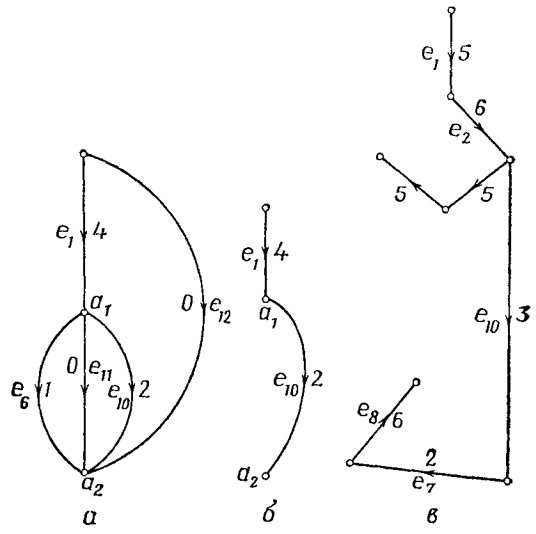


Рис. 15.24.

а — граф G_1; *б* — H_1 — критический подграф графа G_1; *в* — оптимальное ветвление для графа G (рис. 15, 23, *а*).

Тарьян [15.56] предложил $O(m \log n)$-реализацию алгоритма Эдмондса, где m — число ребер, а n — число вершин. Интересны также работы [15.57, 15.58], в которых независимо друг от друга был открыт алгоритм Эдмондса.

15.9. Замечания, касающиеся литературы

В литературе имеются сообщения о многих интересных алгоритмах, которые предназначены для решения графовых задач. Некоторые из них перечислены ниже с соответствующими ссылками на связанные с этими задачами работы.

Изоморфизм. Работы [15.59—15.62].

Планарность. Работы [15.26, 15.51, 15.63, 15.64].

Связность. Работы [15.65, 15.66].

Хроматическое число. Работы [15.68, 15.69].
Раскраска графа. Работы [15.8, 15.70—15.73].
Независимые множества и число устойчивости. Работы [15.74—15.79].
Минимальные остовы. Работы [15.10, 15.80—15.85].
Проблемы расширения. Работы [15.86, 15.87].
Параллельные вычисления. Работа [15.88].
Ветвления. Работы [15.89—15.92].
Оптимизация сети. Работа [15.93].
Расписания. Работы [15.94—15.100].
Теоретико-графовые исследования линейных уравнений. Работы [15.101—15.104].
Алгоритмы аппроксимации. Работы [15.105, 15.106].

Упражнения

15.1. Докажите корректность следующей модификации алгоритма Дейкстры для нахождения кратчайших путей между определенной вершиной s и всеми другими вершинами во взвешенном ориентированном графе G, который не содержит ориентированных циклов отрицательной длины. В последующем E — множество ребер графа G, а $w(u, v)$ — длина ориентированного ребра (u, v).

S1. Положить $\sigma(s)=0$, а $\sigma(u)=\infty$ для $u\neq s$.

S2. Положить $S=\{s\}$.

S3. Если $S=\varnothing$, то HALT, иначе выбрать такое u^*, что $u^*\in S$ и $\sigma(u^*)=\min\limits_{u\in S}\{\sigma(u)\}$.

S4. Для каждого такого v, что $(u^*, v)\in E$, положить $\sigma(v)=\min\{\sigma(v), \sigma(u^*)+w(u^*, v)\}$.
Если при этом уменьшается величина $\sigma(v)$, то положить $S=S\cup\{v\}$.

S5. Положить $S=S-\{u^*\}$ и идти к шагу S3 [15.4].

15.2. Докажите корректность следующего алгоритма, предложенного Данцигом [15.107], для выделения кратчайших путей между всеми парами вершин во взвешенном графе, который не содержит ориентированных циклов отрицательной длины. $W^k(i, j)$ — расстояние от i до j, где $1\leqslant i, j\leqslant k$ и ни одна вершина с номером, большим k, не используется в пути, вдоль которого определялось это расстояние. $w(i, j)=W^1(i, j)$ — длина ребра (i, j), а $w(ij)=\infty$, если такого ребра не существует. Кроме того, $w(i, i)=0$ для всех i.

S1. Положить $k=2$.

S2. Для $1\leqslant i<k$ выполнить $W^k(i, k)=\min\{w(i, k), W^{k-1}(i, j)+w(j, k)\}$, $W^k(k, i)=\min\limits_{1\leqslant j<k}\{w(k, i), w(k, j)+W^{k-1}(j, i)\}$.

S3. Для $1\leqslant i, j<k$ выполнить $W^k(i, j)=\min\limits_{1\leqslant j\leqslant n}\{W^{k-1}(i, j), W^k(i, k)+W^k(k, j)\}$.

S4. Если $k=n$, то STOP , иначе положить $k=k+1$ и идти к шагу S2.

Покажите также, как можно обнаружить циклы с отрицательной длиной при выполнении приведенного выше алгоритма.

15.3. Используя топологическую сортировку, сконструируйте алгоритм для нахождения кратчайшего пути из вершины s до всех остальных вершин в ациклическом взвешенном графе.

15.4. Предположим, что кратчайший путь из i в j не единствен. Какой из путей будет выбран для алгоритмов Флойда 15.2?

15.5. Докажите, что если вектор длин путей $(l_1, l_2, \ldots, l_n)$ удовлетворяет условию, налагаемому на характеристическую сумму, и если $l_1\leqslant l_2\leqslant l_3\leqslant\ldots\leqslant l_n$ и $l_j<l_{j+1}=l_{j+2}=\ldots=l_n$, то вектор $(l_1, l_2, \ldots, l_j, l_{j+1}-1, l_{j+d+1}, l_{j+d+2}, \ldots, l_n)$ также удовлетворяет условию, налагаемому на характеристическую сумму,

и длины путей расположены в неубывающем порядке [d то же, что и в выражении (15.17)].

15.6. Используя результат упражнения 15.5, покажите, как построить m-дерево, в котором порядок листьев слева направо совпадает с порядком величин в векторе длин путей.

15.7. Найдите минимальное дерево бинарного поиска для весов (1, 2, 3, 3, 4, 4).

15.8. Докажите, что замена шага E2 в алгоритме 15.5 на шаг «E2′. (Выбор ребра.) Выбрать ребро... . Если такого ребра не существует, то идти к шагу E7» не влияет на корректность алгоритма выделения максимального паросочетания [15.28].

15.9. Покажите, что если не существует ориентированного s—t-пути в транспортной сети N, то величина максимального потока и пропускная способность минимального разреза равны нулю.

15.10. Если $\langle S, \overline{S}\rangle$ и $\langle T, \overline{T}\rangle$ — минимальные разрезы в транспортной сети N, то покажите, что $\langle S \cup T, \overline{S \cup T}\rangle$ и $\langle S \cap T, \overline{S \cap T}\rangle$ также минимальные разрезы в N.

15.11. Покажите, что в любой транспортной сети с целочисленными пропускными способностями существует максимальный поток f, в котором $f(e)$ — целое число для любого ребра e в N.

15.12. Рассмотрим транспортную сеть N, в которой каждой вершине $v \neq s, t$ сопоставлено неотрицательное целое число $m(v)$. Покажите, как можно определить максимальный поток, в котором поток в каждую вершину $v \neq s, t$ не превышает $m(v)$ с помощью применения помечивающего алгоритма к модифицированной сети. (Примените построение, используемое при доказательстве теоремы 15.14.)

15.13. Рассмотрим транспортную сеть N, в которой определена также и нижняя граница потока в каждом ребре.

а) Найдите необходимые достаточные условия существования потока в сети N.

б) Модифицируйте помечивающий алгоритм для определения максимального потока в сети N [15.108].

15.14. Докажите, что поток в транспортной сети с нижними границами для потока в ребре существует тогда и только тогда, когда каждое ребро e принадлежит ориентированному циклу или ориентированному пути из s в t или ориентированному пути из t в s.

15.15. Опишите метод определения местонахождения в транспортной сети N ребра, которое обладает тем свойством, что увеличение его пропускной способности приводит к увеличению максимального потока в сети N.

15.16. Оптимальное ветвление не обязательно является оптимальным ориентированным остовом. Покажите, как можно модифицировать алгоритм Эдмондса для выделения ориентированного остова.

15.17. а) Докажите корректность следующего алгоритма Прима [15.81] для определения минимального взвешенного остова в связном взвешенном графе $G=(V, E)$. Выбрать произвольную вершину v в графе G. Среди всех ребер, входящих в v, выбрать ребро e_1 с минимальным весом. Стянем e_1, и пусть G' — полученный граф. Повторить эти действия для G' и продолжить до тех пор, пока не будет определено ребро e_{n-1}. Ребра $e_1, e_2, \ldots, e_{n-1}$ образуют минимальный взвешенный остов графа G.

б) Сконструируйте $O(n^2)$-реализацию алгоритма Прима.

15.18. Докажите теорему Холла 8.13, используя теорему 15.9 о максимальном потоке и минимальном разрезе, и наоборот.

15.19. Пусть $G=(X, Y)$ — двудольный граф, в котором существует полное паросочетание X в Y. Тогда известно, что существует такая вершина $v \in X$, что для каждого ребра e, инцидентного v, существует полное паросочетание, содержащее e (упражнение 8.16). Непосредственный алгоритм определения такой вершины будет требовать для своего выполнения $O(mn^{5/2})$ шагов. Попытайтесь сконструировать алгоритм с меньшей сложностью.

ЛИТЕРАТУРА

1.1. Whitney H., 2-Isomorphic Graphs., *Am. J. Math.*, 55, 245—254 (1933).
1.2. Berge C., Graphs and Hypergraphs, North Holland, Amsterdam, 1973.
1.3. Harary F. Graph Theory, Addison-Wesley, Reading, Mass., 1969. [Имеется перевод: Ф. Харари. Теория графов: — М.: Мир, 1973.]
1.4. J. A. Bondy, U.S.R. Murty, Graph Theory with Applications, Macmillan, London, 1976.
1.5. R. J. Wilson, Introduction to Graph Theory, Oliver and Boyed, Edinburgh, 1972. [Имеется перевод: Р. Уилсон. Введение в теорию графов.— М.: Мир, 1977.]
1.6. C. L. Liu, Introduction to Combinatorial Mathematics, McGraw-Hill, New York, 1968.
1.7. M. Behzad, G. Chartrand, Introduction to the Theory of Graphs, Allyn and Bacon, Boston, 1971.
1.8. N. Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, Englewood Cliffs, N.J., 1974.
2.1. S. Seshu, M. B. Reed, Linear Graphs and Electrical Networks, Addison-Wesley, Reading, Mass., 1961.
2.2. M. B. Reed. The Seg: A New Class of Subgraphs, *IRE Trans. Circuit Theory*, Vol. CT-8, 17—22 (1961).
2.3. W. H. Kim, R. T. Chien, Topological Analysis and Synthesis of Communication Networks, Colombia University Press, New York, 1962.
2.4. W. K. Chen, Applied Graph Theory: Graphs and Electrical Networks, North-Holland, Amsterdam, 1971.
2.5. W. Mayeda, Graph Theory, Wiley-Interscience, New York, 1972.
2.6. H. Whitney, On the Abstract Properties of Linear Dependence, *Am. J. Math.*, Vol. 57, 509—533 (1935).
2.7. C. St. J. A. Nash-Williams, Edge-Disjoint Spanning Trees of Finite Graphs, *J. London Math. Soc.*, Vol. 36, 445—450 (1961).
2.8. W. T. Tutte, On the Problem of Decomposing a Graph into *n* Connected Factors, *J. London Math. Soc.*, Vol. 36, 221—230 (1961).
2.9. R. L. Cummins, Hamilton Circuits in Tree Graphs, *IEEE Trans. Circuit Theory*, Vol. CT-13, 82-90 (1966).
3.1. V. Chvátal, On Hamilton's Ideals, *J. Combinatorial Theory B*, Vol. 12, 163—168 (1972).
3.2. G. A. Dirac, Some Theorems on Abstract Graphs, *Proc. London Math. Soc.*, Vol. 2, 69—81 (1952).
3.3. O. Ore, Arc Coverings of Graphs, *Ann. Mat. Pura Appl.*, Vol. 55, 315—321 (1961).
3.4. L. Pósa, A Theorem Concerning Hamilton Lines, *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, Vol. 7, 225—226 (1962).
3.5. J. A. Bondy, Properties of Graphs with Constraints on Degrees, *Studia Sci. Math. Hungar.*, Vol. 4, 473—475 (1969).
3.6. C. L. Liu, Introduction to Combinatorial Mathematics, McGraw-Hill, New York, 1968.

3.7. M. Behzad and G. Chartrand, Introduction to the Theory of Graphs, Allyn and Bacon, Boston, 1971.
3.8. S. Lin, Computer Solutions of the Traveling Salesman Problem, *Bell Syst. Tech. J.*, Vol. 44, 2245—2269 (1965).
3.9. M. Belmore and G. L. Nemhauser, The Traveling Salesman Problem: A Survey, *Operations Res.*, Vol. 16, 538—558 (1968).
3.10. M. Held and R. M. Karp, The Traveling Salesman Problem and Minimum Spanning Trees, *Operations Res.*, Vol. 18, 1138—1162 (1970).
3.11. M. Held and R. M. Karp, The Traveling Salesman Problem and Minimum Spanning Trees: Part II, *Math. Programming*, Vol. 1, 6—25 (1971).
3.12. C. Berge, Graphs and Hypergraphs, North Holland, Amsterdam, 1973.
3.13. J. A. Bondy and U.S.R. Murty, Graph Theory with Applications, Macmillan, London, 1976, chap. 4.
3.14. C. St. J. A. Nash-Williams, Hamilton Arcs and Circuits, in Recent Trends in Graph Theory, Springer, Berlin, 1971, pp. 197—210.
3.15. J. A. Bondy and V. Chvátal, A Method in Graph Theory, *Discrete Math.*, Vol. 15, 111—135 (1976).
3.16. C. St. J. A. Nash-Williams, Hamiltonian Circuits, in Studies in Graph Theory, Part. II, MAA Press, 1975, pp. 301—360.
3.17. L. Lesniak-Foster, Some Recent Results in Hamiltonian Graphs, *J. Graph Theory*, Vol. 1, 27—36 (1977).
3.18. P. Erdös and T. Gallai, On Maximal Paths and Circuits of Graphs, *Acta Math. Acad. Sci. Hung.*, Vol. 10, 337—356 (1959).
3.19. R. L. Cummins, Hamilton Circuits in Tree Graphs, *IEEE Trans. Circuit Theory*, Vol. CT-13, 82—90 (1966).
3.20. H. Shank, A Note on Hamilton Circuits in Tree Graphs, *IEEE Trans. Circuit Theory*, Vol. CT-15, 86 (1968).
4.1. S. MacLane and G. Birkhoff, Algebra, Macmillan, New York, 1967.
4.2. P. R. Halmos, Finite—Dimensional Vector Spaces, Van Nostrand Reinhold, New York, 1958.
4.3. F. E. Hohn, Elementary Matrix Algebra, Macmillan, New York, 1958.
4.4. W. K. Chen, On Vector Spaces Associated with a Graph, *SIAM J. Appl. Math.*, Vol., 20, 526—529 (1971).
4.5. T. W. Williams and L. M. Maxwell, The Decomposition of a Graph and the Introduction of a New Class of Subgraphs, *SIAM J. Appl. Math.*, Vol. 20, 385—389 (1971).
4.6. R. Gould, Graphs and Vectors Spaces, *J. Math. Phys.*, Vol. 37, 193—214 (1958).
5.1. S. W. Golomb, Shift Register Sequences, Holden-Day, San Francisco, 1967.
5.2. M. Hall, Jr., Combinarotial Theory, Blaisdell, Waltham, Mass., 1967.
5.3. T. Van Aardenne-Ehrenfest and N. G. de Bruijn, Circuits and Trees in Oriented Linear Graphs, *Simon Stevin*, Vol. 28, 203—217 (1951).
5.4. J. W. Moon, On Subtournaments of a Tournament, *Canad. Math. Bull.*, Vol. 9, 297—301 (1966).
5.5. C. Berge, Graphs and Hypergraphs, North Holland, Amsterdam, 1973.
5.6. J. A. Bondy and U.S.R. Murty, Graph Theory with Applications, Macmillan, London, 1976.
5.7. M. G. Kendall, Further Contributions to the Theory of Paired Comparisons, *Biometrics*, Vol. 11, 43—62 (1955).
5.8. J. W. Moon and N. J. Pullman, On Generalized Tournament Matrices, *SIAM Rev.*, Vol. 12, 384—389 (1970).
5.9. Ф. Харари Теория графов.—М.: Мир, 1973.
5.10. J. W. Moon, Topics on Tournaments, Holt, Rinehart and Winston, New York, 1968.
5.11. C. St. J. A. Nash-Williams, Hamiltonian Circuits, in Studies in Graph Theory, Part II, MAA Press, 1975, pp. 301—360.

5.12. D. E. Knuth, The Art of Computer Programming, Vol. 1: Fundamental Algorithms, Addison-Wesley, Reading Mass, 1968.
5.13. D. E. Knuth, The Art of Computer Programming, Vol. 3: Sorting and Searching, Addison-Wesley, Reading, Mass., 1973. [Имеется перевод: Д. Е. Кнут. Искусство программирования для ЭВМ. т. 3: Сортировка и поиск:— М.: Мир, 1978.]
5.14. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974. [Имеется перевод: А. Ахо, Дж. Хопкрофт, Дж. Ульман. Построение и анализ вычислительных алгоритмов.— М.: Мир, 1979.]
5.15. E. M. Reingold, J. Nievergelt, and N. Deo, Combinatorial Algorithms: Theory and Practice, Prentice Hall, Englewood Cliffs, N.J., 1977. [Имеется перевод: Э. М. Рейнгольд, Ю. Нивергельт, Н. Део. Комбинаторные алгоритмы: Теория и практика: — М.: Мир, 1980.]
5.16. F. Harary, R. Z. Norman, and D. Cartwright, Structural Models: An Introduction to the Theory of Directed Graphs, Wiley, New York, 1965.
5.17. V. Chvátal and L. Lovász, Every Directed Graph Has a Semi-Kernel, in Hypergraph Seminar (Eds. C. Berge and D. K. Ray-Chaudhuri), Springer, New York, 1974, p. 175.
6.1. F. E. Hohn, Elementary Matrix Algebra, Macmillan, New York, 1958.
6.2. W. K. Chen, Applied Graph Theory, North Holland, Amsterdam, 1971.
6.3. G. Kirchhoff, Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird, *Ann. Phys. Chem.*, Vol. 72, 497—508 (1947).
6.4. A. Cayley, A Theorem on Trees, *Quart. J. Math.*, Vol. 23, 376—378 (1889).
6.5. J. W. Moon, Various Proofs of Cayley's Formula for Counting Trees, in A Seminar on Graph Theory (Ed. F. Harary and L. W. Beinke), Holt, Rinehart and Winston, New York, 1967, pp. 70—78.
6.6. H. Prüfer, Neuer Beweis eines Satzes über Permutationen, *Arch. Math. Phys.*, Vol. 27, 742—744 (1918).
6.7. K. Sankara Rao, V. V. Bapeswara Rao, and V. G. K. Murti, Two—Three Admittance Products, *Electron. Lett.*, Vol. 6, 834—835 (1970).
6.8. W. T. Tutte, The Dissection of Equilateral Triangles into Equilateral Triangles, *Proc. Cambridge Phil. Soc.*, Vol. 44, 203—217 (1948).
6.9. F. Harary, The Determinant of the Adjacency Matrix of a Graph, *SIAM Rev.*, Vol. 4, 202—210 (1962).
6.10. C. L. Coates. Flow-Graph Solutions of Linear Algebraic Equations IRE Trans., *Circuit Theory*, Vol. CT-6, 170—187 (1959).
6.11. S. J. Mason, Feedback Theory: Some Properties of Signal Flow Graphs, *Proc. IRE.*, Vol. 41, 1144—1156 (1953).
6.12. S. J. Mason, Feedback Theory: Further Properties of Signal Flow Graphs, *Proc. IRE.*, Vol. 44, 920—926 (1956).
6.13. S. Seshu, M. B. Reed, Linear Graphs and Electrical Networks, Addison-Wesley, Reading, Mass., 1961.
6.14. W. Mayeda, Graph Theory, Wiley-Interscience, New York, 1972.
6.15. N. Deo. Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, Englewood Cliffs, N.J., 1974.
6.16. F. Harary and E. M. Palmer, Graphical Enumeration, Academic Press, New York, 1973. [Имеется перевод: Ф. Харари, Е. Палмер. Перечисление графов: — М.: Мир, 1977.]
6.17. B. R. Myers, Number of Trees in a Cascade of 2-Port Networks, IEEE Trans. *Circuit Theory*, Vol CT-14, 284—290 (1967).
6.18. B. R. Myers, Number of Spanning Trees in a Wheel, *IEEE Trans. Circuit Theory*, Vol. CT-18, 280—282 (1971).
6.19. S. D. Bedrosian, Number of Spanning Trees in Multigraph Wheels, *IEEE Trans. Circuit Theory*, Vol. CT-19, 77—78 (1972).
6.20. N. K. Bose, R. Feick, and F. K. Sun, General Solution to the Spanning Tree

Enumeration Problem in Multigraph Wheels, *IEEE Trans. Circuit Theory*, Vol. CT-20, 69—70 (1973).
6.21. M.N.S. Swamy and K. Thulasiraman, A Theorem in the Theory of Determinants and the Number of Spanning Trees of a Graph, *Proc. IEEE Int. Symp. on Circuits and Systems*, 153—156 (1976).
6.22. C. Berge, Graphs and Hypergraphs, North-Holland, Amsterdam, 1973.
6.23. L.P.A. Robichaud, M. Boisvert, and J. Robert, Signal Flow Graphs and Applications, Prentice-Hall, Englewood Cliffs, N.J., 1962.
6.24. N. Balabanian and T. A. Bickart, Electrical Network Theory, Wiley, New York, 1969.
6.25. Y. Kajitani and S. Ueno, On the Rank of Certain Classes of Cutsets and Tie-Sets of a Graph, *IEEE Trans. Circuits and Sys.*, Vol. CAS-26, 666—668 (1979).
7.1. K. Wagner, Bemerkungen zum Vierfarbenproblem, *Über Deutsh. Math. Verein*, Vol. 46, 26—32 (1936).
7.2. I. Fary, On Straight Line Representation of Planar Graphs, *Acta Sci. Math. Szeged*, Vol. 11, 229—233 (1948).
7.3. S. K. Stein, Convex Maps, *Proc. Am. Math. Soc.*, Vol. 2, 464—466 (1951).
7.4. H. Whitney, Non-Separable and Planar Graphs, *Trans. Am. Math. Soc.*, Vol. 34, 339—362 (1932).
7.5. C. Kuratowski, Sur le problème des courbes gauches en topologie, *Fund. Math.*, Vol. 15, 271—283 (1930).
7.6. См. [1.3].
7.7. W. T. Tutte, How to Draw a Graph, *Proc. London Math. Soc.*, Vol. 13, 743—767 (1963).
7.8. K. Wagner, Über eine Eigenschaft der ebenen Komplexe, *Math. Ann.*, Vol. 114, 570—590 (1937).
7.9. F. Harary and W. T. Tutte, A Dual Form of Kuratowski's Theorem, *Canad. Math. Bull.*, Vol. 8, 17—20 (1965).
7.10. S. MacLane, A Structural Characterization of Planar Combinatorial Graphs, *Duke Math. J.*, Vol. 3, 340—372 (1937).
7.11. H. Whitney, Planar Graphs, *Fund. Math.*, Vol. 21, 73—84 (1933).
7.12. T. D. Parsons, On Planar Graphs, *Am. Math. Monthly*, Vol. 78, 176—178 (1971).
7.13. S. Seshu, M. B. Reed, Linear Graphs and Electrical Networks, Addison-Wesley, Reading, Mass., 1961.
7.14. H. Whitney A Set of Topological Invariants for Graphs, *Am. J. Math.*, Vol. 55, 231—235 (1933).
7.15. O. Ore, The Four Colour Problem, Academic Press, New York, 1967.
7.16. J. A. Bondy and U.S.R. Murty, Graph Theory with Applications Macmillan, London, 1976.
7.17. J. E. Hopcroft and R. E. Tarjan, Efficient Planarity Testing, *J. ACM*, Vol. 21, 549—568 (1974).
7.18. N. K. Bose and K. A. Prabhu, Thickness of Graphs with Degree Constrained Vertices, *IEEE Trans. Circuits and Sys.*, Vol. CAS—24, 184—190 (1975).
7.19. W. T. Tutte, On the Nonbiplanar Character of the Complete 9-graph, *Canad. Math. Bull.*, Vol. 6, 319—330 (1963).
8.1. F. Harary The Maximum Connectivity of a Graph, *Proc. Nat. Acad. Sci., U.S.*, Vol. 48, 1142—1146 (1962).
8.2. J. A. Bondy, Properties of Graphs with Constraints on Degrees, Studia Sci, *Math. Hung.*, Vol. 4, 473—475 (1969).
8.3. G. Chartrand, A Graph Theoretic Approach to a Communication Problem, *SIAM J. Appl. Math.*, Vol. 14, 778—781 (1966).
8.4. S. L. Hakimi, On the Realizability of a Set of Integers as Degrees of the Vertices of a Graph, *SIAM J. Appl. Math.*, Vol. 10, 496—506 (1962).
8.5. V. Havel, A Remark on the Existence of Finite Graphs (in Hungarian), *Časopois Pest. Mat.*, Vol. 80, 477—480 (1955).

8.6. D. L. Wang and D. L. Kleitman, On the Existence of n-Connected Graphs with Prescribed Degrees ($n \geqslant 2$), *Networks*, Vol. 3, 225—239 (1973).
8.7. P. Erdös and T. Gallai, Graphs with Prescribed Degrees of Vertices (in Hungarian), *Mat. Lapok*, Vol. 11, 264—274 (1960).
8.8. См. [1.3].
8.9. J. Edmonds, Existence of k-Edge-Connected Ordinary Graphs with Prescribed Degrees, *J. Res. Nat. Bur. Stand. B.*, Vol. 68, 73—74 (1964).
8.10. D. L. Wang and D. J. Kleitman, A Note on n-Edge Connectivity, *SIAM J. Appl. Math.*, Vol. 26, 313—314 (1974).
8.11. K. Menger, Zur allgemeinen Kurventheorie, *Fund. Math.*, Vol. 10, 96—115 (1927).
8.12. H. Whitney, Congruent Graphs and the Connectivity of Graphs, *Am. J. Math.*, Vol. 54, 150—168 (1932).
8.13. W. T. Tutte, A Theory of 3-Connected Graphs, *Indag. Math.*, Vol. 23, 441—455 (1961).
8.14. L. R. Ford and D. R. Fulkerson, Maximal Flow Through a Network, *Canad. J. Math.*, Vol. 8, 399—404 (1956).
8.15. P. Elias, A. Feinstein, C. E. Shannon, A Note on the Maximum Flow Through a Network, *IRE Trans. Inform. Theory*, IT-2, 117—119 (1956). [Имеется перевод: П. Элиас, А. Файнстейн, К. Шеннон. О максимальном потоке через сеть: В сб. К. Шеннон. Работы по теории информации и кибернетике: — М.: ИЛ, 1963.]
8.16. P. Hall, On Representatives of Subsets, *J. London Math. Soc.*, Vol. 10, 26—30 (1935).
8.17. P. R. Halmosn, H. E. Vaughan, The Marriage Problem, *Am. J. Math.*, Vol. 72, 214—215 (1950).
8.18. D. König, Graphs and Matrices, *Mat. Fiz. Lapok*, Vol. 38, 116—119 (1931).
8.19. O. Ore, Graphs and Matching Theorems, *Duke Math J.*, Vol. 22, 625—639 (1955).
8.20. R. Rado, Note on the Transfinite Case of Hall's Theorem on Representatives, *J. London Math. Soc.*, Vol. 42, 321—324 (1967).
8.21. C. Berge, Two Theorems in Graph Theory, *Proc. Nat. Acad. Sci. U.S.*, Vol. 43, 842—844 (1957).
8.22. N.S. Mendelsohn, A. L. Dulmage, Some Generalizations of the Problem of Distinct Representatives, *Canad, J. Math.*, Vol. 10, 230—241 (1958).
8.23. W. T. Tutte, The Factorization of Linear Graphs, *J. London Math. Soc.*, Vol. 22, 107—111 (1947).
8.24. I. Anderson, Perfect Matchings of a Graph., *J. Combinatorial Theory* B., Vol. 10, 183—186 (1971).
8.25. L. Lovász, Three Short Proofs in Graph Theory, *J. Combinatorial Theory* B, Vol. 19, 111—113 (1975).
8.26. C. Berge, Graphs and Hypergraphs, North-Holland, Amsterdam, 1973.
8.27. F. T. Boesch (Ed.), Large-Scale Networks: Theory and Design, IEEE Press, New York, 1976.
8.28. S. L. Hakimi, An Algorithm for Construction of Least Vulnerable Communication Networks or the Graph with the Maximum Connectivity, *IEEE Trans. Circuit Theory*, Vol. CT-16, 229—230 (1969).
8.29. F. T. Boesch and R. E. Thomas, On Graphs of Invulnerable Communication Nets, *IEEE Trans. Circuit Theory*, Vol. CT-17, 183—192 (1970).
8.30. A. T. Amin and S. L. Hakimi, Graphs with Given Connectivity and Independence Number or Networks with Given Measures of Vulnerability and Survivability, *IEEE Trans. Circuit Theory*, Vol. CT-20, 2—10 (1973).
8.31. H. Frank and I. T. Frisch, Communication, Transmission, and Transsortation Networks, Addison-Wesley, Reading, Mass., 1971.
8.32. L. Mirsky and H. Perfect, Systems of Representatives, *J. Math. Anal. Applic.*, Vol. 15, 520—568 (1966).

8.33. R. A. Brualdi, Transversal Theory and Graphs, in Studies in Graph Theory, Part II, MAA Press, 1975, pp. 23—88.
8.34. L. Mirsky, Transversal Theory, Academic Press, New York, 1971.
8.35. J. A. Bondy and U.S.R. Murty, Graph Theory with Applications, Macmillan, London, 1976.
8.36. См. [1.5].
9.1. D. König, Graphs and Matrices (in Hungarian), *Mat. Fiz. Lapok*, Vol. 38, 116—118 (1931).
9.2. C. Berge, Graphs and Hypergraphs, North Holland, Amsterdam, 1973.
9.3. T. Gallai, Über extreme Punkt und Kantenmengen, *Ann. Univ. Sci. Budapest, Eötvös Sect. Math.*, Vol. 2, 133—138 (1959).
9.4. V. G. Vizing, On an Estimate of the Chromatic Class of a p-Graph (in Russian), *Diskret. Analiz.*, Vol. 3, 25—30 (1964).
9.5. J. C. Fournier, Colorations des arêtes d'un graphe, *Cahiers du CERO*, Vol. 15, 311—314 (1973).
9.6. J. A. Bondy and U.S.R. Murty, Graph Theory with Applications, Macmillan, London, 1976.
9.7. R. L. Brooks, On Colouring the Nodes of a Network, *Proc. Cambridge Phil. Soc.*, Vol. 37, 194—197 (1941).
9.8. L. S. Melnikov and V. G. Vizing, New Proof of Brooks' Theorem, *J. Combinatorial Theory*, Vol. 7, 289—290 (1969).
9.9. L. Lovász, Three Short Proofs in Graph Theory, *J. Combinatorial Theory B*, Vol. 19, 111—113 (1975).
9.10. P. J. Heawood, Map Colour Theorems, *Quart. J. Math.*, Vol. 24, 332—338 (1890).
9.11. K. I. Appel and W. Haken, Every Planar Map is Four-Colorable, *Bull. Am. Math. Soc.*, Vol. 82, 711—712 (1976).
9.12. O. Ore, The Four-Color Problem, Academic Press, New York, 1967.
9.13. T. L. Saaty, Thirteen Colorful Variations on Guthrie's Four-Color Conjecture, *Am. Math. Monthly*, Vol. 79, 2—43 (1972).
9.14. T. L. Saaty and P. C. Kainen, The Four-Color Problem: Assaults and Conquests, McGraw-Hill, New York, 1977.
9.15. H. Whitney and W. T. Tutte, Kempe Chains and the Four Colour Problem, in Studies in Graph Theory, Part II, MAA Press, 1975, pp. 378—413.
9.16. См. [1.3].
9.17. P. Turán, An Extremal Problem in Graph Theory (in Hungarian), *Mat. Fiz. Lapok*, Vol. 48, 436—452 (1941).
9.18. P. Erdös, On the Graph Theorem of Turán (in Hungarian), *Mat. Lapok*, Vol. 21, 249—251 (1970).
9.19. B. Bollobás, Extremal Graph Theory, Academic Press, New York, 1978.
9.20. H. Frank and I. T. Frisch, Communication, Transmission, and Transportation Networks, Addison-Wesley, Reading, Mass., 1971.
9.21. F. T. Boesch (Ed.), Large-Scale Networks: Theory and Design, IEEE Press, New York, 1976.
9.22. P. Karivaratharajan and K. Thulasiraman, K-Sets of a Graph and Vulnerability of Communication Nets, *Matrix and Tensor Quart.*, Vol. 25, 63—66 (1974), 77—86 (1975).
9.23. P. Karivaratharajan and K. Thulasiraman, An Extremal Problem in Graph Theory and Its Applications, Proc. IEEE Intl. Symp. on Circuits and Systems, Tokyo, Japan, 1979.
9.24. C. L. Liu, Introduction to Combinatorial Mathematics, McGraw-Hill, New York, 1968.
9.25. L. Lovász, On the Shannon Capacity of a Graph, *IEEE Trans. Inform. Theory*, Vol. IT-25, 1—7 (1979).
9.26. R. C. Read, An Introduction to Chromatic Polynomials, *J. Combinatorial Theory*, Vol. 4, 52—71 (1968).

9.27. G. D. Birkhoff, A Determinan Formula for the Number of Ways of Coloring a Map, *Ann. Math.*, Vol. 14, 42—46 (1912).
9.28. W. T. Tutte, On Chromatic Polynomials and the Golden Ratio, *J. Combinatorial Theory*, Vol. 9, 289—296 (1970).
9.29. W. T. Tutte, Chromials in Studies in Graph Theory, Part II, MAA Press, 1975, pp. 361—377.
9.30. D. C. Wood, A Technique for Colouring a Graph Applicable to Large-Scale Timetabling Problems, *Computer J.*, Vol. 12, 317 (1969).
9.31. N. Christofides, Graph Theory: An Algorithmic Approach, Academic Press, New York, 1975. [Имеется перевод: Н. Кристофидес. Теория графов: Алгоритмический подход: — М.: Мир, 1978.]
9.32. E. J. Cockayne and S. T. Hedetniemi, Towards a Theory of Domination in Graphs, *Networks*, Vol. 7, 247—267 (1977).
9.33. E. J. Cockayne, S. T. Hedetniemi, Optimal Domination in Graphs, *IEEE Trans. Circuits and Sys.*, Vol. CAS-22, 41—44 (1975).
9.34. D.J.A. Welsh, M. B. Powell, An Upper Bound for the Chromatic of a Graph and Its Application of Timetabling Problems, *Computer J.*, Vol. 10, 85—87 (1967).
10.1. H. Whitney, On the Abstract Properties of Linear Dependence, *Am. J. Math.*, Vol. 57, 509—533 (1935).
10.2. G. J. Minty, On the Axiomatic Foundations of the Theories of Directed Linear Graphs, Electrical Networks and Network Programming, *J. Math. and Mech.*, Vol. 15, 485—520 (1966).
10.3. W. T. Tutte, Introduction to the Theory of Matroids, American Elsevier, New York, 1971.
10.4. D.J.A. Welsh, Matroid Theory, Academic Press, New York, 1976.
10.5. G. J. Minty, Monotone Networks, *Proc. Roy. Soc.*, *A*, Vol. 257, 194—212 (1960).
10.6. G. J. Minty, Solving Steady-State Non-Linear Networks of 'Monotone' Elements, *IRE Trans. Circuit Theory*, Vol. CT-8, 99—104 (1961).
10.7. J. B. Kruskal, On the Shortest Spanning Subgraph of a Graph and the Travelling Salesman Problem, *Proc. Am. Math. Soc.*, Vol. 7, 48—49 (1956).
10.8. R. Rado, Note on Independence Functions, *Proc. London Math. Soc.*, Vol. 7, 300—320 (1957).
10.9. J. Edmonds, Matroids and the Greedy Algorithm, *Math. Programming*, Vol. 1, 127—136 (1971).
10.10. D. Gale, Optimal Assignments in an Ordered Set: An Application of Matroid Theory, *J. Combinatorial Theory*, Vol. 4, 176—180 (1968).
10.11. D.J.A. Welsh, Kruskal's Theorem for Matroids, *Proc. Cambridge Phil. Soc.*, Vol. 64, 3—4 (1968).
10.12. Rabe von Randow, Introduction to the Theory of Matroids, Springer Lecture Notes in Mathematical Economics, Vol. 109, 1975.
10.13. C. Berge, Graphs and Hypergraphs, North Holland, Amsterdam, 1973.
10.14. См. [1.5].
10.15. R. Rado, A Theorem on Independence Relations, *Quart, J. Math. (Oxford)*, Vol. 13, 83—89 (1942).
10.16. A. Lehman, A Solution of the Shannon Switching Game, *SIAM J. Appl. Math.*, Vol. 12, 687—725 (1964).
10.17. W. T. Tutte, A Homotopy Theorem for Matroids-I and II, *Trans. Am. Math. Soc.*, Vol. 88, 144—174 (1958).
10.18. W. T. Tutte, Lectures on Matroids, *J. Res. Nat. Bur. Stand.*, Vol. 69B, 1—48 (1965).
10.19. W. T. Tutte, Matroids and Graphs, *Trans. Am. Math. Soc.*, Vol. 90, 527—552 (1959).
10.20. W. T. Tutte, Connectivity in Matroids, *Canad. J. Math.*, Vol. 18, 1301—1324 (1966).

10.21. L. Mirsky, Application of the Notion of Independence to Combinatorial Analysis, *J. Combinatorial Theory*, Vol. 2, 327—357 (1967).
10.22. F. Harary and D.J.A. Welsh, Matroids versus Graphs, in The Many Facets of Graph Theory, Springer Lecture Notes, Vol. 110, 1969, pp. 155—170.
10.23. R. J. Wilson, An Introduction to Matroid Theory, *Am. Math. Monthly*, Vol. 80, 500—525 (1973).
10.24. J. Edmonds, Minimum Partition of a Matroid into Independent Subsets, *J. Res. Nat. Bur. Stand.*, Vol. 69B, 67—72 (1965).
10.25. J. Edmonds and D. R. Fulkerson, Transversals and Matroids Partition, *J. Res. Nat. Bur. Stand.*, Vol. 69B, 147—153 (1965).
10.26. L. Mirsky, Tranversal Theory, Academic Press, London, 1971.
10.27. E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.
10.28. R.G. Bland and M. Las Vergnas, Orientability of Matroids, *J. Combinatorial Theory B*, Vol. 24, 94—123 (1978).
10.29. J. Folkman and J. Lawrence, Oriented Matroids, *J. Combinatorial Theory B*, Vol. 25, 199—236 (1978).
10.30. J. Edmonds, Lehman's Switching Game and a Theorem of Tutte and Nash-Williams, *J. Res. Nat. Bur. Stand.*, Vol. 69B, 73—77 (1965).
10.31. J. Bruno and L. Weinberg, A Constructive Graph-Theoretic Solution of the Shannon Switching Game, *IEEE Trans. Circuit Theory*, Vol. CT-17, 74—81 (1970).
10.32. D. E. Knuth, Matroid Partitioning, Stanford University Rep. STAN-CS-73-342, 1—12 (1973).
10.33. J. Edmonds, Matroid Partition, in Lectures in Appl. Math., Vol. 11: Mathematics of Decision Sciences, 1967, pp. 335—346.
10.34. R. J. Duffin, Topology of Series-Parallel Networks, *J. Math. Anal. Appl.*, Vol. 10, 303—318 (1965).
10.35. R. J. Duffin and T. D. Morley, Wang Algebra and Matroids, IEEE Trans. Circuits and Syst., Vol. CAS-25, 755—762 (1978).
10.36. H. Narayanan, Theory of Matroids and Network Analysis, Ph. D. Thesis, Indian Institute of Technology, Bombay, India, 1974.
10.37. J. Bruno and L. Weinberg, Generalized Networks: Networks Embedded on a Matroid, Part 1, *Networks*, Vol. 6, 53—94 (1976).
10.38. J. Bruno and L. Weinberg, Generalized Networks: Networks Embedded on a Matroid, Part 2, *Networks*, Vol. 6, 231—272 (1976).
10.39. L. Weinberg, Matroid, Generalized Networks and Electric Network Synthesis, *J. Combinatorial Theory B*, Vol. 23, 106—126 (1977).
10.40. M. Iri and N. Tomizawa, A Unifying Approach to Fundamental Problems in Network Theory by Means of Matroids, *Electron. Commun. in Japan*, Vol. 58-A, 28—35 (1975).
10.41. A. Recski, On Partitional Matroids with Applications, in *Coll. Math. Soc. J. Bolyai*, Vol. 10: Infinite and Finite Sets, North-Holland-American Elsevier, Amsterdam, 1974, pp. 1169—1179.
10.42. A. Recski, Matroids and Independent State Variables, Proc. 2nd European Conf. Circuit Theory and Design, Genova, 1976.
10.43. B. Petersen, Investigating Solvability and Complexity of Linear Active Networks by Menas of Matroids, *IEEE Trans. Circuits and Syst.*, Vol. CAS-26, 330—342 (1979).
10.44. C. Greene, A Multiple Exchange Property for Bases, *Proc. Am. Math. Soc.*, Vol. 39, 45—50 (1973).
10.45. J. H. Mason, On a Class of Matroids Arising from Paths in Graphs, *Proc. London Math. Soc.*, Vol. 25, 55—74 (1972).
10.46. R. C. Prim, Shortest Connection Networks and Some Generalizations, *Bell Sys. Tech. J.*, Vol. 36, 1389—1402 (1957).
11.1. B. D. H. Tellegen, A General Network Theorem with Applications, *Philips Res. Rept.*, Vol. 7, 259—269 (1952).

11.2. P. Penfield, Jr., R. Spence, and S. Duinker, A Generalized Form of Tellegen's Theorem, *IEEE Trans. Circuit Theory*, Vol. CT-17, 302—305 (1970).
11.3. P. Penfield, Jr., R. Spence, and S. Duinker, Tellegen's Theorem and Electrical Networks, M.I.T. Press, Cambridge, Mass., 1970.
11.4. J. L. Bordewijk, Inter-Reciprocity Applied to Electrical Networks, *Appl. Sci. Res.*, Vol. B6, 1—74 (1956).
11.5. S. W. Director and R. A. Rohrer, Automated Network Design—The Frequency Domain Case, *IEEE Trans. Circuit Theory*, Vol. CT-16, 330—337 (1969).
11.6. G. Kishi and T. Kida, Edge-Port Conservation in Networks, *IEEE Trans. Circuit Theory*, Vol. CT-15, 274—276 (1968).
11.7. G. Kishi and Y. Kajitani, Maximally Distant Trees and Principal Partition of a Linear Graph, *IEEE Trans. Circuit Theory*, Vol. CT-16, 323—330 (1969).
11.8. T. Ohtsuki, Y. Ishizaki, and H. Watanabe, Topological Degrees of Freedom and Mixed Analysis of Electrical Networks, *IEEE Trans. Circuit Theory*, Vol. CT-17, 491—499 (1970).
11.9. P. M. Lin, An Improved Algorithm for Principal Partition of Graphs, Proc. IEEE Intl. Symp. Circuits and Systems, 1976, pp. 145—148.
11.10. J. Bruno and L. Weinberg, The Principal Minors of a Matroid, *Linear Algebra and Its Appl.*, Vol. 4, 17—54 (1971).
11.11. N. Balabanian and T. A. Bickart, Electrical Network Theory, Wiley, New York, 1969.
11.12. E. S. Kuh and R. A. Rohrer, The State Variable Approach to Network Analysis, *Proc. IEEE*, Vol. 53, 672—686 (1965).
11.13. D. H. Wolaver, Proof in Graph Theory of the 'No-Gain' Property of Resistor Networks, *IEEE Trans. Circuit Theory*, Vol. CT-17, 436—437 (1970).
11.14. A. Talbot, Some Fundamental Properties of Networks without Mutual Inductance, *Proc. IEE (London)*, Vol. 102, 168—175 (1955).
11.15. R. J. Schwartz, A Note on the Transfer Ratio of Resistive Networks with Positive Elements, *Proc. IRE*, Vol. 43, 1670 (1955).
11.16. См. [2.1].
11.17. W. H. Kim and R. T. Chien, Topological Analysis and Synthesis of Communication Networks, Columbia Univ. Press, New York, 1962.
11.18. W. K. Chen, Applied Graph Theory, North-Holland, Amsterdam, 1971.
11.19. W. Mayeda, Graph Theory, Wiley-Interscience, New York, 1970.
11.20. G. Kron, A Set of Principles to Interconnect the Solution of Physical Systems, *J. Appl. Phys., Vol.* 24, 965—980 (1953).
11.21. G. Kron, Diakoptics: The Piecewise Solution of Large Scale Systems, MacDonald, London, 1963.
11.22. H. H. Happ, Diakoptics and Networks, Academic Press, New York, 1971.
11.23. H. H. Happ, Diakoptics—The Solution of System Problems by Tearing, *Proc. IEEE*, Vol. 62, 930—940 (1974).
11.24. F. H. Branin, The Relation Between Kron's Method and the Classical Methods of Network Analysis, *Matrix and Tensor Quart*, Vol. 12, 69—115 (1962).
11.25. F. H. Branin, A Sparse Matrix Modification of Kron's Method of Piecewise Analysis, Proc. IEEE Intl. Symp. Circuits and Systems, 1975, pp. 21—23.
11.26. R. Onodera, Diakoptics and Codiakoptics of Electric Network, *RAAG Memoirs*, Vol. 2, 369—388 (1958).
11.27. J. Ruth, An Application of Algebraic Topology: Kron's Method of Tearing, *Quart. Appl. Math.*, Vol. 17, 1—24 (1959).
11.28. S. Amari, Topological Foundations of Kron's Tearing of Electrical Networks, *RAAG Memoirs*, Vol. 3, 322—350 (1962).
11.29. K. U. Wang and T. Chao, An Algebraic Theory of Network Topology, Proc. IEEE Intl. Symp. Circuits and Systems, 1974, pp. 324—328.
11.30. F. F. Wu, Solutions of Large—Scale Networks by Tearing, *IEEE Trans. Circuits and Sys.*, Vol. CAS-23, 706—713 (1976).

11.31. L. O. Chua and L. K. Chen, Nonlinear Diakoptics, Proc. IEEE Intl. Symp. Circuits and Systems, 1975, pp. 373—376.
11.32. L. O. Chua and L. K. Chen, Diakoptic and Generalized Hybrid Analysis, *IEEE Trans. Circuits and Sys.*, Vol. CAS-23, 694—705 (1976).
11.33. L. O. Chua and L. K. Chen, On Optimally Sparse Cycle and Coboundary Basis for a Linear Graph, *IEEE Trans. Circuit Theory*, Vol. CT-20, 495—503, (1973).
11.34. L. O. Chua and D. N. Green, Graph-Theoretic Properties of Dynamic Nonlinear Networks, *IEEE Trans. Circuits and Sys.*, Vol. CAS-23, 292—312 (1976).
11.35. T. R. Bashkow, The *A*-Matrix—A New Network Description, *IRE Trans. Circuit Theory*, Vol. CT-4, 117—119 (1957).
11.36. P. R. Bryant, The Explicit Form of Bashkow's *A*-Matrix, *IRE Trans. Circuit Theory*, Vol. CT-9, 303—306 (1962).
11.37. E. J. Purslow. Solvability and Analysis of Linear Active Networks by Use of the State Equations, *IEEE Trans. Circuit Theory*, Vol. CT-17, 469—475 (1970).
11.38. O. Tosun and A. Dervisoglu, Formulation of State Equations in Active RLC Networks, *IEEE Trans. Circuits and Sys.*, Vol. CAS-21, 36—38 (1974).
11.39. S. K. Mark and M.N.S. Swamy, The Generalized Tree for State Variables in Linear Networks, *Int. J. Circuit Theory and Appl.*, Vol. 4, 87—92 (1976).
11.40. W. K. Chen and F. N. Chan, On the Unique Solvability of Linear Active Networks, *IEEE Trans. Circuits and Sys.*, Vol. CAS-21, 26—35 (1974).
11.41. J. Bruno and L. Weinberg, Generalized Networks Embedded on a Matroid—Part I, *Networks*, Vol. 6, 53—94 (1976).
11.42. J. Bruno and L. Weinberg, Generalized Networks: Networks Embedded on a Matroid—Part II, *Networks*, Vol. 6, 231—272 (1976).
11.43. L. Weinberg, Matroid, Generalized Networks, and Electric Network Synthesis, *J. Combinat. Theory B*, Vol. 23, 106—126 (1977).
11.44. R. J. Duffin and T. D. Morley, Wang Algebra and Matroids, *IEEE Trans. Circuits and Sys.*, Vol. CAS-25, 755—762 (1978).
11.45. R. J. Duffin, An Analysis of the Wang Algebra of Networks, *Trans. Am. Math. Soc.*, Vol. 93, 114—131 (1959).
11.46. K. T. Wang, On a New Method of Analysis of Electrical Networks, Memoirs 2, Nat. Res. Inst. Eng. Academia Sinica (1934).
11.47. B. Petersen, Investigating Solvability and Complexity of Linear Active Networks by means of Matroids, *IEEE Trans. Circuits and Sys.*, Vol. CAS-26, 330—342 (1979).
11.48. M. Iri and N. Tomizawa, A Unifying Approach to Fundamental Problems in Network Theory by means of Matroids, *Electron. and Commun. in Japan.*, Vol. 58-A, 28-35 (1975).
11.49. R. J. Duffin, Electrical Network Models, in Studies in Graph Theory, Part II, The Mathematical Association of America, 1975, pp. 94—138.
12.1. I. Cederbaum, Applications of Matrix Algebra to Network Theory, *IRE Trans. Circuit Theory* (special supplement), Vol. CT-6, 127—137 (1959).
12.2. L. Weinberg, Network Analysis and Synthesis, McGraw-Hill, New York, 1962.
12.3. E. A. Guillemin, On the Analysis and Synthesis of Single-Element Kind Networks, *IRE Trans. Circuit Theory*, Vol. CT-7, 303—312 (1960).
12.4. V. V. Bapeswara Rao, The Tree-Path Matrix of a Network and Its Applications., Ph. D. Thesis, Dept. of Electric Engineering, Indian Institute of Technology, Madras, India, 1970.
12.5. F. T. Boesch and D. C. Youla, Synthesis of $(n+1)$-Node Resistor n-Ports, *IEEE Trans. Circuit Theory*, Vol. CT-12, 515—520 (1965).
12.6. G. Biorci and P. P. Civalleri, On the Synthesis of Resistive N-Port Networks, *IRE Trans. Circuit Theory*, Vol. CT-8, 22—28 (1961).

12.7. C. C. Halkias, I. Cederbaum, and W. H. Kim, Synthesis of Resistive n-Port with $(n+1)$ Nodes, *IRE Trans. Circuit Theory*, Vol. CT-9, 69—73 (1962)
12.8. E. A. Guillemin, On the Realization of an n-th Order G-Matrix, *IRE Trans. Circuit Theory*, Vol. CT-8, 318—323 (1961).
12.9. K. R. Swaminathan and I. T. Frisch, Necessary Conditions for the Realizability of n-Port Resistive Networks with more than $(n+1)$ Nodes, *IEEE Trans. Circuit Theory*, Vol. CT-12, 520—527 (1965).
12.10. G. Biorci and P. P. Civalleri, Analysis of Resistive N-Port Networks Based on $(n+2)$ Nodes, in Aspects of Network and System Theory (Eds. R. E. Kalman and N. De Claris), Holt, Rinehart, Winston, New York, 1971.
12.11. P. Subbarami Reddy and K. Thulasiraman, Synthesis of $(n+2)$-Node Resistive n-Port Networks, *IEEE Trans Circuit Theory*, Vol. CT-19, 20—25 (1972).
12.12. D. P. Brown, N-Port Synthesis of N-order Positive Entry Resistance Matrices, *J. Franklin Inst.*, Vol. 284, 26—38 (1967).
12.13. C. Eswaran and V. G. K. Murti, Realization of Positive Entry Resistance Matrices, AEÜ, Vol. 29, 212—216 (1975).
12.14. C. Eswaran and V. G. K. Murti, On a Relationship between Terminal Capacity and Impedance Matrices, *IEEE Trans. Circuits and Sys.*, Vol. CAS-21, 732—734 (1974).
12.15. I. T. Frish and W. H. Kim, n-Port Resistive Networks and Communication Nets, *IRE Trans. Circuit Theory*, Vol. CT-8, 493—496 (1961).
12.16. I. Cederbaum, On Equivalence of Resistive N-Port Networks, *IEEE Trans. Circuit Theory*, Vol. CT-12, 338—344 (1965).
12.17. P. Subbarami Reddy, V. G. K. Murti, and K. Thulasiraman, Realization of Modified Cutset Matrix and Applications, *IEEE Trans. Circuit Theory*, Vol. CT-17, 475—486 (1970).
12.18. A. Lempel and I. Cederbaum, Parallel Interconnection of n-port Networks, *IEEE Trans. Circuit Theory*, Vol. CT-14, 274—279 (1967).
12.19. A. Lempel and I. Cederbaum, Terminal Configurations of n-Port Networks, *IEEE Trans. Circuit Theory*, Vol. CT-15, 51—53 (1968).
12.20. S. Prigozy and L. Weinberg, Realization of Fourth-Order Singular and Quasi-Singular Resistance and Conductance Matrices, *IEEE Trans. Circuits and Sys.*, Vol. CAS-23, 245—253 (1976).
12.21. M. G. Govindarajulu Naidu, P. Subbarami Reddy, and K. Thulasiraman, $(n+2)$-Node Resistive n-Port Realizability of Y-Matrices, *IEEE Trans. Circuits and Sys.*, Vol. CAS-23, 254—261 (1976).
12.22. W. T. Tutte, An Algorithm for Determining whether a Given Binary Matroid is Graphic, *Proc. Am. Math. Soc.*, Vol. 11, 905—917 (1960).
12.23. W. T. Tutte, From Matrices to Graphs, *Can. J. Math.*, Vol. 56, 108—127 (1964).
12.24. W. Mayeda, Necessary and Sufficient Conditions for the Realizability of Cutset and Circuit Matrices, *IRE Trans. Circuit Theory*, Vol. CT-7, 79—81 (1960).
12.25. W. Mayeda, A Proof of Tutte's Realizability Condition, *IEEE Trans. Circuit Theory*, Vol. CT-17, 506—511 (1970).
12.26. W. H. Kim and R. T. Chien, Topological Analysis and Synthesis of Communication Networks, Columbia Univ. Press, New York, 1962.
12.27. M.N.S. Swamy and K. Thulasiraman, Realization of the A-Matrix of RLC Networks, *IEEE Trans. Circuit Theory*, Vol. CT-19, 515—518 (1972).
12.28. L. O. Chua and D. N. Green, Graph-Theoretic Properties of Dynamic Nonlinear Networks, *IEEE Trans. Circuits and Sys.*, Vol. CAS-23, 292—302 (1976).
12.29. K. Thulasiraman and V. G. K. Murti, Pseudo-Series Combination of n-Port Networks, Proc. IEEE, Vol. 56, 1143—1144 (1968).
13.1. A. C. Aitken, Determinants and Matrices, Interscience, New York, 1956.

13.2. N. Balabanian and T. A. Bickart, Electrical Network Theory, Wiley, New York, 1969.
13.3. S. K. Mitra, Analysis and Synthesis of Linear Active Networks, Wiley, New York, 1969.
13.4. G. E. Sharpe and B. Spain, On the Solution of Networks by means of the Equi-Cofactor Matrix, *IRE Trans. Circuit Theory*, Vol. CT-7, 230—239 (1960).
13.5. J. L. Bordewijk, Inter-Reciprocity Applied to Electrical Networks, *Appl. Csi, Res.*, Vol. B6, 1—74 (1956).
13.6. S. W. Director and R. A. Rohrer, Automated Network Design—The Frequency Domain Case, *IEEE Trans. Circuit Theory*, Vol. CT-16, 330—337 (1969).
13.7. B. B. Bhattacharyya and M. N. S. Swamy, Network Transposition and Its Application in Synthesis, *IEEE Trans. Circuit Theory*, Vol. CT-18, 394—397 (1971).
13.8. M.N.S. Swamy, C. Bhushan, and B. B. Bhattacharyya, Generalized Duals, Generalized Inverses, and Their Applications, *Radio and Electron. Eng.*, Vol. 44, 95—97 (1974).
13.9. M.N.S. Swamy, C. Bhushan, and B. B. Bhattacharyya, Generalized Dual Transposition and Its Applications, *J. Franklin Inst.*, Vol. 301, 465—474 (1976).
13.10. K. Dagget and J. Vlach, Sensitivity-Compensated Active Networks, *IEEE Trans. Circuit Theory*, Vol. CT-16, 416—422 (1969).
13.11. W. K. Chen, Applied Graph Theory, North-Holland, Amsterdam, 1971.
13.12. S. Seshu and M. B. Reed, Linear Graphs and Electrical Networks, Addison-Wesley, Reading, Mass., 1961.
13.13. S. P. Chan, Introductory Topological Analysis of Electrical Networks, Holt, Rinehart and Winston, New York, 1969.
13.14. W. Mayeda, Graph Theory, Wiley-Interscience, New York, 1972.
13.15. J. Numata and M. Iri, Mixed-Type Topological Formulas for General Linear Networks, *IEEE Trans. Circuit Theory*, Vol. CT-20, 488—494 (1973).
13.16. P. Penfield, Jr., R. Spence, and S. Duinker, Tellegen's Theorem and Electrical Networks, M. I. T. Press, Cambridge, Mass., 1970.
13.17. P. Penfield, Jr., R. Spence, and S. Duinker, A Generalized Form of Tellegen's Theorem, *IEEE Trans. Circuit Theory*, Vol. CT-17, 302—305 (1970).
13.18. C. A. Desoer, On the Description of Adjoint Networks, IEEE Trans. Circuits and Sys., CAS-22, 585—587 (1975), Erratum, *IEEE Trans. Circuits and Sys.*, CAS-23, 58 (1976).
13.19. S. W. Director, R. A. Rohrer, On the Design of Resistance n-Port Networks by Digital Computer, *IEEE Trans. Circuit Theory*, CT-16, 337—346 (1969).
13.20. M.N.S. Swamy, C. Bhushan, and K. Thulasiraman, Bounds on the Sum of Element Sensitivity Magnitudes for Network Functions, *IEEE Trans. Circuit Theory*, CT-19, 502—504 (1972).
13.21. M.N.S. Swamy, C. Bhushan, and K. Thulasiraman, Sensitivity, IEEE Trans, *Circuit Theory*, CT-20, 21—24 (1973).
14.1. D. Gries, Compiler Construction fo Digital Computers, Wiley, NewYork, 1971.
14.2. S. Warshall, A Theorem on Boolean Matrices, J. *ACM*, Vol. 9, 11—12 (1962).
14.3. H. S. Warren, A Modification of Warshall's Algorithm for the Transitive Closure of Binary Relations, *Comm. ACM*, Vol. 18, 218—220 (1975).
14.4. Арлазаров В. Л., Диниц Е. А., Кронрод М. А., Фараджев И. А. Экономическое построение транзитивного замыкания ориентированного графа, ДАН, 1970, т. 11, с. 1209—1210.
14.5. V. Strassen, Gaussian Elimination is Not Optimal, *Numerische Math.*, Vol. 13, 354—356 (1969).

14.6. M. J. Fischer, A. R. Meyer, Boolean Matrix Multiplication and Transitive Closure, Conf. Record. IEEE 12th Annual Symp. on Switching and Automata Theory, 1971, pp. 129—131.
14.7. M. E. Furman, Application of a Method of Fast Multiplication of Matrices in the Problem of Finding the Transitive Clolsure of a Graph, *Soviet Math.*, Dokl., Vol. 11, 1252 (1970).
14.8. I. Munro, Efficient Determination of the Transitive Closure of a Directed Graph, *Information Processing Lett.*, Vol. 1, 56—58 (1971).
14.9. P. E. O'Neil, E. J. O'Neil, A Fast Expected Time Algorithm for Boolean Matrix Multiplication and Transitive Closure, *Inform. and Control*, Vol. 22, 132—138 (1973).
14.10. J. Eve, R. Kurki-Suonio, On Computing the Transitive Closure of a Relation, *Acta Informatika*, Vol. 8, 303—314 (1977).
14.11. P. Purdom, A Transitive Closure Algorithm, *BIT*, Vol. 10, 76—94 (1970).
14.12. C. P. Schnorr, An Algorithm for Transitive Closure with Linear Expected Time, *SIAM J. Computing*, Vol. 7, 127—133 (1978).
14.13. M. M. Syslo and J. Dzikiewicz, Computational Experience with Some Transitive Closure Algorithms, *Computing*, Vol. 15, 33—39 (1975).
14.14. A. Pnueli, A. Lempel, and S. Even, Transitive Orientation of Graphs and Identification of Permutation Graphs, *Canad. J. Math.*, Vol. 23, 160—175 (1971).
14.15. P. C. Gilmore and A. J. Hoffman, A Characterization of Comparability Graphs and of Interval Graphs, *Canad. J. Math.*, Vol. 16, 539—548 (1964).
14.16. S. Even, A. Pnueli, and A. Lempel, Permutation Graphs and Transitive Graphs, *J. ACM*, Vol. 19, 400—410 (1972).
14.17. S. Even. Algorithmic Combinatorics, Macmillan, New York, 1973.
14.18. C. L. Liu, Introduction to Combinatorial Mathematics, McGraw-Hill, New York, 1968.
14.19. R. E. Tarjan, Depth-First Search and Linear Graph Algorithms, *SIAM J. Computing*, Vol. 1, 146—160 (1972).
14.20. J. Hopcroft and R. Tarjan, Efficient Algorithms for Graph Manipulation, *Comm. ACM*, Vol. 16, 372—378 (1973).
14.21. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, On Finding the Least Common Ancestors in Trees, *SIAM J. Computing*, Vol. 5, 115—132 (1976).
14.22. J. Cocke, Global Common Subexpression Eliminaiton, *SIGPLAN Notices*, Vol. 5, 20—24 (1970).
14.23. J. D. Ullman, East Algorithms for the Elimination of Common Subexpressions *Acta Informatica*, Vol. 2, 191—213 (1973).
14.24. K. Kennedy, A Global Flow Analysis Algorithm, *Int. J. Computer Math.*, Vol. 3, 5—16 (1971).
14.25. M. Schaefer, A Mathematical Theory of Global Program Optimization, Prentice-Hall, Englewood C. Cliffs, N. J., 1973.
14.26. A. V. Aho and J. D. Ullman, The Theory of Parsing, Translation and Compiling, Vol. II—Compiling, Prentice-Hall, Englewood, Cliffs, N.J. 1973. [Имеется перевод: А. Ахо, Дж. Ульман. Теория синтаксического анализа, перевода и компиляции: — М.: Мир, 1978.]
14.27. F. E. Allen, Control Flow Analysis, *SIGPLAN Notices*, Vol. 5, 1—19 (1970).
14.28. F. E. Allen, Program Optimization, Annual Review in Automatic Programming, Vol. 5., Pergamon, New York, 1969.
14.29. M. S. Hecht, Flow Analysis of Computer Programs, Elsevier, New York, 1977.
14.30. M. S. Hecht and J. D. Ullman, Flow Graph Reductibility, *SIAM J. Computing*, Vol. 1, 188—202 (1972).
14.31. J. Cocke, R. E. Miller, Some Analysis Techniques for Optimizing Computer Programs, Proc. 2nd Int. Conf. on System Sciences, Honolulu, Hawaii, 1969.

14.32. J. E. Hopcroft, J. D. Ullman, An *n*log *n* Algorithm for Detecting Reducible Graphs, Proc. 6th Annual Princeton Conf. on Information Sciences and Systems, Princeton, J. N. 1972, pp. 119—122.
14.33. R. E. Tarjan, Testing Flow Graph Reductibility, Proc. 5th Annual ACM Symp. on Theory of Computing, 1973, pp. 96—107.
14.34. R. E. Tarjan, Testing Flow Graph Reductibility, *J. Comput. Sys. Sci.*, Vol. 9, 355—365 (1974).
14.35. M. S. Hecht and J. D. Ullman, Characterizations of Reducible Flow Graphs, *J. ACM*, Vol. 21, 367—375 (1974).
14.36. J. M. Adams, J. M. Phelan, and R. H. Stark, A Note on the Hecht-Ullman Characterization of Non-Reducible Flow Graphs, *SIAM J. Computing*, Vol. 3, 222—223 (1974).
14.37. M. Fischer, Efficiency of Equivalence Algorithms, in Complexity of Computer Computations (R. E. Miller and J. W. Thatcher, Eds.), Plenum Press, New York, 1972, pp. 153—168.
14.38. J. E. Hopcroft and J. D. Ullman, Set Merging Algorithms, *SIAM J. Computing*, Vol. 2, 294—303 (1973).
14.39. R. E. Tarjan, On the Efficiency of a Good but Not Linear Set Union Algorithm, *J. ACM*, Vol. 22, 215—225 (1975).
14.40. См. [5.14].
14.41. E. Horowitz and S. Sahni, Fundamentals of Data Structures, Computer Science Press, Potomac, Md., 1976.
14.42. T. Lengauer and R. E. Tarjan, A Fast Algorithm for Finding Dominators in a Flow Graph, *Trans. on Prog. Lang. and Sys.*, Vol. 1, 121—141 (1979).
14.43. R. E. Tarjan, Finding Dominators in Directed Graphs, *SIAM J. Computing*, Vol. 3, 62—89 (1974).
14.44. R. E. Tarjan, Applications of Path Compression on Balanced Trees, *J. ACM*, Vol. 26, 690—715 (1979).
14.45. A. V. Aho and J. D. Ullman, Principles of Compiler Design, Addison-Wesley, Reading, Mass., 1977.
14.46. P. W. Purdom and E. F. Moore, Algorithm 430: Immediate Predominators in a Directed Graph, *Comm. ACM*, Vol. 15, 777—778 (1972).
14.47. H. Frank and I. T. Frisch, Communication, Transmission and Transportation Networks, Addison-Wesley, Reading, Mass., 1971.
14.48. См. [5.12].
14.49. См. [9.31].
14.50. E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.
14.51. См. [5.15].
14.52. E. Minieka, Optimization Algorithms for Networks and Graphs, Marchel Dekker, New York, 1978. [Имеется перевод: Э. Майника. Алгоритмы оптимизации на сетях и графах.— М.: Мир, 1981.]
14.53. S. Even, Graph Algorithms, Computer Science Press, Potomac, Md., 1979.
14.54. S. A. Cook, The Complexity of Theorem Proving Procedures, Proc. 3rd ACM Symp. on Theory of Computing 1971, pp. 151—158. [Имеется перевод: С. А. Кук. Сложность процедур доказательства теорем: Кибернетический сборник. Вып. 18: — М.: Мир, 1981.]
14.55. R. M. Karp, Reducibility among Combinatorial Problems, in Complexity of Computer Computations (R. E. Miller and J. W. Thatcher, Eds.) Plenum Press, New York, 1972, pp. 85—104. [Имеется перевод: Р. М. Карп. Сводимость комбинаторных задач: Кибернетический сборник. Вып. 18: — М.: Мир, 1981.]
14.56. M. R. Garey and D. S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, Freeman, San Francisco, Ca., 1979.
14.57. E. Horowitz and S. Sahni, Fundamentals of Computer Algorithms, Computer Science Press, Potomac, Md., 1978.

14.58. A. V. Aho, M. R. Garey, and J. D. Ullman, The Transitive Reduction of a Directed Graph, *SIAM J. Computing*, Vol. 1, 131—137 (1972).
14.59. H. Prüfer, Neuer Beweis eines Satzes über Permutationen, *Arch. Math. Phys.*, Vol. 27, 742—744 (1918).
15.1. L. R. Ford and D. R. Fullkerson, Flows in Networks, Princeton Univ. Press, Princeton, N. J. 1962. [Имеется перевод: Л. Р. Форд, Д. Фалкерсон. Потоки в сетях: — М.: Мир, 1966.]
15.2. E. W. Dijkstra, A Note on Two Problems in Connexion with Graphs, *Numerische Math.*, Vol. 1, 269—271 (1959).
15.3. D. B. Johnson, A Note on Dijkstra's Shortest Path Algorithm, *J. ACM*, Vol. 20, 385—388 (1973).
15.4. J. Edmonds and R. M. Karp, Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems, *J. ACM*, Vol. 19, 248—264 (1972).
15.5. S. E. Dreyfus, An Appraisal of Some Shortest-Path Algorithms, *Operations Research*, Vol. 17, 395—412 (1969).
15.6. T. C. Hu, A Decomposition Algorithm for Shortest Paths in a Network, *Operations Research*, Vol. 16, 91—102 (1968).
15.7. H. Frank and I. T. Frisch, Communication, Transmission and Transportation Networks, Addison-Wesley, Reading, Mass., 1971.
15.8. См. [9.31].
15.9. E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.
15.10. P. M. Spira and A. Pan, On Finding and Updating Spanning Trees and Shortest Paths, *SIAM J. Comput.*, Vol. 4, 375—380 (1975).
15.11. D. B. Johnson, Efficient Algorithms for Shortest Paths in Sparse Networks, *J. ACM*, Vol. 24, 1—13 (1977).
15.12. R. A. Wagner, A Shortest Path Algorithm for Edge-Sparse Graphs, *J. ACM*, Vol. 23, 50—57 (1976).
15.13. R. W. Floyd, Algorithm 97: Shortest Path, *Comm. ACM*, Vol. 5, 345 (1962).
15.14. Y. Tabourier, All Shortest Distances in a Graph: An Improvement to Dantzig's Inductive Algorithm, *Discrete Math.*, Vol. 4, 83—87 (1973).
15.15. J. Y. Yen, Finding the Lengths of All Shortest Paths in N-Node, Non-Negative Distance Complete Networks Using $N^3/2$ Additions and N^3 Comparisons, *J. ACM*, Vol. 19, 423—424 (1972).
15.16. T. A. Williams and G. P. White, A Note on Yen's Algorithm for Finding the Length of All Shortest Paths in N-Node Non-Negative Distance Networks, *J. ACM*, Vol. 20, 389—390 (1973).
15.17. A. R. Pierce, Bibliography on Algorithms for Shortest Path, Shortest Spanning Tree and Related Circuit Routing Problems (1956—1974), *Networks*, Vol. 5, 129—149 (1975).
15.18. D. A. Huffman, A Method for the Construction of Minimum Redundancy Codes, *Proc. IRE*, Vol. 40, 1098—1101 (1952).
15.19. E. N. Gilbert and E. F. Moore, Variable-Length Binary Encodings, *Bell Sys. Tech. J.*, Vol. 38, 933—968 (1959).
15.20. D. E. Knuth, Optimum Binary Search Trees, *Acta Informatica*, Vol. 1, 14—25 (1971).
15.21. A. Itai, Optimal Alphabetic Trees, *SIAM J. Comput.*, Vol. 5, 9—18 (1976).
15.22. T. C. Hu and A. C. Tucker, Optimal Computer Search Trees and Variable-Length Alphabetical Codes, *SIAM J. Appl. Math.*, Vol. 21, 514—532 (1971).
15.23. T. C. Hu, A New Proof of the T-C Algorithm, *SIAM J. Appl. Math.*, Vol. 25, 83—94 (1973).
15.24. См. [5.12].
15.25. A. M. Garsia and M. L. Wachs, A New Algorithm for Minimum Cost Binary Trees, *SIAM J. Comput.*, Vol. 6, 622—642 (1977).
15.26. См. [5.15].

15.27. M. Miyakawa, T. Yuba, Y. Sugito, and M. Hoshi, Optimum Sequence Trees, *SIAM J. Comput.*, Vol. 6, 201—234 (1977).
15.28. J. Edmonds, Paths, Trees and Flowers, *Canad. J. Math.*, Vol. 17, 449—467 (1965).
15.29. H. N. Gabow, An Efficient Implementation of Edmonds' Algorithm for Maximum Matching on Graphs, *J. ACM*, Vol. 23, 221—234 (1976).
15.30. M. I. Balinski, Labelling to Obtain a Maximum Matching, in Combinatorial Mathematics and Its Applications (R. C. Bose and T. A. Dowling, Eds.), Univ. North Carolina Press Chappel Hill, N.C., 1967, pp. 585—602.
15.31. D. Witzgall and C. T. Zahn, Jr., Modification of Edmonds' Algorithm for Maximum Matching of Graphs, *J. Res. Nat. Bur. Std.*, Vol. 69B, 91—98 (1965).
15.32. T. Kameda and I. Munro, $AO(|V| \cdot |E|)$ Algorithm for Maximum Matching of Graphs, *Computing*, Vol. 12, 91—98 (1974).
15.33. J. Edmonds, Maximum Matching and a Polyhedron with 0,1 Vertices, *J. Res. Nat. Bur. Std.*, Vol. 69B, 125—130 (1965).
15.34. H. Gabow, An Efficient Implementation of Edmonds' Maximum Matching Algorithm, Tech. Rep. 31, Stanford Univ. Comp. Science Dept., 1972.
15.35. J. A. Bondy and U.S.R. Murty, Graph Theory with Applications, Macmillan, London, 1976.
15.36. J. E. Hopcroft and R. M. Karp, An $n^{5/2}$ Algorithm for Maximum Matching in Bipartite Graphs, *SIAM J. Comput.*, Vol. 2, 225—231 (1973).
15.37. S. Even and R. E. Tarjan, Network Flow and Testing Graph Connectivity, *SIAM J. Comput.*, Vol. 4, 507—518 (1975).
15.38. S. Even and O. Kariv, An $O(n^{5/2})$ Algorithm for Maximum Matching in General Graphs, Proc. 16th Annual Symp. on Foundations of Comp. Science, IEEE, 1975, pp. 100—112.
15.39. A. Itai, M. Rodeh, and S. L. Tanimoto, Some Matching Problems for Bipartite Graphs, *J. ACM*, *Vol.* 25, 517—525 (1978).
15.40. H. W. Kuhn, The Hungarian Method for the Assignment Problem, *Naval Res. Logist. Quart.*, Vol. 2, 83—97 (1955).
15.41. J. Munkres, Algorithms for the Assignment and Transportation Problems, *J. SIAM*, Vol. 5, 32—38 (1957).
15.42. N. Megido, A. Tamir, An $O(N \log N)$ Algorithm for a Class of Matching Problems, *SIAM J. Comput.*, Vol. 7, 154—157 (1978).
15.43. S. Even, A. Itai, A. Shamir, On the Complexity of Time-Table and Multicommodity Flow Problems, *SIAM J. Comput.*, Vol. 5, 691—703 (1976).
15.44. L. R. Ford, D. R. Fulkerson, Maximal Flow through a Network, *Canad. J. Math.*, Vol. 8, 399—404 (1956).
15.45. P. Elias, A. Feinstein, C. E. Shannon, A Note on the Maximum Flow Through a Network, *IRE Trans. Inform. Theory*, IT-2, 117—119 (1956).
15.46. C. Berge, Graphs and Hypergraphs, North-Holland, Amsterdam, 1973.
15.47. N. Zadeh, Theoretical Efficiency of the Edmonds-Karp Algorithm for Computing Maximal Flows, *J. ACM*, Vol. 19, 184—192 (1972).
15.48. Диниц Е. А. Алгоритм решения задачи о максимальном потоке в сети с оценкой мощности, *ДАН*, 1970, т. 11, с. 1277—1280.
15.49. Карзанов А. В. Определение максимального потока в сети методом предикатов, *ДАН*, 1974, т. 15, с. 434—437.
15.50. V. M. Malhotra, M. Pramodh Kumar, and S. N. Maheswari, An $O(V^3)$ Algorithm for Maximum Flows in Networks, *Information Processing Lett.*, 7, 277—278 (1978).
15.51. S. Even, Graph Algorithms, Computer Press, Potomac, Md. 1979.
15.52. C. Berge, A. Ghouila-Houri, Programming, Games and Transportation Networks, Wiley, New York, 1962.
15.53. T. C. Hu, Integer Programming and Network Flows, Addison-Wesley, Reading, Mass., 1969.

15.54. J. Edmonds, Optimum Branchings, *J. Res. Nat. Bur. Std.*, Vol. 71B, 233—240 (1967).
15.55. R. M. Karp, A Simple Derivation of Edmonds' Algorithms for Optimum Branchings, *Networks*, Vol. 1, 265—272 (1972).
15.56. R. E. Tarjan, Finding Optimum Branchings, *Networks*, 7, 25—35 (1977).
15.57. Y. Chu, T. Liu, On the Shortest Arborescencef of a Directed Graph, *Scientia Sinica (Peking)*, Vol. 4, 1396—1400 (1965); *Math. Rev.*, Vol. 33, 1245 (G. W. Walkup).
15.58. F. C. Bock, An Algorithm to Construct a Minimum Directed Spanning Tree in a Directed Network, in Developments in Operations Research, (B. Avi-Itzak, Ed.), Gordon and Breach, New York, 1971, pp. 29—44.
15.59. D. G. Corneil and C. C. Gotlieb, An Efficient Algorithm for Graph Isomorphism *J. ACM*, Vol. 17, 51—64 (1970).
15.60. L. Weinberg, A Simple and Efficient Algorithm for Determining Isomorphism of Planar Triply Connected Graphs, *IEEE Trans. Circuit Theory*, Vol. CT-13, 142—148 (1966).
15.61. J. E. Hopcroft and R. E. Tarjan, A V log V Algorithm for Isomorphism of Triconnected Planar Graphs, *J. Comput. Syst. Sci.*, Vol. 7, 323—331 (1973).
15.62. J. E. Hopcroft and J. K. Wong, Linear Time Algorithm for Isomorphism on Planar Graphs, Proc. 6th Annual ACM Symp. on Theory of Computing, 1974, pp. 172—184.
15.63. J. E. Hopcroft and R. E. Tarjan, Efficient Planarity Testing, *J. ACM*, Vol. 21, 549—568 (1974).
15.64. N. Deo, Note on Hopcroft and Tarjan's Planarity Algorithm, *J. ACM*, Vol. 23, 74—75 (1976).
15.65. J. E. Hopcroft and R. E. Tarjan, Dividing a Graph into Triconnected Components, *SIAM J. Comput.*, Vol. 2, 135—158 (1973).
15.66. D. J. Kleitman, Methods for Investigating the Connectivity of Large Graphs, *IEEE Trans. Circuit Theory*, Vol. CT-16, 232—233 (1969).
15.67. S. Even, An Algorithm for Determining whether the Connectivity of a Graph is at Least k, *SIAM J. Comput.*, Vol. 4, 393—396 (1975).
15.68. D. G. Corneil and B. Graham, An Algorithm for Determining the Chromatic Number of a Graph, *SIAM J. Comput.*, Vol. 2, 311—318 (1973).
15.69. C. McDiarmid, Determining the Chromatic Number of a Graph, *SIAM J. Comput.*, Vol. 8, 1—14 (1979).
15.70. D.J.A. Welsh and M. B. Powell, An Upper Bound for the Chromatic Number of a Graph and Its Applications to Timetabling Problems, *The Computer J.*, Vol. 10, 85—86 (1967).
15.71. D. C. Wood, A Technique for Colouring a Graph Applicable to Large Scale Timetabling Problems, *The Computer J.*, Vol. 10, 317—319 (1969).
15.72. D. Matula, G. Marble, and J. Isaacson, Graph Colouring Algorithms, in Graph Theory and Computing (R. C. Read, Ed.), Academic Press, New York, 1972, pp. 109—122.
15.73. D. Brélaz, New Methods to Color the Vertices of a Graph, *Comm. ACM*, Vol. 22, 251—256 (1979).
15.74. M. C. Paull and S. H. Unger, Minimizing the Number of States in Incompletely Specified Sequential Switching Functions, *IRE Trans. Elect. Comput.*, Vol. EC-8, 356—357 (1959).
15.75. C. Bron and J. Kerbosch, Finding All Cliques of an Undirected Graph — Algorithm 457, *Comm. ACM*, Vol. 16, 575—577 (1973).
15.76. E. A. Akkoyunlû, The Enumeration of Maximal Cliques of Large Graphs, *SIAM J. Comput.*, Vol. 2, 1—6 (1973).
15.77. S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, A New Algorithm for Generating All the Maximal Independent Sets, *SIAM J. Comput.*, Vol. 6, 505—517 (1977).
15.78. R. E Tarjan and A. E. Trojanowski, Finding a Maximum Independent Set, *SIAM J. Comput.*, Vol. 6, 537—546 (1977).

15.79. V. Chvátal, Determining the Stability Number of a Graph, *SIAM J. Comput.*, Vol. 6, 643—662 (1977).
15.80. J. B. Kruskal, Jr., On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem, *Proc. Am. Math. Soc.*, Vol. 7, 48—50 (1956).
15.81. R. C. Prim, Shortest Connection Networks and Some Generalizations, *Bell Sys. Tech. J.*, Vol. 36, 1389—1401 (1957).
15.82. A. Kerschenbaum and R. Van Slyke, Computing Minimum Spanning Trees Efficiently, Proc. 25th Ann. Conf. of the ACM, 1972, 518—527.
15.83. A. C. Yao, An $O(|E|\log\log|V|)$ Algorithm for Finding Minimum Spanning Trees, Information Processing Lett., Vol., 4, 21—23 (1975).
15.84. D. Cheriton and R. E. Tarjan, Finding Minimum Spanning Trees, *SIAM J. Comput.*, Vol. 5, 724—742 (1976).
15.85. H. N. Gabow, Two Algorithms for Generating Spanning Trees in Order, *SIAM J. Comput.*, Vol. 6, 139—150 (1977).
15.86. K. P. Esqaran and R. E. Tarjan, Augmentation Problems, *SIAM J. Comput.*, Vol. 5, 653—665 (1976).
15.87. A. Rosenthal and A. Goldner, Smallest Augmentations to Biconnect a Graph, *SIAM J. Comput.*, Vol. 6, 55—66 (1977).
15.88. E. Reghbati and D. G. Corneil, Parallel Computations in Graph Theory, *SIAM J. Comput.*, Vol. 7, 230—237 (1978).
15.89. J. Edmonds, Edge-Disjoint Branching, in Combinatorial Algorithms (R. Rustin, Ed.), Algorithmics Press, New York, 1973, pp. 91—96.
15.90. D. R. Fulkerson and G. C. Harding, On Edge-Disjoint Branchings, *Networks*, Vol. 6, 97—104 (1976).
15.91. R. E. Tarjan, Edge-Disjoint Spanning Trees and Depth-First Search, *Acta Informatica*, Vol. 6, 171—185 (1976).
15.92. L. Lovász, On Two Minimax Theorems in Graph, *J. Combinatorial Theory B*, Vol. 21, 96—103 (1976).
15.93. B. L. Golden and T. L. Magnanti, Deterministic Network Optimization: A Bibliography, *Networks*, Vol. 7, 149—183 (1977).
15.94. M. Fujii, T. Kasami, and K. Ninomiya, Optimal Sequencing of Two Equivalent Processors, *SIAM J. Appl. Math.*, Vol. 17, 784—789 (1969); Erratum, Vol. 20, 141 (1971).
15.95. E. G. Coffman, Jr., and R. L. Graham, Optimal Scheduling for Two-Processor Systems, *Acta Informatica*, Vol. 1, 200—213 (1972).
15.96. E. G. Coffman, Jr., and P. J. Denning, Operating System Theory, Prentice Hall, Englewood Cliffs, N.J., 1973.
15.97. R. Sethi, Scheduling Graphs on Two Processors, *SIAM J. Comput.*, Vol. 5, 73—82 (1976).
15.98. R. Sethi, Algorithms for Minimal Length Schedules, in Computers and Job Scheduling Theory (E. G. Coffman, Jr., Ed.), Wiley, New York, 1976, pp. 51—99.
15.99. J. D. Ullman Complexity of Sequencing Problems, in Computer and Job Scheduling Theory (E. G. Coffman, Jr., Ed.), Wiley, New York, 1976, pp. 139—164.
15.100. S. Lam and R. Sethi, Worst Case Analysis of Two Scheduling Algorithms, *SIAM J. Comput.*, Vol. 6, 518—536 (1977).
15.101. D. J. Rose, A Graph-Theoretic Study of Numerical Solutions of Sparse Positive Definite Systems of Linear Equations, in Graph Theory and Computing (R. C. Read, Ed.), Academic Press, New York, 1972, pp. 183—217.
15.102. T. Ohtsuki, A Fast Algorithm for Finding an Optimal Ordering for Vertex Elimination on a Graph, *SIAM J. Comput.*, Vol. 5, 133—145 (1976).
15.103. D. J. Rose, R. E. Tarjan, and G. S. Lueker, Algorithmic Aspects of Vertex Elimination on Graphs, *SIAM J. Comput.*, Vol. 5, 266—283 (1976).
15.104. R. E. Tarjan, Graph Theory and Gauss Elimination, in Sparse Matrix Computations (J. R. Bunch and D. J. Rose, Eds.), Academic Press, New York, 1976.

15.105. E. Horowitz and S. Sahni, Fundamentals of Computer Algorithms, Computer Science Press, Potomac, Md., 1978.
15.106. M. R. Garey and D. S. Johnson, Approximation Algorithms for Combinatorial Problems: An Annotated Bibliography, in Algorithms and Complexity: New Directions and Recent Results (J. Traub, Ed.), Academic Press, New York, 1976.
15.107. G. B. Dantzig, All Shortest Routes in a Graph, in Theory of Graphs, Gordon and Breach, New York, 1967, pp. 91—92.
15.108. S. Even, Algorithmic Combinatorics, Macmillan, New York, 1973,

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Аксиома баз матроида 198
— независимости матроида 193
— ранга матроида 198
— циклов матроида 199
Алгоритм *Габова* 385
— *Гильберта* и *Мура* 166
— *Данцига* 423
— двусвязности 340
— *Дейкстры* 364
— декомпозиции *Седербаума* схемной матрицы 263
— доминаторов 354
— «жадный» 212
— *Кана — Мункреса* 402
— кратчайших путей из данной вершины ко всем остальным вершинам 364
— *Ленгауэра* и *Тарьяна* 354
— максимального паросочетания в графе общего вида 385
— — — — двудольном графе 394
— — — — транспортной сети (помечивающий алгоритм) 410
— между всеми парами вершин 367
— нахождения минимального остова 424
— оптимального вектора длин путей 374
— — ветвления 421
— — дерева бинарного поиска 381
— — назначения 402
— *Пнueли, Лампла* и *Ивена* 323
— поиска в глубину в неориентированном графе 334
— — — — — ориентированном графе 334
— приближенный 423
— реализации графов с заданными степенями 156—159
— — матрицы сечений — цикломатической матрицы 268
— — $(n+1)$-узловой резистивной n-полюсной цепи 263
— сводимости графа программы 348
— сильной связности 343
— совершенного паросочетания 399
— столбце-ориентированный 319
— строко-ориентированный 319
— *Тарьяна* 348
— транзитивного замыкания 320
— транзитивной ориентации 323
— *Уоррена* 320
— *Уоршолла* 318
— *Флойда* 367
— *Форда — Фалкерсона* 197
— *Хаффмана* 374
— Анализ интервальный 345

База матроида 194
Базис векторного пространства 65
Блок графа 26

Валентность 12
Вектор базисный 65
— длин путей 370
— инциденций 100
— координаты 67
— линейно независимый 65
— ортогональный 67
— разреза 76
— циклический 76, 106
n-вектор 67
Вершина 11
— висячая 12
— внешняя 384
— внутренняя 17
— — n-полюсной цепи 17
— звезды 255
— изолированная 79
— инцидентная 12
— конечная 78
— — маршрута 80
— — ориентированного маршрута 80
— концевая 11, 16, 78, 80, 255
— короткозамыкаемая 255
— насыщенная (в паросочетании) 162
— начальная 80

— — в графе программы 131
— — маршрута 97
— — ориентированного маршрута 80
— несоотносимая 334
— полностью сканированная 332
— помеченная *88*
— связанная 18
— сильно связанная 81
— смежная 12, 79
— соответствующая усекаемой строке 101
— терминальная 16
— центральная 49
Вес ребра 59
— цикла 59
Ветвление оптимальное 418
Ветвь дерева 33
— неполюсная 247
— полюсная 247
Включение последовательности 141
Вычисления параллельные 210

Гаммоид 217
Гиперплоскость матроида 216
Гипотеза четырех красок 190
Глубина 332
Граф ассоциированный ориентированный 171
— ациклический 33, 80
— вершинно непересекающийся 28
— — k-раскрашиваемый 184
— взвешенный 135
— гамильтонов 55
— гамильтоново связный 61
— гомеоморфный 142
— двойственный 143
— двудольный 23
— деревьев 49
— k-дольный 23
— изоморфный 28
— 1-изоморфный 28
— 2-изоморфный 28
— квазисильно связный 83
— конденсированный 82
— *Коутса* 127
— кубический 31
— лежащий в основе неориентированный 79
— минимально связный 48, 82
— — — ориентированный 82
— *Мэзона* 131, 126
— неориентированный 11, 100, 103
— неразделимый 25
— однородный 23
— k-однородный 23
— ориентированный 11, 78, 100, 103
— — ациклический 80
— — гамильтонов 93
— — эйлеров 88
— параллельно-последовательный 150
— *Петерсена* 143
— планарный 137
— — с одной терминальной парой 150
— полный 23
— — двудольный 23
— — k-дольный 23
— — ориентированный 93
— потоков *Коутса* 127
— программы 130
— произвольно гамильтонов 59
— — эйлеров 54, 55
— простой 11
— пустой 11
— разделимый 25
— реберно непересекающийся 28
— — k-раскрашиваемый 180
— — k-хроматический 180
— κ-реберно связный 155, 172
— рефлексивный ориентированный 85
— самодвойственный 150
— самодополнительный 30
— сводимый 344
— связной 18, 80
— — ориентированный 80
— сигнальный *Мэзона* 131
— сильно связный 81
— симметрично ориентированный 85
— с одной терминальной парой 150
— стягиваемый 22
— транзитивно ориентируемый 85, 322
— тривиальный 25
— укладываемый 137
— k-факторизуемый 172
— k-хроматический 185
— эйлеров 51
Графоид 203
Группа абелева 62, 63

Декодирование 369
Дерево 33, 85
— бинарного поиска 376
— бинарное 87
— в графе 33
— венгерское 397
— доминаторов 351
— корневое 85
— линейное 255
— нормальное 236
— оптимальное бинарного поиска 378
— ориентированное 85

— ПВГ 333
— помеченное 117
— расширенное бинарного поиска 376
k-дерево остовное 37
Дефицит 163
Диакоптика 242
Диаметр графа 30
Длина взвешенных путей 87
— ориентированного ребра 361
— пути 17, 361
— цикла 17
Доминатор (вершины) непосредственный 351
Дополнение ортогональное 67
— подграфа 16
— простого графа 15
Дуга заходящая 79
— инцидентная 79
— исходящая 79
— параллельная 78

Задача о кёнигсбергских мостах 50
— — коммивояжере 59
— — свадьбах 161
— составления расписаний 396
Закон *Кирхгофа* для напряжений 218
— — — токов 218
Замыкание бинарного отношения транзитивное 316
— графа 61
— — транзитивное 316
— подмножества 216
Звезда 255

Изоморфность 66
Индекс хроматический 180
Источник напряжения 218
— тока 218

Клика 192
Кобаза матроида 200
Кодерево 33
k-кодерево 37
Код префиксный 88
Ко-лес 38
Колесо 38
Коллапсирование вершины 350
Комбинация линейная 65
— параллельная параллельно-последовательных графов 150
— последовательная параллельно-последовательных графов 150
Компонента 18, 169
— графа 80
— двусвязная 341
— нечетная 169
— ориентированного графа 80
— сильно связная 81
— четная 169
Конденсированный образ 82
Контур 80
— гамильтонов 93
Конфигурация полюсная 245
— — в виде звезды 245
— — — — линейного дерева 245
Координата вектора 67
Копетля матроида 200
Коранг матроида 200
Корень в ориентированном графе 85
— компоненты (в ПВГ) 336
— ПВГ 332
Коцикл 200
— базисный матроида 201
— матроида 200

Лемма о раскраске дуг 98
Лес 38
Лист 15

Мажорантность 56
Маршрут 16, 80
— замкнутый 17, 80
— ориентированный 79
— открытый 17
Матрица базисная разрезающих множеств 105
— — цикломатическая 107, 208
— бистохастическая 172
— гипердоминантная 258
— достижимости 121
— инциденций графа 100
— — ориентированного графа 100
— коциклическая матроида 209
— модифицированная сечений 289
— — цикломатическая 290
— однородно сужающаяся 260
— подстановок 172
— полных сопротивлений контура 225
— — — холостого хода 33
— полустепеней захода 121
— — исхода 121
— проводимостей короткого замыкания 246

— — сечений 248
— разрезов графа 103
— сечений 219
— — базисная 220
— смежности взвешенного ориентированного графа 126
— — графа 123, 126
— — ориентированного графа 123
— с равными алгебраическими дополнениями 117
степеней 117
— схемная 44
— узловых проводимостей 228
— унимодулярная 113
— усеченная инциденций 101
— цикломатическая графа 106
— — матроида 209
— — ориентированного графа 106
(0, 1)-матрица 166
Матроид 193
— бинарный 207
— графический 196
— двойственный 200
— двудольный 217
— изоморфный 196
— кографический 195
— неразделимый 216
— несвязный 216
— однородный 215
— ориентируемый 216
— паросочетаний 195
— правильный 217
— представимый 226
— разделимый 216
— разрезов 194
— регулярный 215
— связей 194, 216
— связный 216
— трансверсалей
— *Фано* 216
— циклический 194
— эйлеров 217
Место узкое 413
Метка вершины 384
— начальная 384
— ребра 384
Метод *Коутса* 126
— Мэзона 131
— *Татта* 121
Минор матроида 206
Множество базисное циклов 40, 43
— внутреннее устойчивое 174
— доминирующее 190
— зависимое матроида 194
— замкнутое матроида 216
— — по отношению к бинарному отношению 62
— инциденций 76
— лексикографически большее 213
— — максимальное 213
— независимое графа 174
— — максимальное графа 174
— матроида 194
— оптимальное по *Гейлу* 213
— разделяющее 152
— разрешающее 40
— — базисное 43, 76
L-множество 271
Модификация *Эдмондса* и *Карпа* помечивающего алгоритма 412
Мост 31

Назначение оптимальное 400
Напряжение между узлом и базой 222
— узловое 222
Нуль-граф 11
Нуль-пространство 207

Область планарного графа 139
Образ конденсированный графа 82
Обхват 30
Объединение графов 21
— матроидов 217
Ограничение графа 204
— матроида 205
— по пропускной способности 405
Окрестность 219
Операции над графами 62
Определитель главный 115
Оптимизация кода 374
— сети 410
Ориентация графа 78
— ребра 78
— транзитивная *109*
Ортогональность 67
Остов графа 33
— максимально удаленный 230
— ориентированный 87
Отец (вершины) 332
Отношение бинарное 84
— рефлексивное 84
— симметричное 84
— транзитивное 84
— эквивалентности 84
Отождествление (замыкание) вершины 22

Паросочетание максимальное 162
— полное (в двудольном графе) 163
— совершенное 169

Переворачивание 28
Переменная контура 222
— сечения 222
Пересечение графов 21
Петля 11, 78
— матроида 194
— обратной связи 133
Подграф 13, 79
— венгерский 387
— вершинно-порожденный 14
— главный 233
— критический 418
— максимальный 14
— минимальный 14
— остовный 13
— порожденный 14, 79
— равенств 188
— реберно-порожденный 14
— собственный 13
k-подграф 20
Поддерево 33
— левое 376
— правое 376
Подмножество зависимое 194
— максимальное 14, 199
— минимальное 14
Подпространство ортогональное 67
— разрезов 70
— циклов 69
Поиск в глубину в графе 332
— — — — ориентированном графе 335
— — ширину 360
Покрытие вершинное 174
— минимальное 175, 179
— реберное 179
Поле *Галуа* 63, 64
Полином хроматический 186
Полудоминатор 352
Полустепень захода 79
— исхода 79
Полюс концевой 255
— подобно ориентированный 255
— противоположно ориентированный 255
Помечивание обратное 410
— прямое 410
Порядок редукции 350
Последовательность выводимая 231
— графическая 56
— *де Брёйна* 88
— остаточная 157
— очков 97
— *Пруфера* 147
— степеней 56
— чередующаяся (по отношению к независимому множеству) 176
Поток максимальный 406
Потомок (вершины) собственный 334
Предок (вершины) собственный 334
Преобразование контурное 221
— полюсное 222
— сечения 222
— узловое 222
Проблема оптимизации кода 372
— NP-полная 359
— расширения 383
— NP-сложная 359
— четырех красок 189
Представление матроида стандартное 206
Проекция стереографическая 138
Произведение полных сопротивлений 293
— проводимостей 293
— скалярное векторов 67
Пространство векторное 64
— нуль матрицы 207
Проход (через вершину в ПВГ) 332
Псевдовершина 420
Путь 17, 180
— добавляющий 168
— f-дополняющий 408
— кратчайший 361
— — по отношению к паросочетанию 390
— максимальный 282
— f-ненасыщенный 195
— ориентированный 80
— — гамильтонов 55
— M-чередующийся 167

Разбиение главное 233
— двудольное 23, 176
Разделитель 42
Размерность векторного пространства 65
Разметка допустимая вершинная 399
Разрез ориентированный 42, 98
Ранг 82
— графа 39
— матроида 194
— ориентированного графа 82
Ранжирование 97
Раскраска 202
— k-вершинная 182
— — правильная 184
— ориентируемого матроида 211
— k-реберная 180
— — оптимальная 180
— — правильная 180
Расстояние между вершинами 30
— — остовами 230

Расщепление вершины 28
— точки сочленения 28
Реализация $(n+1)$-полюсной резисторной цепи 263
Ребро 11
— висячее 12
— критическое 418
— f-насыщенное 408
— независимое 162
— f-ненасыщенное 408
— нециклическое 17
— f-нулевое 408
— обратное 332
— — по отношению к ориентированному пути 407
— пересекающее 121
— подходящее 418
— f-положительное 408
— полюсное 245
— помеченное 329
— последовательное 141
— прямое 332
— — по отношению к ориентированному пути 407
— — — — — ПВГ 332
— рассмотренное (в ПВГ) 332
— светлое 167
— смежное 12, 79
— темное 167
— циклическое 17
Редукция транзитивная 359

Свойство ассоциативности 62
— коммутативности 62
— неусиления 241
— схемности 257
Связность вершинная 151
— реберная 155
Сеть транспортная 404
Сечение ориентированное 241
— — полюсной цепи 241
Система различных представителей 165
— уравнений контурная 224
— — сечений 226
— — со смешанными переменными 230
Скаляр 65
Слияние последовательности 141
Слово кодовое 372
Соединение параллельное 290
— псевдопоследовательное 290
— 1-факторное ориентированного графа 128
Сортировка топологическая 95
Составление расписания 403
Список смежности 316
Способность пропускная разреза 406
— — ребра 405
Степени свободы топологические 235
Степень вершины 79
— области 141
Сток 404
Стягивание ребра 22
Сужение матроида 204
Сумма кольцевая графов 21
— прямая подпространств 65
— характеристическая 372

Теорема *Бержа* о чередующейся цепи 168
— *Бине — Коши* 116
— *Брукса* 185
— *Визинга* 183
— добавления 196
— *Кёнига* 165
— *Кёнига — Эгервари* 166
— *Куратовского* 141
— *Кэли* 117
— *Менгера* 160
— *Мендельсона* и *Далмеджа* 169
— о максимальном потоке и минимальном разрезе 409
— — пяти красках 189
— *Татта* о совершенных паросочетаниях 170
— *Теллежена* 222
— *Холла* 164
— четырех красок 190
— *Эдмондса* о k-реберной связности 159
Теория экстремальных графов 190
Толщина 149
Точка сочленения 25
Трансверсаль частичная 165, 216
Транспозиция цепи 312
Треугольник 31
Турнир 96

Удаление вершины 22
— ребра 22
Узел 219
Укладка планарная 137
Улучшение реберной раскраски 183
Уравнения состояния 235
— узловые 228
Уязвимость 181
1-фактор ориентированного графа 124
k-фактор 172
Формула коэффициента усиления *Кайтса* 130

— — — *Мэзона* 133
— *Эйлера* 139
Функция Аккермана 136
— коранга матроида 200
— ранга матроида 194

Хорда 33
— неполюсная 249
— полюсная 249

«**Ц**веток» 168
Цепь 17, 80
— вывода 327
— замкнутая 17
— — ориентированная 80
— ориентированная 80
— — эйлерова 51, 88
— открытая 17, 80
— — ориентированная 80
— — — эйлерова 51, 88
— n-полюсная 246
— — нагруженная 246
— чередующаяся (по отношению к паросочетанию) 167
Цикл 17, 80
— базисный 40
— — матроида 410
— гамильтонов 55
— матроида 410
— ориентированный 80

Число вершинного покрытия 175
— внутренней устойчивости 174
— независимости 174
— паросочетаний 162
— реберного покрытия 179
— реберно-хроматическое 180
— скрещиваний 149
— хроматическое 185
— цикломатическое графа 82, 89

Эквивалентности классы 81
Элементы матроида 63, 193

Ячейка 140

Оглавление

Предисловие редактора перевода 5

Предисловие 7

Часть I. Теория графов 11

1. Основные понятия 11

1.1. Основные определения 11
1.2. Подграфы и дополнения 13
1.3. Маршруты, цепи, пути и циклы 16
1.4. Связность и компоненты графа 18
1.5. Операции над графами 19
1.6. Специальные графы 23
1.7. Точки сочленения и разделимые графы 25
1.8. Изоморфизм и 2-изоморфизм 27
1.9 Замечания, касающиеся литературы 29
Упражнения 30

2. Деревья, разрезающие множества и циклы 33

2.1. Деревья, остовы и кодеревья 33
2.2. k-деревья, остовные k-деревья, леса 37
2.3. Ранг и цикломатическое число 39
2.4. Базисные циклы 39
2.5. Разрезающие множества 40
2.6. Разрез 41
2.7. Базисные разрезающие множества 43
2.8. Остовы, циклы и разрезающие множества 44
2.9. Замечания, касающиеся литературы 47
Упражнения 48

3. Эйлеровы и гамильтоновы графы 50

3.1. Эйлеровы графы 51
3.2. Гамильтоновы графы 55
3.3. Замечания, касающиеся литературы 60
Упражнения 60

4. Графы и векторные пространства 62

4.1. Группы и поля 62
4.2. Векторные пространства 64
4.3. Векторное пространство графа 67
4.4. Размерность подпространств циклов и разрезов 71

4.5. Связь между подпространствами циклов и разрезов 73
4.6. Ортогональность подпространств циклов и разрезов 74
4.7. Замечания, касающиеся литературы 76
Упражнения 76

5. Ориентированные графы 78

5.1. Основные определения и понятия 78
5.2. Графы и отношения 84
5.3. Ориентированные и корневые деревья 85
5.4. Ориентированные эйлеровы графы 88
5.5. Ориентированные остовы и ориентированные эйлеровы цепи. . . 91
5.6. Ориентированные гамильтоновы графы 93
5.7. Ациклические ориентированные графы 95
5.8. Турниры 96
5.9. Замечания, касающиеся литературы 98
Упражнения 98

6. Матрицы графов 100

6.1. Матрица инциденций 100
6.2. Матрица разрезов 102
6.3. Цикломатическая матрица 105
6.4. Соотношение ортогональности 108
6.5. Подматрицы матриц разрезов, инциденций и циклов 110
6.6. Унимодулярные матрицы 113
6.7. Число остовов 115
6.8. Число остовных 2-деревьев 118
6.9. Число ориентированных остовов в ориентированном графе . . 120
6.10 Матрица смежности 123
6.11. Графы Коутса и Мэзона 126
6.12. Замечания, касающиеся литературы 134
Упражнения 134

7. Планарность и двойственность 137

7.1. Планарные графы 137
7.2. Формула Эйлера 139
7.3. Теорема Куратовского и другие характеризации планарности 141
7.4. Двойственные графы 143
7.5. Планарность и двойственность 147
7.6. Замечания, касающиеся литературы 149
Упражнения 149

8. Связность и паросочетания 151

8.1. Связность или вершинная связность 151
8.2. Реберная связность 155
8.3. Графы с заданными степенями 156
8.4. Теорема Менгера 160
8.5. Паросочетания 161
8.6. Паросочетания в двудольных графах 163
8.7. Паросочетания графов общего вида 167
8.8. Замечания, касающиеся литературы 171
Упражнения 171

9. Покрытия и раскраски 174

9.1. Независимые множества и вершинные покрытия 174
9.2. Реберные покрытия 179
9.3. Реберная раскраска и хроматический индекс 180
9.4. Вершинная раскраска и хроматическое число 184
9.5. Хроматические полиномы 186
9.6. Проблема четырех красок 189
9.7. Замечания, касающиеся литературы 190
Упражнения 190

10. Матроиды 193

10.1. Основные определения 193
10.2. Фундаментальные свойства 196
10.3. Эквивалентные системы аксиом 198
10.4. Двойственность матроидов и графоиды 200
10.5. Ограничение, сужение и миноры матроида 204
10.6. Представимость матроидов 206
10.7. Бинарные матроиды 207
10.8. Ориентируемые матроиды 210
10.9. Матроиды и «жадный» алгоритм 212
10.10. Замечания, касающиеся литературы 214
Упражнения 215

Часть II. Теория электрических цепей

11. Графы и электрические цепи 218

11.1. Преобразование контуров и сечений 220
11.2. Системы контурных уравнений и уравнений сечений 223
11.3. Метод смешанных переменных 228
11.4. Главное разбиение графа 230
11.5. Уравнения состояния 235
11.6. Свойство неусиления в резистивных цепях 241
11.7. Замечания, касающиеся литературы 242
Упражнения 243

12. Резистивные n-полюсные цепи 245

12.1. Введение 245
12.2. Y-матрицы резистивной n-полюсной цепи ранга n 253
12.3. Реализация $(n+1)$-узловых резистивных n-полюсных цепей (подход Седербаума) 263
12.4. Реализация цикломатической матрицы и матрицы сечений . . 268
12.5. Реализация $(n+1)$-узловых резистивных n-полюсных цепей (подход Гуиллемина) 281
12.6. Замечания, касающиеся литературы 287
Упражнения 288

13. Функция цепи и чувствительность цепи 292

13.1. Топологические формулы для RLC-цепей без взаимных индуктивностей 292
13.2. Топологические формулы для общих линейных цепей 298

13.3. Сопряженная цепь и вычисление чувствительности цепи . . . 305
13.4. Замечания, касающиеся литературы 312
Упражнения 313

Часть III. Теория электрических цепей

14. Алгоритмы анализа графов 315

14.1. Транзитивное замыкание 316
14.2. Транзитивная ориентация 322
14.3. Поиск в глубину 331
14.4. Двусвязность и сильная связность 338
14.5. Сводимость графа программы 343
14.6. Доминаторы в графе программы 351
14.7. Замечания, касающиеся литературы 359
Упражнения 359

15. Оптимизационные алгоритмы 361

15.1. Кратчайшие пути 361
15.2. Деревья с минимальной длиной взвешенных путей 368
15.3. Оптимальные деревья бинарного поиска 376
15.4. Максимальные паросочетания в графе 380
15.5. Максимальные паросочетания в двудольном графе 389
15.6. Совершенное паросочетание, оптимальное назначение и составление расписаний 396
15.7. Потоки в транспортной сети 404
15.8. Оптимальное ветвление 418
15.9. Замечания, касающиеся литературы 422
Упражнения 423

Литература 425

Предметный указатель 444