

А. А. Вороненко

ОСНОВЫ КИБЕРНЕТИКИ

Учебное пособие

Москва — 2018

УДК 519.71

Рецензенты: *Дьяконов А. Г.*, профессор, д. ф.-м. н.

Дайняк А. Б., доцент, к. ф.-м. н.

Вороненко А. А.

Основы кибернетики: Учебное пособие. — М.: Издательство, 2018.
— 186 с.

В пособии представлены материалы курса «Основы кибернетики», читающегося автором на четвертом курсе факультета ВМК МГУ, расширенные набором результатов из других курсов с тем же названием и современными научными результатами. Пособие рассчитано на студентов третьего и четвертого курсов.

Автор выражает благодарность всем своим коллегам за существенную помощь в подготовке пособия.

E-mail: `dm6@cs.msu.ru`

© Вороненко А. А. 2018

Оглавление

Введение	8
1 Дизъюнктивные нормальные формы	10
1.1 Определение. Геометрическая интерпретация. Совершенная ДНФ	10
1.2 Сложность ДНФ	12
1.3 Функции Шеннона для ДНФ	12
1.4 Тупиковая ДНФ. Сокращенная ДНФ и методы ее построения	14
1.5 Карты Карно. Геометрический подход	18
1.6 ДНФ Квайна	23
1.7 Пример функции с большим количеством тупиковых и минимальных ДНФ	25
1.8 ДНФ сумма тупиковых	27
1.9 Критерий вхождения конъюнкции в ДНФ сумма тупиковых	29
1.10 Сокращенная ДНФ для монотонных функций	30
1.11 Максимальная длина и сложность сокращенной ДНФ	31

1.12 Алгоритм построения всех тупиковых покрытий матрицы	33
1.13 Неразрешимость нахождения локальными алгоритмами ДНФ сумма минимальных	34
2 Градиентный алгоритм	39
2.1 Оценка сложности покрытия градиентного алгоритма	39
2.2 Пример модификации градиентного алгоритма	41
3 Сложность реализации булевых функций СФЭ и КС	46
3.1 Нижняя оценка функции Шеннона для СФЭ	46
3.2 Универсальный многополюсник. Дешифратор	50
3.3 Метод Шеннона для синтеза СФЭ в базисе $\{\&, \vee, \neg\}$	54
3.4 Метод О.Б. Лупанова для синтеза схем в базисе $\{\&, \vee, \neg\}$	55
3.5 Контактные схемы. Метод Шеннона	59
3.6 Нижняя оценка функции Шеннона для КС	63
3.7 Метод каскадов для КС и СФЭ	66
3.8 Самокоррекция контактных схем	71

3.9	Схема Кардо. Тестирование на размыкание . . .	75
3.10	Самокорректирующиеся схемы Кардо	77
4	Эквивалентные преобразования формул	81
4.1	Эквивалентные преобразования формул в базисе $\{\&, \vee, \neg, 0, 1\}$	81
4.2	Теорема перехода	84
4.3	Теорема Янова	87
4.4	Теорема Мучника	88
4.5	Пример Линдона. Свойства функции Линдона . .	90
4.6	Бесконечная полная система тождеств для класса, порожденного функцией Линдона . .	91
4.7	Отсутствие КПСТ для замыкания системы из функции Линдона	92
5	Проблема контроля управляющих систем . . .	94
5.1	Тесты для таблиц. Тривиальные оценки	94
5.2	Верхняя оценка длины диагностического теста для почти всех таблиц	96
5.3	Алгоритм построения всех тупиковых тестов	97
6	Вопросы сложности алгоритмов	99
6.1	Проблема NP -полноты. Теорема Кука (формулировка)	99
6.2	Язык КЛИКА. NP -полнота языка КЛИКА . . .	102
6.3	NP -полнота языка 3-ВЫП	103

6.4	Полиномиальность языка 2-ВЫП	105
6.5	Задача о кратчайшем остовном дереве. Жадный алгоритм	107
6.6	Свойства приближенных алгоритмов для задачи МВП	108
7	Дополнительные вопросы	113
7.1	Сортировка массива. Сортировка слиянием . . .	113
7.2	Об одновременном нахождении максимума и минимума в массиве	116
7.3	Нахождение k -го наименьшего элемента в массиве из n элементов	117
7.4	Задача тестирования линейности булевой функции	121
7.5	Задача доказательства нелинейности булевой функции	122
7.6	Дешифровка монотонных функций	123
7.7	Распознавание монотонности по вектор-столбцу .	130
7.8	Критерий кографа. Кографы и 2-деревья	134
7.9	Каноническое дерево неповторной функции в элементарном базисе	140
7.10	Наличие запретных конфигураций у функции Стеценко	144
7.11	Повторность. Свойство Субботовской	147
7.12	Наличие запретных конфигураций у функции Субботовской	149

Оглавление	7
7.13 Теорема В.А. Стеценко	150
8 Задачи	159
Библиографический список	185

Введение

Курс «Основы кибернетики» является продолжением курса «Дискретная математика», читающегося во втором семестре для бакалавров факультета ВМК. Для освоения курса «Основы кибернетики» необходимы минимальные знания линейной алгебры и математического анализа (соответствующий материал полностью входит в программу первых семестров соответствующих курсов). Из курса «Дискретная математика» (см.[10]) необходимо знать разделы по булевым функциям, теории графов и сложности схем. Основными практическими навыками являются умение оперировать с булевыми функциями в различных представлениях и четкое владение элементарными понятиями теории графов (путь, цикл, орграф, изоморфизм и связанные с ними).

Ожидаемые навыки от слушателей по итогам курса.

1. Умение решать задачи на различные представления булевых функций в виде ДНФ.
2. Умение решать задачи, связанные с понятием NP -полноты.

3. Понимание проблематики задач на эквивалентные преобразования и умение решать задачи, связанные со свойствами абстрактных дискретных функций.
4. Углубленные знания по теории схемной сложности.
5. Знание элементарных результатов по тестированию (теории надежности)
6. Понимание ряда нестандартных современных понятий, близких к вышеизложенным (соответствующие разделы читаются вариативно в объеме одной-двух лекций).

Глава 1

Дизъюнктивные нормальные формы

1.1 Определение. Геометрическая интерпретация. Совершенная ДНФ

Через x^0 обозначим отрицание переменной x (основное обозначение — $\bar{x}$), а через x^1 — саму переменную x . Произвольную запись x^σ , где σ принимает значения 0 или 1, назовем *литералом*, или *буквой*. Выражение K вида $x_{i_1}^{\sigma_1} \& \dots \& x_{i_s}^{\sigma_s}$, где все i_j различны, называется *элементарной конъюнкцией*. Конъюнкции считаются различными, если они отличаются набором литералов. Выражение вида $K_1 \vee \dots \vee K_j$, где все K_i — различные элементарные конъюнкции, называется *дизъюнктивной нормальной формой* (ДНФ).

Конъюнкции $K = x_{i_1}^{\sigma_1} \& \dots \& x_{i_s}^{\sigma_s}$, входящей в ДНФ функции $f(x_1, \dots, x_n)$, в булевом кубе соответствует грань размерности $n - s$, состоящая из единиц этой элементарной конъюнкции.

Кодом этой грани называется набор $(-\dots-\sigma_1-\dots-\dots-\sigma_s-\dots-)$ из символов 0, 1 и $-$, где каждая константа σ_j расположена на i_j -й позиции. Для того чтобы набор из $\{0, 1\}^n$ попадал в эту грань, все его i_j -е координаты должны быть равны σ_j , а остальные — произвольны.

Для функции $f(x_1, \dots, x_n)$ можно рассмотреть разложение по всем переменным:

$$\begin{aligned} f(x_1, \dots, x_n) &= \bigvee_{(\sigma_1, \dots, \sigma_n)} x_1^{\sigma_1} \& \dots \& x_n^{\sigma_n} \& f(\sigma_1, \dots, \sigma_n) = \\ &= \bigvee_{(\sigma_1, \dots, \sigma_n) | f(\sigma_1, \dots, \sigma_n) = 1} x_1^{\sigma_1} \& \dots \& x_n^{\sigma_n}. \end{aligned}$$

Последнее представление является ДНФ, оно справедливо для любой функции, отличной от тождественного нуля, и называется *совершенной дизъюнктивной нормальной формой* (совершенной ДНФ). При дальнейшем изложении вопросов, подразумевающих автоматическое существование ДНФ, предполагается, что функция не равна тождественному нулю.

Вопросы и задания.

1. Дайте определения следующих понятий:

- буква (литерал);
- элементарная конъюнкция (ЭК);
- дизъюнктивная нормальная форма (ДНФ);
- совершенная дизъюнктивная нормальная форма (совершенная ДНФ).

* — задание повышенной сложности.

1.2 Сложность ДНФ

Для ДНФ используются различные меры сложности. Количество букв в ДНФ D обозначается через $L(D)$ и носит название *сложности* ДНФ, а количество конъюнкций — через $l(D)$ и называется *длиной* ДНФ для заданной функции f , на которых достигается минимум данных функционалов, называются соответственно *минимальными* и *кратчайшими*, а значение их мер сложности называется *сложностью* функции f и обозначается $L(f)$ (соответственно *длиной* функции $l(f)$).

1.3 Функции Шеннона для ДНФ

Функция Шеннона $L(n)$ для сложности определяется как максимум по всем функциям n переменных величины $L(f)$. Функция Шеннона $l(n)$ для длины определяется как максимум по всем функциям n переменных величины $l(f)$. Фактически функция Шеннона определяется как мера сложности самой сложной функции. Для ДНФ функцию Шеннона удастся вычислить точно как для функционала сложности, так и для функционала длины.

Т е о р е м а 1.

$$L(n) = n \cdot 2^{n-1}, \quad l(n) = 2^{n-1}.$$

Д о к а з а т е л ь с т в о. Рассмотрим произвольную функцию $f(x_1, \dots, x_n)$. Воспользуемся ее разложением по $n - 1$ пе-

переменной:

$$f(x_1, \dots, x_n) = \bigvee_{(\sigma_1, \dots, \sigma_{n-1})} x_1^{\sigma_1} \& \dots \& x_{n-1}^{\sigma_{n-1}} \& f(\sigma_1, \dots, \sigma_{n-1}, x_n).$$

Данное разложение представляет любую функцию в виде дизъюнкции не более чем 2^{n-1} конъюнкций не более n литералов, что обеспечивает верхнюю оценку для обеих функций Шеннона $L(n) \leq n \cdot 2^{n-1}$, $l(n) \leq 2^{n-1}$. Для получения нижней оценки рассмотрим функцию

$$g(x_1, \dots, x_n) = x_1 \oplus \dots \oplus x_n.$$

Поскольку изменение значения любой переменной меняет значение функции $g(x_1, \dots, x_n)$, любое слагаемое для ее ДНФ должно содержать все литералы и обращаться в единицу ровно на одном наборе (в одной точке). Количество единиц функции $g(x_1, \dots, x_n)$ равно 2^{n-1} , поэтому число слагаемых в ДНФ будет также не меньше 2^{n-1} , а количество букв — не меньше $n \cdot 2^{n-1}$. Из этого следует, что $L(n) \geq n \cdot 2^{n-1}$, $l(n) \geq 2^{n-1}$. Из последнего неравенства и верхней оценки следует утверждение теоремы. $\square$

Вопросы и задания.

1. Дайте определения следующих понятий:

- а) сложность и минимальная сложность ДНФ;
- б) длина и кратчайшая длина ДНФ;

- в) функция Шеннона для сложности ДНФ ($L(n)$) и для длины ДНФ ($l(n)$).
2. Сформулируйте теорему о функции Шеннона для сложности ДНФ и для длины ДНФ.
- 3*. Может ли кратчайшая ДНФ не быть минимальной?
- 4*. Получить верхнюю оценку длины кратчайших ДНФ и верхнюю оценку длины сложности ДНФ для почти всех функций.

1.4 Тупиковая ДНФ. Сокращенная ДНФ и методы ее построения

Пусть задана ДНФ $D = K_1 \vee \dots \vee K_j$, реализующая функцию $f(x_1, \dots, x_n)$. Рассмотрим следующие преобразования.

1. Удаление множителя (буквы) из конъюнкции — замена конъюнкции $K_i = K'_i x$ на конъюнкцию K'_i .
2. Удаление конъюнкции из ДНФ.

Эти преобразования уменьшают сложность ДНФ, а второе из них уменьшает также и ее длину. Однако применение любого из этих преобразований может привести к изменению реализуемой функции. Если ни одно из преобразований 1–2 нельзя применить к ДНФ D , реализующей функцию f , без изменения

реализуемой функции, то ДНФ D называется тупиковой для функции f .

Остановимся более внимательно на первой операции. Конъюнкцию K назовем *допустимой* для функции f (*импликантой* функции f), если для любого набора $\mathbf{x}$ выполняется неравенство $K(\mathbf{x}) \leq f(\mathbf{x})$. Любая конъюнкция, входящая в ДНФ для функции f , является импликантой этой функции.

Импликанта функции f называется *простой*, если при удалении любой буквы она перестает быть импликантой функции f . *Сокращенной ДНФ* функции f называется дизъюнкция всех простых импликант функции f . Поскольку любая конъюнкция однозначно либо является, либо не является простой импликантой функции f , сокращенная ДНФ любой функции f также определяется однозначно (напомним, что различные ДНФ функции определяются с точностью до порядков слагаемых и букв внутри конъюнкций).

По аналогии с дизъюнктивной нормальной формой (ДНФ) вводится понятие конъюнктивной нормальной формы (КНФ). Выражение S вида $x_{i_1}^{\sigma_1} \vee \dots \vee x_{i_s}^{\sigma_s}$, где все i_j различны, называется *элементарной дизъюнкцией*. Дизъюнкции считаются различными, если они отличаются набором литералов. Выражение вида $S_1 \& \dots \& S_j$, где все S_i — различные элементарные дизъюнкции, называется *конъюнктивной нормальной формой* (КНФ).

Метод Нельсона построения сокращенной ДНФ функции f состоит в следующем.

1. Взять произвольную КНФ, реализующую функцию f .
2. Раскрыть скобки, удаляя лишние буквы по правилу $x \cdot x = x$ и удаляя лишние конъюнкции, одновременно содержащие буквы x и $\bar{x}$.
3. Произвести поглощения конъюнкций по правилу $K_1 \vee \vee K_1 K_2 = K_1$.

Т е о р е м а 2. Метод Нельсона корректно строит сокращенную ДНФ функции, не являющейся константой 0 или 1.

Д о к а з а т е л ь с т в о. Во-первых, в результате работы алгоритма метода Нельсона будет построена ДНФ, реализующая функцию f , так как все преобразования не меняют реализуемую функцию (являются эквивалентными), а результатом является ДНФ. Во-вторых, все импликанты, не являющиеся простыми, будут поглощены на третьем этапе при условии, что при раскрытии скобок будут получены все простые импликанты. Докажем, что это будет именно так. Рассмотрим произвольную простую импликанту функции $f(x_1, \dots, x_n)$, равную $K = x_{i_1}^{\sigma_1} \& \dots \& x_{i_s}^{\sigma_s}$. Рассмотрим подстановку $x_{i_1} = \sigma_1, \dots, x_{i_s} = \sigma_s$. В силу того, что K — импликанта, КНФ, реализующая функцию f , должна обращаться в единицу при любых значениях оставшихся переменных. Последнее

означает, что в каждой скобке КНФ есть литерал из множества $\{x_{i_1}^{\sigma_1}, \dots, x_{i_s}^{\sigma_s}\}$. Рассмотрим теперь произвольную конъюнкцию, получающуюся при раскрытии скобок, если брать в каждой скобке литерал из множества $\{x_{i_1}^{\sigma_1}, \dots, x_{i_s}^{\sigma_s}\}$. Данная конъюнкция будет допустимой для функции f , а также будет содержать подмножество букв ее простой импликанты K . Следовательно, это будет конъюнкция K . $\square$

Метод Блейка построения сокращенной ДНФ [1].

Взять произвольную ДНФ функции $f(x_1, \dots, x_n)$ и производить, пока возможно, сначала первое из преобразований, затем, пока возможно, второе.

1. Обобщенное склеивание: $K_1x \vee K_2\bar{x} = K_1x \vee K_2\bar{x} \vee K_1K_2$.

2. Поглощение конъюнкций по правилу $K_1 \vee K_1K_2 = K_1$.

Т е о р е м а 3. Метод Блейка всегда строит сокращенную ДНФ.

Д о к а з а т е л ь с т в о. Достаточно доказать, что после выполнения всех возможных обобщенных склеиваний получится ДНФ, содержащая все простые импликанты. На множестве ДНФ введем отношение частичного порядка. Пусть $D = K_1 \vee \dots \vee K_s$ и $D' = K'_1 \vee \dots \vee K'_t$. Тогда $D \leq D'$, если для любой конъюнкции K_i найдется конъюнкция K'_j , такая, что $K_i \leq K'_j$. Пусть после всех обобщенных склеиваний не все простые импликанты входят в полученную ДНФ D . Рассмотрим

рим множество R импликант, не находящихся в отношении « $\leq$ » с ДНФ D . Среди них есть импликанта K с наибольшей длиной. В силу того, что D реализует функцию $f(x_1, \dots, x_n)$, эта длина не может быть равна n . При этом существует переменная x_i , такая, что $x_i K \leq D$ и $\bar{x}_i K \leq D$. Последнее означает наличие в ДНФ D конъюнкций P и Q , таких, что $x_i K \leq P$ и $\bar{x}_i K \leq Q$. Если конъюнкция P свободна от литерала x_i (конъюнкция Q свободна от литерала $\bar{x}_i$), то $K \leq P$ ($K \leq Q$). Это противоречит тому, что $K \in R$. Если же $P = x_i P'$, $Q = \bar{x}_i Q'$, то применимо обобщенное склеивание

$$P'x_i \vee Q'\bar{x}_i = P'x_i \vee Q'\bar{x}_i \vee P'Q'.$$

При этом $K \leq P'$, $K \leq Q'$, откуда $K \leq P'Q'$, что вновь противоречит тому, что $K \in R$. Следовательно, такой конъюнкции K не существует, множество R пусто, что доказывает теорему. $\square$

1.5 Карты Карно. Геометрический подход

Для построения сокращенной ДНФ функции четырех переменных используется ее представление в виде квадратной, а точнее тороидальной, таблицы [18]. Четырем строкам соответствуют записанные в состоящем из соседних наборов коде Грея $\{(0, 0), (0, 1), (1, 1), (1, 0)\}$ значения переменных x_1, x_2 , а столбцам — записанные в коде Грея $\{(0, 0), (0, 1), (1, 1), (1, 0)\}$

значения переменных x_3, x_4 . При этом паре соседних строк — первой и второй — соответствуют все значения наборов при $x_1 = 0$, а паре соседних строк — третьей и четвертой — соответствуют все значения наборов при $x_1 = 1$. Пары соседних столбцов — первому и второму — соответствуют все значения наборов при $x_3 = 0$, а паре соседних столбцов — третьему и четвертому — соответствуют все значения наборов при $x_3 = 1$. Пары соседних строк — первой и четвертой — соответствуют все значения наборов при $x_2 = 0$, а пары соседних строк — второй и третьей — соответствуют все значения наборов при $x_2 = 1$. Пары соседних столбцов — первому и четвертому — соответствуют все значения наборов при $x_4 = 0$, а пары соседних столбцов — второму и третьему — соответствуют все значения наборов при $x_4 = 1$. Ниже приведена таблица положения наборов при задании функции картой Карно по номерам (в десятичной записи).

		x_3	0	0	1	1
		x_4	0	1	1	0
x_1	x_2					
0	0		0	1	3	2
0	1		4	5	7	6
1	1		12	13	15	14
1	0		8	9	11	10

Карта Карно рассматривается как тор. Прямоугольники из

единиц площади, равной степени двойки (1, 2, 4, 8, 16), назовем 2-прямоугольниками. Из вышеизложенного следует теорема.

Т е о р е м а 4. *Существует взаимно-однозначное соответствие между максимальными 2-прямоугольниками на карте Карно для произвольной функции $f(x_1, x_2, x_3, x_4)$ и простыми импликантами этой функции.*

Метод карт Карно можно применить и для функций трех переменных, однако для функций пяти и большего числа переменных он принципиально не работает. Для построения сокращенной ДНФ функций четырех переменных чаще всего имеет смысл выбирать между методом карт Карно и методом Нельсона. Выбор в пользу метода карт Карно делается при отсутствии КНФ с небольшим числом сомножителей.

При применении метода карт Карно случаются как правило две ошибки: неправильное заполнение таблицы из-за незнания кода Грея и пропуск «тороидальных» граней.

Геометрический подход. При геометрическом подходе вершины куба рисуются точками на плоскости и соединяются ребрами так, чтобы ребра, соответствующие изменению одной переменной, были параллельны. Наборы, на которых функция равна единице, помечаются звездочкой, и соседние вершины, помеченные звездочкой, соединяются жирными ребрами. Грани, соответствующие простым импликантам, восстанавливаются по соединенным звездочкам естественным образом. Тео-

ретически данный подход применим при любой размерности булева куба (количестве переменных функции). На практике он весьма удобен при $n = 3$ (см. рис. 1.1). При применении метода на размерности четыре начинают возникать следующие ошибки: неверное размещение звездочек, неверное соединение по четвертой размерности, неверное декодирование граней, пропуск граней.

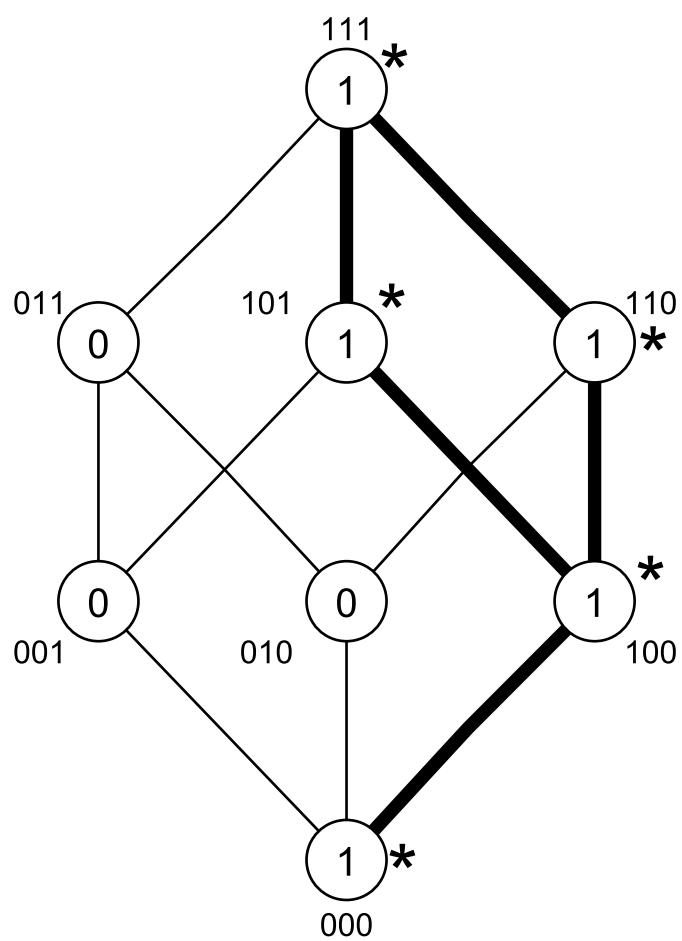


Рис. 1.1

Вопросы и задания.

1. Дайте определения следующих понятий:
 - а) простая импликанта (ПИ);
 - б) сокращенная ДНФ;
 - в) тупиковая ДНФ;
 - г) элементарная дизъюнкция (ЭД);
 - д) конъюнктивная нормальная форма (КНФ).
2. Сформулируйте следующие теоремы:
 - а) о методе Нельсона;
 - б) о методе Блейка;
 - в) о карте Карно.
3. Опишите принцип работы:
 - а) метода Нельсона,
 - б) метода Блейка,
 - в) карт Карно.
- 4*. Почему метод карт Карно не работает при числе переменных функции, большем четырех?

1.6 ДНФ Квайна

Рассмотрим простую импликанту K , которая обладает следующим свойством: найдется такой набор α , для которого им-

импликанта K будет единственной простой импликантой, обращающейся на наборе α в единицу. Такие точки и соответствующие им простые импликанты называются *ядровыми*. Множество ядровых простых импликант называется *ядром*. Ядровая импликанта входит в любую тупиковую ДНФ. Множество конъюнкций $\{K_1, \dots, K_s\}$ покрывает конъюнкцию K , если для любого $\mathbf{x}$, такого, что $K(\mathbf{x}) = 1$, выполняется соотношение $K_1(\mathbf{x}) \vee \dots \vee K_s(\mathbf{x}) = 1$. ДНФ Квайна — это ДНФ, получаемая из сокращенной удалением конъюнкций, покрываемых ядром.

Пример. Функция $f(x, y, z) = (0100 \ 0111)$. Ее сокращенная ДНФ имеет вид $xy \vee \bar{y}z \vee xz$. Импликанты xy и $\bar{y}z$, покрывающие наборы (110) и (001) соответственно являются ядровыми. В то же время они покрывают импликанту xz . Таким образом, ДНФ Квайна функции f имеет вид $xy \vee \bar{y}z$.

Вопросы и задания.

1. Дайте определения следующих понятий: ядровая точка, ядро, ДНФ Квайна.

1.7 Пример функции с большим количеством тупиковых и минимальных ДНФ

Рассмотрим функцию $g(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$. Ее сокращенная ДНФ имеет вид

$$x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee x_3\bar{x}_1 \vee x_1\bar{x}_3 \vee x_3\bar{x}_2 \vee x_2\bar{x}_1.$$

Эта функция имеет (см. зад. 10 главы 8) две минимальных ДНФ

$$x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee x_3\bar{x}_1,$$

$$x_1\bar{x}_3 \vee x_3\bar{x}_2 \vee x_2\bar{x}_1$$

и еще три тупиковых неминимальных

$$x_1\bar{x}_2 \vee x_3\bar{x}_1 \vee x_1\bar{x}_3 \vee x_2\bar{x}_1,$$

$$x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee x_3\bar{x}_2 \vee x_2\bar{x}_1,$$

$$x_2\bar{x}_3 \vee x_3\bar{x}_1 \vee x_1\bar{x}_3 \vee x_3\bar{x}_2.$$

На рис. 1.2 показаны первая минимальная и первая тупиковая неминимальная ДНФ функции $g(x_1, x_2, x_3)$.

Итого, функция $g(x_1, x_2, x_3)$ имеет две минимальных и пять тупиковых ДНФ.

Рассмотрим теперь функцию $f(x_1, \dots, x_n) = g(x_1, x_2, x_3) \oplus x_4 \oplus \dots \oplus x_n$. Произвольная импликанта K функции $f(x_1, \dots, x_n)$ обязана содержать литералы, соответствующие всем переменным, начиная с четвертой, так как при изменении любого аргумента сумма по модулю два меняет значение. Пусть $f(\sigma_1, \dots, \sigma_n) = 1$ и $\sigma_4 \oplus \dots \oplus \sigma_n = 1$. Тогда K содержит

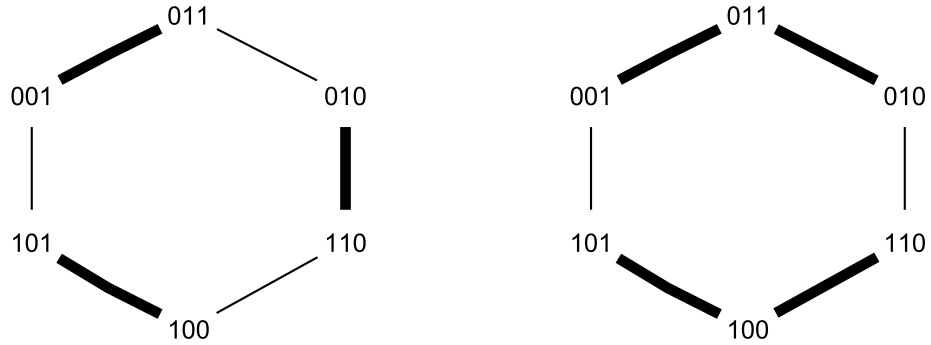


Рис. 1.2

все литералы и имеет вид либо $x_1 \cdot x_2 \cdot x_3 \cdot x_4^{\sigma_4} \cdot \dots \cdot x_n^{\sigma_n}$, либо $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4^{\sigma_4} \cdot \dots \cdot x_n^{\sigma_n}$. При $\sigma_4 \oplus \dots \oplus \sigma_n = 0$ функция $g(x_1, x_2, x_3)$ должна быть равна единице. Для таких наборов $(\sigma_4, \dots, \sigma_n)$ в тупиковую (минимальную) ДНФ функции f мы выбираем конъюнкции вида $x_i \cdot \bar{x}_j \cdot x_4^{\sigma_4} \cdot \dots \cdot x_n^{\sigma_n}$, соответствующие одной из тупиковых (минимальных) ДНФ функции $g(x_1, x_2, x_3)$. Число наборов $(\sigma_4, \dots, \sigma_n)$, таких, что $\sigma_4 \oplus \dots \oplus \sigma_n = 1$, равно 2^{n-4} . Поэтому функция $f(x_1, \dots, x_n)$ имеет $5^{2^{n-4}}$ тупиковых и $2^{2^{n-4}}$ минимальных ДНФ.

Вопросы и задания.

- 1*. Усилить пример параграфа, используя последовательность функций вида $g(x_1, x_2, x_3, x_4) \oplus x_5 \oplus \dots \oplus x_n$.

1.8 ДНФ сумма тупиковых

ДНФ сумма тупиковых (ДНФ ΣT) для заданной функции $f(x_1, \dots, x_n)$ называется дизъюнкция простых импликант, входящих хотя бы в одну тупиковую ДНФ функции $f(x_1, \dots, x_n)$.

Л е м м а 1. Если конъюнкция входит в ДНФ ΣT функции $f(x_1, \dots, x_n)$, то она входит и в ДНФ Квайна этой функции.

Д о к а з а т е л ь с т в о. Ядровые конъюнкции входят в любую тупиковую ДНФ. Следовательно, любая импликанта, покрываемая ядром (т.е. не входящая в ДНФ Квайна), не входит ни в одну из тупиковых ДНФ, значит, и в ДНФ ΣT . $\square$

Рассмотрим функцию $g(x_1, x_2, x_3, x_4)$, задаваемую таблицей:

		x_3	0	0	1	1
		x_4	0	1	1	0
x_1	x_2					
0	0		1	1	1	1
0	1		1	0	1	1
1	1		1	0	0	1
1	0		1	1	1	0

Функция $g(x_1, x_2, x_3, x_4)$ имеет семь простых импликант, каждая из которых является конъюнкцией двух литералов. Три из импликант являются ядровыми:

- 1) $K_1 = \bar{x}_1 \cdot x_3$ с ядровой точкой (0111);
- 2) $K_2 = x_2 \cdot \bar{x}_4$ с ядровой точкой (1110);
- 3) $K_3 = \bar{x}_2 \cdot x_4$ с ядровой точкой (1011).

ДНФ $K_1 \vee K_2 \vee K_3$ принимает значение 1 на всех наборах, на которых функция $g(x_1, x_2, x_3, x_4)$ обращается в единицу, за исключением двух: (0000) и (1000).

Таблицей Квайна булевой функции называется таблица, строки которой соответствуют простым импликантам, столбцы — наборам, на которых функция обращается в единицу, а элементы — значениям импликант на наборах. Приведем подтаблицу таблицы Квайна для оставшихся четырех конъюнкций и этих двух наборов.

	(0000)	(1000)
$K_4 = \bar{x}_3 \cdot \bar{x}_4$	1	1
$K_5 = \bar{x}_2 \cdot \bar{x}_3$	1	1
$K_6 = \bar{x}_1 \cdot \bar{x}_2$	1	0
$K_7 = \bar{x}_1 \cdot \bar{x}_4$	1	0

Для покрытия точки (1000) нужно взять одну из импликант K_4 или K_5 . При этом $K_1 \vee K_2 \vee K_3 \vee K_4 = g(x_1, x_2, x_3, x_4)$ и $K_1 \vee K_2 \vee K_3 \vee K_5 = g(x_1, x_2, x_3, x_4)$, т.е. $K_1 \vee K_2 \vee K_3 \vee K_4$ и $K_1 \vee K_2 \vee K_3 \vee K_5$ — тупиковые ДНФ, а конъюнкции K_6 и K_7 ни в одну тупиковую ДНФ не входят. При этом конъюнкции K_6 и K_7 входят в ДНФ Квайна, так как не покрываются

ядром $\{K_1, K_2, K_3\}$. Таким образом, ДНФ Квайна функции $g(x_1, x_2, x_3, x_4)$ отлична от ее ДНФ ΣT .

1.9 Критерий вхождения конъюнкции в ДНФ сумма тупиковых

Пусть задана функция f . Для произвольного $\mathbf{x}$, такого, что $f(\mathbf{x}) = 1$, будем называть *лучком* $\Pi(\mathbf{x})$, проходящим через точку $\mathbf{x}$, множество всех простых импликант K функции f , таких, что $K(\mathbf{x}) = 1$. Точка α называется *регулярной* относительно грани (простой импликанты) K , если существует точка β , такая, что $f(\beta) = 1$, $K(\beta) = 0$ и при этом $\Pi(\beta) \subseteq \Pi(\alpha)$. Конъюнкция (грань) называется *регулярной*, если все ее точки регулярны относительно этой конъюнкции (грани).

Т е о р е м а 5. (Первая теорема Ю.И.Журавлева. Критерий вхождения конъюнкции в ДНФ сумма тупиковых). Простая импликанта входит в ДНФ сумма тупиковых тогда и только тогда, когда соответствующая ей грань не является регулярной.

Д о к а з а т е л ь с т в о. Необходимость. Пусть произвольная ДНФ содержит конъюнкцию K . Пусть $\alpha_1, \dots, \alpha_r$ — регулярные точки конъюнкции K , а $\beta_1, \dots, \beta_r$ — соответствующие им точки из определения регулярности. Тогда ДНФ должна содержать конъюнкции $K_1 \in \Pi(\beta_1) \subseteq \Pi(\alpha_1), \dots, K_r \in \Pi(\beta_r) \subseteq \Pi(\alpha_r)$, которые покрывают K , и из нее можно уда-

лить конъюнкцию K , следовательно, ДНФ не является тупиковой.

Достаточность. K нерегулярна, следовательно, в K есть нерегулярная точка γ . Для любого β такого, что $f(\beta) = 1$ и $K(\beta) = 0$, выполняется $\Pi(\beta) \not\subseteq \Pi(\gamma)$. Рассмотрим следующую ДНФ. Для каждой точки β , такой, что $K(\beta) = 0$, возьмем конъюнкцию из пучка $\Pi(\beta)$, не принадлежащую пучку $\Pi(\gamma)$. Добавим ко всем взятым конъюнкциям конъюнкцию K . Полученная ДНФ состоит из простых импликант. Удалить конъюнкцию K из нее нельзя, так как она единственная равна единице в точке γ . Следовательно, K входит во все тупиковые конъюнкции, получаемые из данной ДНФ. $\square$

1.10 Сокращенная ДНФ для монотонных функций

Будем говорить, что для булевых наборов $\mathbf{x}, \mathbf{y}$ выполняется соотношение $\mathbf{x} \leq \mathbf{y}$, если для любого i справедливо неравенство $x_i \leq y_i$. Булева функция f называется *монотонной*, если для любой пары ее аргументов $\mathbf{x}, \mathbf{y}$, таких, что $\mathbf{x} \leq \mathbf{y}$, справедливо неравенство $f(\mathbf{x}) \leq f(\mathbf{y})$.

Т е о р е м а 6. *Сокращенная ДНФ монотонной функции не содержит отрицаний переменных, является тупиковой и минимальной.*

Д о к а з а т е л ь с т в о. От противного. Пусть $K\bar{x}_i$ — простая импликанта монотонной функции f . Тогда $K\bar{x}_i \leq f$. В силу монотонности f имеем $Kx_i \leq f$. Но тогда K — импликанта f . Последнее противоречит тому, что $K\bar{x}_i$ — простая импликанта.

Пусть $K = x_{i_1} \cdot \dots \cdot x_{i_s}$ — произвольная простая импликанта монотонной функции f . Тогда точка $\mathbf{x}$ с компонентами $x_{i_1} = \dots = x_{i_s} = 1$, а остальными, равными нулю, является ядровой точкой K . Поэтому все простые импликанты ядровые, что доказывает оставшуюся часть теоремы. $\square$

1.11 Максимальная длина и сложность сокращенной ДНФ

Рассмотрим для $n = 3k$ булеву функцию $f(x_1, \dots, x_n)$, задаваемую соотношением

$$f(x_1, \dots, x_{3k}) = 1 \iff k \leq x_1 + \dots + x_{3k} \leq 2k. \quad (1.1)$$

Рассмотрим произвольную элементарную конъюнкцию $K = x_{i_1} \cdot \dots \cdot x_{i_s} \cdot \bar{x}_{j_1} \cdot \dots \cdot \bar{x}_{j_t}$. Если $s < k$, то при $x_{i_1} = \dots = x_{i_s} = 1$ и остальных компонентах набора $\mathbf{x}$, равных нулю, $K(\mathbf{x}) = 1$, а $f(\mathbf{x}) = 0$, поэтому K не является допустимой конъюнкцией. Аналогично, если $t < k$, то при $x_{j_1} = \dots = x_{j_t} = 0$ и остальных компонентах набора $\mathbf{x}$, равных единице, $K(\mathbf{x}) = 1$, а $f(\mathbf{x}) = 0$, поэтому K не является допустимой конъюнкцией. Если же $s = t = k$, то в силу соотношения (1.1) K является

допустимой конъюнкцией, а следовательно, и простой импликантой (см. рис. 1.3, где набор α и β имеют k общих нулевых компонент и k общих единичных компонент, при этом несовпадающие компоненты наборов в α принимают значение единица, а в β — ноль). Общее число простых импликант функции $f(x_1, \dots, x_n)$ равно

$$\binom{n}{\frac{n}{3}} \cdot \binom{\frac{2n}{3}}{\frac{n}{3}} = \frac{n!}{\left(\left(\frac{n}{3}\right)!\right)^3} \sim \frac{3\sqrt{3}}{2\pi} \cdot \frac{3^n}{n}.$$

Таким образом, функция $f(x_1, \dots, x_n)$ имеет сокращенную ДНФ, длина и сложность которых асимптотически равны $\frac{3\sqrt{3}}{2\pi} \cdot \frac{3^n}{n}$ и $\frac{\sqrt{3}}{\pi} \cdot 3^n$ соответственно.

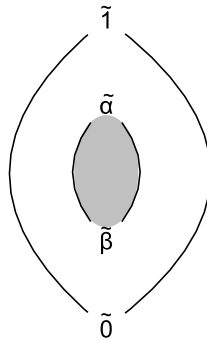


Рис. 1.3

Вопросы и задания.

1. Дайте определение ДНФ сумма тупиковых (ДНФ ΣT) функции.

2. Сформулируйте:

- а) первую теорему Ю.И.Журавлева (критерий вхождения конъюнкции в ДНФ сумма тупиковых);
- б) теорему о сокращенной ДНФ монотонной функции;
- в) оценку максимальной длины и сложности сокращенной ДНФ функции.

3*. Приведите пример функции, у которой ДНФ сумма тупиковых и сумма минимальных различны.

1.12 Алгоритм построения всех тупиковых покрытий матрицы

Матрицы, элементами которых являются только нули и единицы, назовем $(0, 1)$ -матрицами. Покрытием $(0, 1)$ -матрицы $A_{m \times n}$ называется множество ее строк $B = \{\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_k}\}$, такое, что для любого столбца матрицы $A_{m \times n}$ с номером s найдется строка $\mathbf{a}_j$, принадлежащая B , такая, что элемент $a_{j,s}$ матрицы $A_{m \times n}$ равен единице. Покрытие называется *тупиковым*, если любое его собственное подмножество не является *покрытием*. Предполагается, что матрица не имеет нулевых столбцов.

Опишем алгоритм построения всех тупиковых покрытий. Вначале построим КНФ, состоящую из n скобок, соответствующих столбцам матрицы $A_{m \times n}$. Переменная $\mathbf{a}_i$, входит в качестве слагаемого j -й скобки КНФ тогда и только тогда, когда

$a_{i,j} = 1$. Затем перемножаем скобки КНФ и преобразуем полученную дизъюнкцию конъюнкций по правилам $x \cdot x = x$ и $K_1 K_2 \vee K_1 = K_1$. Получим ДНФ переменных $\mathbf{a}_1, \dots, \mathbf{a}_m$.

Т е о р е м а 7. *Каждое слагаемое построенной ДНФ соответствует тупиковому покрытию матрицы $A_{m \times n}$. Построенная по заданному алгоритму ДНФ содержит слагаемые, соответствующие каждому из тупиковых покрытий.*

Д о к а з а т е л ь с т в о. Пусть $B = \{\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_k}\}$ — произвольное тупиковое покрытие матрицы $A_{m \times n}$. Тогда из описания алгоритма следует, что в каждую скобку КНФ войдет некоторая переменная $\mathbf{a}_{i_j} \in B$. При раскрытии скобок получится слагаемое, содержащее только переменные из B . Если при этом данное слагаемое будет содержать не все переменные, то по построению КНФ это будет означать наличие покрытия — собственного подмножества множества B , что противоречит его тупиковости. Слагаемые, соответствующие нетупиковым покрытиям, поглощаются по правилу $K_1 K_2 \vee K_1 = K_1$ слагаемыми, соответствующими тупиковым. $\square$

1.13 Неразрешимость нахождения локальными алгоритмами ДНФ сумма минимальных

ДНФ сумма тупиковых функции f называется дизъюнкция всех тех различных простых импликант этой функции,

которые входят хотя бы в одну тупиковую ДНФ функции f . ДНФ сумма минимальных функции f называется дизъюнкция всех тех различных простых импликант этой функции, которые входят хотя бы в одну минимальную ДНФ функции f .

Рассмотрим таблицу Квайна сокращенной ДНФ произвольной функции $f(x_1, \dots, x_n)$. Опишем алгоритм построения по ней некоторой иной ДНФ функции. Рассмотрим граф G с множеством вершин – объединением множеств простых импликант K_i и наборов $\tilde{\alpha}_j$ – единиц функции $f(x_1, \dots, x_n)$. Ребрами (единичной длины) в графе G соединены только вершины, соответствующие конъюнкциям K_i , с вершинами, соответствующими наборам $\tilde{\alpha}_j$, при условии $K_i(\tilde{\alpha}_j) = 1$. Графом $G_k(K_i)$ называется подграф графа G с множеством вершин, удаленных от K_i в G на расстояние не больше k , и множеством ребер, лежащих на путях, содержащих K_i , длины не больше k .

Локальный алгоритм (с параметром k)

1. Приписать всем конъюнкциям K_i пометки "—" и упорядочить их в список произвольным образом.
2. Если список пуст — закончить работу. Иначе, взять первую конъюнкцию K_i в списке и построить для нее граф $G_k(K_i)$.
3. По графу $G_k(K_i)$ (можно учитывать пометки конъюнкций) принять одно из трех решений: взять K_i в ДНФ,

пометить K_i единицей и удалить K_i из списка; не брать K_i в ДНФ, пометить K_i нулем и удалить K_i из списка; отправить K_i в конец списка с пометкой "–".

4. Перейти к пункту 2.

Рассмотрим сокращенную ДНФ функции f , равную $K_1 \vee K_2 \vee \dots \vee K_n$, в трех случаях

1. Множество единиц функции f_1^n состоит из $n + 1$ единиц, представляющих собой цепочку нечетной длины (n нечетно).
2. Множество единиц функции f_2^n состоит из $n + 1$ единиц, представляющих собой цепочку четной длины (n четно).
3. Множество единиц функции f_3^n состоит из n единиц, представляющих собой цикл.

В первом случае функция f имеет единственную минимальную ДНФ $K_1 \vee K_3 \vee K_5 \vee \dots \vee K_n$. В третьем случае – две: $K_1 \vee K_3 \vee K_5 \vee \dots \vee K_{n-1}$ и $K_2 \vee K_4 \vee K_6 \vee \dots \vee K_n$. Вторая же функция имеет в качестве минимальных все ДНФ вида $K_1 \vee K_3 \vee \dots \vee K_{2m-1} \vee K_{2m} \vee \dots \vee K_n$. Поэтому ДНФ сумма минимальных для первой функции равна $K_1 \vee K_3 \vee K_5 \vee \dots \vee K_n$, а для второй и третьей – $K_1 \vee K_2 \vee \dots \vee K_n$.

Т е о р е м а 8. (вторая теорема Ю.И.Журавлева [2]).
Не существует такого k , что локальный алгоритм с пара-

метром k строит ДНФ сумма минимальных произвольной функции.

Д о к а з а т е л ь с т в о. Пусть на вход алгоритма подано множество импликант неизвестной функции f , равной f_1^n, f_2^n или f_3^n . Пусть первой конъюнкцией в списке является K_i . Заметим, что если $k < i < n - k + 1$, то для функций f_1^n и f_2^n граф $G_k(K_i)$ будет представлять собой цепь из $2k$ звеньев с вершиной K_i в середине. Таким же он будет при $k < \frac{n}{2}$ для функции f_3^n . Пусть i четно. Тогда мы должны поставить отметку 1 или "—" для функций f_2^n и f_3^n и отметку 0 или "—" для функции f_1^n . Поэтому на третьем шаге алгоритма мы вынуждены ставить отметку "—" и заносить конъюнкцию в конец списка. Но для функции f_3^n граф $G_k(K_i)$ одинаков для всех конъюнкций, и, следовательно, алгоритм не изменит ни одной пометки. $\square$

Вопросы и задания.

1. Дайте определения $(0, 1)$ -матрицы и ее покрытия.
2. Сформулируйте:
 - а) теорему о построении всех тупиковых ДНФ функции;
 - б) вторую теорему Ю.И.Журавлева.
3. Опишите локальный алгоритм (с параметром k).

- 4*. Получите оценку количества переменных функции, на которой гарантированно не работает локальный алгоритм с параметром k .

Глава 2

Градиентный алгоритм

2.1 Оценка сложности покрытия градиентного алгоритма

Задача о покрытии ставится следующим образом: для заданной прямоугольной матрицы из нулей и единиц требуется найти множество строк, такое, что на их пересечении с любым столбцом имеется хотя бы одна единица. При этом стремятся найти покрытие с как можно меньшим числом строк. Градиентный алгоритм для задачи о покрытии устроен следующим образом: на каждом шаге из текущей матрицы выбирается строка с наибольшим числом единиц и удаляется вместе со всеми столбцами, в которых она равна единице. Процесс продолжается от исходной матрицы до пустой. Следующий пример показывает, что градиентный алгоритм не дает гарантированно ни минимального, ни тупикового покрытия.

Пример неудачи градиентного алгоритма.

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Минимальным покрытием является $\{2, 3, 4\}$. Градиентный же алгоритм строит покрытие $\{1, 2, 3, 4\}$. Тем не менее градиентный алгоритм удобен тем, что при определенных легко проверяемых параметрах матрицы (количество столбцов и минимальная доля единиц в столбце) имеется гарантированная верхняя оценка сложности полученного покрытия.

Т е о р е м а 9. Пусть дана $(0, 1)$ -матрица $A_{m \times n}$. Пусть каждый столбец матрицы $A_{m \times n}$ содержит не менее γt единиц. Тогда градиентное покрытие содержит не более $1 - \frac{\ln n}{\ln(1-\gamma)}$ строк.

Д о к а з а т е л ь с т в о. Из условия теоремы следует наличие не менее $\gamma t n$ единиц в матрице $A_{m \times n}$. Следовательно, в этой матрице имеется строка с не менее чем γn единицами. После применения шага градиентного алгоритма в новой матрице останется не более $(1 - \gamma)n$ столбцов. При этом, так как в оставшихся столбцах в удаленных строках стояли нули, доля единиц в них только увеличится. Таким образом, получаемая на каждом очередном шаге градиентного алгоритма матрица будет иметь не менее $\gamma t'$ единиц в каждом ее столбце, где

m' — количество строк матрицы. Если для числа шагов градиентного алгоритма t выполнено неравенство $(1 - \gamma)^t \leq \frac{1}{n}$, то непокрытым остается максимум один столбец. При этом $t \leq -\frac{\ln n}{\ln(1-\gamma)}$. Отсюда получаем утверждение теоремы. $\square$

2.2 Пример модификации градиентного алгоритма

Будем называть булеву функцию $f(x_1, \dots, x_n)$ *линейной*, если она представима в виде $f(x_1, \dots, x_n) = \alpha_0 \oplus \alpha_1 x_1 \oplus \dots \oplus \alpha_n x_n$, где $\alpha_i \in \{0, 1\}$, $i = \overline{0, n}$.

Рассмотрим следующую задачу. Будем говорить, что *частичная булева функция f порождает заданную линейную булеву функцию g* , если существует такая подобласть области определения функции f , что функция g является единственной линейной функцией, совпадающей с f на этой подобласти. Функция f (возможно, частичная, т.е. не всюду определенная) называется *универсальной* для класса булевых линейных функций n переменных, если она порождает все функции заданного класса. Универсальные функции для класса линейных булевых функций впервые рассматривались в работе [3]. В этой работе построена конструктивная линейная верхняя оценка на минимальную мощность области определения универсальной функции для класса линейных булевых функций.

Л е м м а 2. [3] Размер области определения универсальной

функции для класса булевых линейных функций n переменных не меньше, чем $2n + 2$.

Д о к а з а т е л ь с т в о. При задании линейной булевой функции n переменных в m точках ее коэффициенты определяет система из m линейных уравнений и $n + 1$ неизвестных. Для однозначной разрешимости требуется выполнение неравенства $m \geq n + 1$. Функция тождественный ноль (тождественная единица) порождается минимум $n + 1$ нулями (единицами), откуда следует утверждение леммы. $\square$

В работе [4] нижняя оценка $2n + 2$ леммы 2 поднята до $2^{\frac{1}{6}} \cdot n$.

Л е м м а 3. *Количество упорядоченных пар различных линейных булевых функций n переменных не превосходит 4^{n+1} .*

Л е м м а 4. *Две различных булевых линейных функции n переменных не совпадают минимум в 2^{n-1} точке.*

Д о к а з а т е л ь с т в о. Пусть $g_1(x_1, \dots, x_n)$ и $g_2(x_1, \dots, x_n)$ — различные линейные функции. Тогда $g_1(x_1, \dots, x_n) \oplus g_2(x_1, \dots, x_n)$ — не равная тождественно нулю линейная функция, имеющая не менее 2^{n-1} единиц. ([24], с. 59) $\square$

Т е о р е м а 10. *Начиная с некоторого n , существует частичная универсальная функция для класса линейных булевых функций с областью определения, ограниченной величиной $O(n)$.*

Д о к а з а т е л ь с т в о. Задача построения универсальной функции эквивалентна задаче построения покрытия

$(0-1)$ -матрицы, строкам которой соответствуют всевозможные значения функции на всех наборах ($2 \cdot 2^n$ строк), столбцам — упорядоченные пары различных линейных функций ($2^{n+1}(2^{n+1}-1)$ пар). При этом единица соответствует ситуации, когда значение функции на наборе в строке совпадает со значением первой линейной функции пары и не совпадает со значением второй. В покрытии, соответствующем универсальной функции, дополнительно запрещается наличие двух строк, соответствующих различным значениям функции в одной точке. Построим покрытие матрицы градиентным алгоритмом ([5], с. 136,137). На каждом шаге будем выбирать строку с наибольшим числом единиц и удалять ее из матрицы вместе со всеми столбцами, в пересечении с которыми в ней стоят единицы. По лемме 4 изначально в каждом столбце матрицы не менее 2^{n-1} единиц. В отличие от классической задачи о покрытии в силу вышесказанного вместе со строкой мы вынуждены удалять еще одну, соответствующую противоположному значению функции в данной точке. При этом в худшем случае мы можем потерять одну единицу в любом непокрытом столбце. После того, как мы взяли t строк, количество единиц в оставшихся столбцах будет не меньше, чем $2^{n-1} - t$. Пусть M_t — количество оставшихся столбцов после t шагов. Тогда число единиц в матрице после t шагов не меньше $(2^{n-1} - t)M_t$. Количество строк этой матрицы равно $2(2^n - t)$, поэтому в ней

есть хотя бы одна строка с не менее чем

$$\frac{(2^{n-1} - t)M_t}{2(2^n - t)}$$

единицами. Таким образом,

$$M_{t+1} \leq M_t \left(1 - \frac{2^{n-1} - t}{2(2^n - t)} \right). \quad (2.1)$$

Множитель $\left(1 - \frac{2^{n-1} - t}{2(2^n - t)} \right)$ растёт с ростом t . Поэтому, если положить значение $t = c \cdot n$, где c — некоторая константа, и многократно применить неравенство (2.1), получим

$$M_t \leq 4^{n+1} \cdot \left(1 - \frac{2^{n-1} - c \cdot n + 1}{2 \cdot (2^n - c \cdot n + 1)} \right)^{c \cdot n}. \quad (2.2)$$

Если при этом выполнено неравенство

$$c \cdot n \leq 2^{n-2},$$

то из неравенства (2.2) следует

$$M_t \leq 4 \cdot \left(4 \cdot \left(1 - \frac{2^{n-1} - 2^{n-2} + 1}{2(2^n - 2^{n-2} + 1)} \right)^c \right)^n.$$

Из последнего неравенства в силу того, что

$$\frac{2^{n-1} - 2^{n-2} + 1}{2^n - 2^{n-2} + 1} \geq \frac{1}{3},$$

получим

$$M_t \leq 4 \cdot \left(4 \cdot \left(\frac{5}{6} \right)^c \right)^n.$$

При $\left(4 \cdot \left(\frac{5}{6} \right)^c \right) < 1$ и достаточно большом n величина $M_{c \cdot n}$ равна нулю. $\square$

З а м е ч а н и е 1. Оценка константы с теоремы как сколь угодно мало превосходящей $\frac{\ln 4}{\ln 1.2} < 7.604$ намного хуже прямой оценки $c = 3.2$ работы [3].

Вопросы и задания.

1. Дайте определения следующих понятий:
 - а) функция f порождает линейную булеву функцию g ;
 - б) универсальная функция для класса булевых линейных функций n переменных.
2. Сформулируйте следующие теоремы:
 - а) об оценке числа строк градиентного покрытия;
 - б) о существовании частичной универсальной функции для класса булевых линейных функций.

Глава 3

Сложность реализации булевых функций СФЭ и КС

3.1 Нижняя оценка функции Шеннона для СФЭ

Схемой функциональных элементов (СФЭ) в базисе $B = \{f_1, \dots, f_m, \dots\}$ называется ориентированный граф, не имеющий ориентированных циклов (контуров) и обладающий следующими свойствами.

1. Вершинам с полустепенью захода ноль приписаны попарно различные переменные; эти вершины называются входами схемы.
2. Остальным вершинам графа приписаны функции из B , причем полустепень захода вершины должна равняться размерности функции, и входы считаются упорядоченными.

3. Произвольные вершины объявляются множеством выходов схемы.

Функционирование СФЭ определяется естественным образом от входов к выходам: в каждой вершине реализуется соответствующая функция элемента из базиса от его входов. Базис $B_0 = \{\&, \vee, \neg\}$ называется *элементарным*. Конъюнкция и дизъюнкция в нем двухвходовые; эти элементы называются соответственно *конъюнктором* и *дизъюнктором*.

Сложностью $L(s)$ схемы функциональных элементов s называется число функциональных элементов в s . Сложностью $L(f)$ булевой функции $f(x_1, \dots, x_n)$ называется минимальная сложность СФЭ, реализующей данную функцию. Функция Шеннона для сложности СФЭ $L(n) = \max_{f \in P_2^n} L(f)$. При этом к символу L обычно добавляется нижний индекс, обозначающий базис.

Пусть задан связный граф $G = (V, E)$. *Остовным деревом* для него называется любое дерево с множеством вершин V и множеством ребер — подмножеством множества E .

Для получения нижней оценки функции Шеннона для СФЭ оценим сверху количество различных СФЭ заданной сложности и тем самым покажем, что при определенных ограничениях на сложность количество схем будет меньше, чем общее число булевых функций n переменных.

Пусть заданы сложность L и число переменных n . Схему можно считать связным графом с наличием ориентированно-

го пути из любой вершины к выходу (неприводимой). Сначала зададим произвольное остовное дерево этого графа. Это можно сделать (в силу соотношения между числом вершин и числом ребер дерева и оценки числа деревьев с заданным числом ребер) максимум 4^{L+n-1} различными способами. После этого зададим все L элементов. Поскольку у нас для каждого элемента имеется не более трех вариантов (конъюнктор, дизъюнктор и отрицание), общее число вариантов не превысит 3^L . Для задания переменных на входах у нас есть не более $n!$ вариантов. Затем соединим вторым входом те элементы, у которых этот вход не определен деревом (и которые не являются отрицаниями). Для каждого элемента при этом имеется не более $L + n - 1$ возможностей. Общее число вариантов не превосходит $(L + n - 1)^L$. Итого количество различных схем сложности не выше L от n переменных не превосходит

$$\sum_{l=0}^L 4^{l+n-1} \cdot 3^l \cdot n! \cdot (l + n - 1)^l.$$

Заметим, что каждое последующее слагаемое этой суммы минимум в 12 раз больше предыдущего, поэтому вся эта сумма оценивается сверху суммой геометрической прогрессии

$$\frac{12}{11} \cdot 4^{L+n-1} \cdot 3^L \cdot n! \cdot (L + n - 1)^L. \quad (3.1)$$

Т е о р е м а 11. Пусть для некоторого $\delta > 0$ и всех достаточно больших n выполняется соотношение $L \leq (1 - \delta) \frac{2^n}{n}$. Тогда при $n \rightarrow \infty$ для любого сколь угодно малого $\varepsilon > 0$, начиная с некоторого n , количество различных неприводимых

схем сложности не выше L от n переменных не превосходит $\varepsilon \cdot 2^{2^n}$.

Д о к а з а т е л ь с т в о. Прологарифмируем выражения (3.1) и $\varepsilon \cdot 2^{2^n}$ по основанию 2. Логарифм выражения (3.1) равен

$$\log_2 \frac{12}{11} + 2(L+n-1) + L \cdot \log_2 3 + \log_2 n! + L \cdot \log_2 (L+n-1). \quad (3.2)$$

Первые четыре слагаемых являются бесконечно малыми при $n \rightarrow \infty$ относительно 2^n (четвертое в силу формулы Стирлинга $n! \sim \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$ при $n \rightarrow \infty$). Оценим пятое слагаемое:

$$\begin{aligned} L \cdot \log_2 (L+n-1) &\leq (1-\delta) \frac{2^n}{n} \log_2 \left((1-\delta) \frac{2^n}{n} + n-1 \right) \lesssim \\ &\lesssim (1-\delta) \frac{2^n}{n} n \sim (1-\delta) 2^n. \end{aligned}$$

□

Таким образом, доказано, что:

- 1) $L_{B_0}(n) \gtrsim \frac{2^n}{n}$ при $n \rightarrow \infty$;
- 2) какую бы фиксированную долю ε и сколь бы малое число δ мы ни взяли, при достаточно большом n сложности $(1-\delta) \frac{2^n}{n}$ не хватит на реализацию ε -й доли всех функций n переменных.

Вопросы и задания.

1. Дайте определения следующих понятий:

- а) схема функциональных элементов (СФЭ) в базисе $B = \{f_1, \dots, f_m, \dots\}$;
 - б) элементарный базис, конъюнктор, дизъюнктор;
 - в) сложность $L(s)$ схемы функциональных элементов s ;
 - г) сложность $L(f)$ булевой функции $f(x_1, \dots, x_n)$;
 - д) функция Шеннона для сложности СФЭ $L(n)$;
 - е) остовное дерево.
2. Сформулируйте теорему о количестве неприводимых схем сложности не выше L от n переменных.
- 3*. Постройте схему в элементарном базисе, входами которой являются три переменные, а выходами — их отрицания, содержащую только два элемента отрицания.

3.2 Универсальный многополюсник. Дешифратор

Универсальным многополюсником U_n назовем схему функциональных элементов с n входами $x_1, \dots, x_n$ и 2^{2^n} выходами, на которых реализуются все функции переменных $x_1, \dots, x_n$.

Т е о р е м а 12. Сложность универсального многополюсника $L_{B_0}(U_n)$ равна $2^{2^n} - n$.

Д о к а з а т е л ь с т в о. Нижняя оценка. Различным функциям должны соответствовать разные вершины схемы.

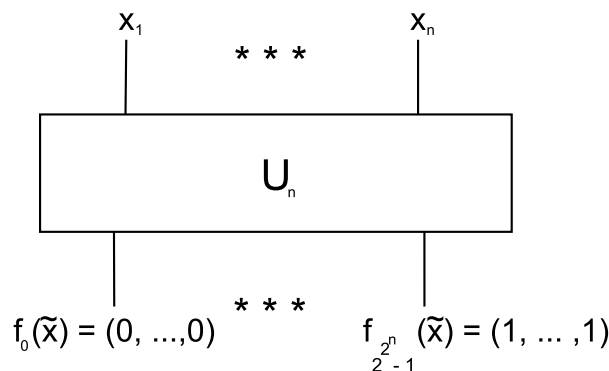


Рис. 3.1

Сложность 0 имеют только n переменных. Отсюда получаем, что $L(U_n) \geq 2^{2^n} - n$.

Верхняя оценка получается методом математической индукции по n . Базис $n = 1$. При помощи одного инвертора получаем отрицание переменной x_1 , при помощи одного конъюнктора получаем константу ноль по формуле $x_1 \& \bar{x}_1$ и при помощи одного дизъюнктора — константу единица по формуле $x_1 \vee \bar{x}_1$. Таким образом, $L_{B_0}(U_1) = 3 \leq 2^{2^1} - 1 = 3$.

Шаг индукции. Пусть $L_{B_0}(U_n) \leq 2^{2^n} - n$. К схеме, реализующей U_n , добавим вход x_{n+1} ; инвертируя его, получим за один элемент функцию $\bar{x}_{n+1}$. Далее при помощи одного конъюнктора реализуем каждую из функций вида $x_{n+1} \& g(x_1, \dots, x_n)$ и $\bar{x}_{n+1} \& g(x_1, \dots, x_n)$. После этого при помощи ровно одного дизъюнктора реализуем каждую из еще не реализованных

функций при помощи разложения по последней переменной

$$f(x_1, \dots, x_{n+1}) = \bar{x}_{n+1} \& f(x_1, \dots, x_n, 0) \vee x_{n+1} \& f(x_1, \dots, x_n, 1).$$

Для реализации каждой функции, зависящей от переменной x_{n+1} , кроме самой этой переменной, использован ровно один элемент. Таким образом, все функции, кроме переменных, реализованы добавлением одного элемента. Общее количество функций $n + 1$ переменной равно $2^{2^{n+1}}$. Шаг индукции обоснован. $\square$

Дешифратор. Дешифратором D_n называется схема функциональных элементов с n входами $x_1, \dots, x_n$ и 2^n выходами, на которых реализуются все конъюнкции переменных $x_1, \dots, x_n$ и их отрицаний, содержащие литералы, соответствующие всем переменным. Дешифратор можно строить, последовательно добавляя переменные.

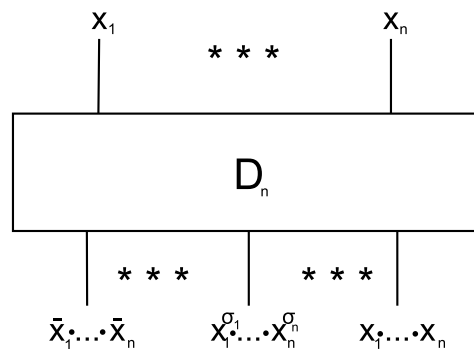


Рис. 3.2

У т в е р ж д е н и е 1. *Сложность дешифратора $L_{B_0}(D_n)$ не превосходит $2^{n+1} + n - 4$.*

Д о к а з а т е л ь с т в о. По индукции. Базис $n = 1$. Дешифратор одной переменной реализуется при помощи единственного отрицания $2^{1+1} + 1 - 4 = 1$.

Шаг индукции. Пусть реализован дешифратор D_n со сложностью не более $2^{n+1} + n - 4$. Добавим к схеме вход x_{n+1} , реализуем отрицание $\bar{x}_{n+1}$. После этого каждый выход дешифратора D_n соединим через конъюнкцию как с переменной x_{n+1} , так и с ее отрицанием $\bar{x}_{n+1}$. Получим дешифратор D_{n+1} . При этом выполняется соотношение $L_{B_0}(D_{n+1}) \leq L_{B_0}(D_n) + 2^{n+1} + 1$, откуда по предположению индукции $L_{B_0}(D_{n+1}) \leq 2^{n+1} + n - 4 + 2^{n+1} + 1 = 2^{n+1+1} + (n + 1) - 4$. $\square$

Т е о р е м а 13. $L_{B_0}(D_n) = 2^n + O(2^{\frac{n}{2}})$ при $n \rightarrow \infty$.

Д о к а з а т е л ь с т в о. Нижняя оценка тривиальна. Для получения верхней оценки построим дешифратор D_n следующим образом: соединим всевозможные пары выходов двух дешифраторов D_k и D_{n-k} через конъюнкции. При этом

$$L_{B_0}(D_n) \leq 2^n + L_{B_0}(D_k) + L_{B_0}(D_{n-k}).$$

Положив $k = \lfloor \frac{n}{2} \rfloor$ в утверждении 1, получим

$$L_{B_0}(D_n) \leq 2^n + 2 \cdot 2^{\frac{n}{2}} + 2 \cdot 2 \cdot 2^{\frac{n}{2}} + n.$$

Из последнего неравенства вытекает утверждение теоремы. $\square$

Вопросы и задания.

1. Дайте определения следующих понятий:
 - а) универсальный многополюсник;
 - б) дешифратор.
2. Сформулируйте следующие теоремы:
 - а) о сложности универсального многополюсника в базисе B_0 ;
 - б) о верхней оценке сложности дешифратора в базисе B_0 ;
 - в) об асимптотической оценке сложности дешифратора в базисе B_0 .

3.3 Метод Шеннона для синтеза СФЭ в базисе $\{\&, \vee, \neg\}$

Т е о р е м а 14. Для функции Шеннона $L_{B_0}(n)$ выполняется соотношение $L_{B_0}(n) = O(\frac{2^n}{n})$ при $n \rightarrow \infty$.

Д о к а з а т е л ь с т в о. (Метод Шеннона.) Для произвольной булевой функции $f(x_1, \dots, x_n)$ воспользуемся ее разложением по первым k переменным

$$f(x_1, \dots, x_n) = \bigvee_{(\sigma_1, \dots, \sigma_k)} x_1^{\sigma_1} \& \dots \& x_k^{\sigma_k} \& f(\sigma_1, \dots, \sigma_k, x_{k+1}, \dots, x_n). \quad (3.3)$$

При этом для реализации произвольной функции n переменных достаточно $2^k - 1$ дизъюнкторов, одного дешифратора D_k , 2^k конъюнкторов и одного универсального многополюсника U_{n-k} . Таким образом,

$$L_{B_0}(n) \leq 2^k - 1 + 2^k + L_{B_0}(D_k) + L_{B_0}(U_{n-k}) \leq 3 \cdot 2^k + C \cdot 2^{\frac{k}{2}} + 2^{2^{n-k}}. \quad (3.4)$$

Подберем параметр k так, чтобы $2^{2^{n-k}} \leq \frac{2^n}{n^2}$. Положим $k = \lfloor n - \log_2(n - 2 \log_2 n) \rfloor$. При этом из неравенства (3.4) вытекает утверждение теоремы. $\square$

3.4 Метод О.Б. Лупанова для синтеза схем в базисе $\{\&, \vee, \neg\}$

Опишем метод, предложенный О.Б. Лупановым, позволяющий доказать следующее утверждение

Т е о р е м а 15. *Для каждой функции $f(x_1, \dots, x_n)$ можно построить схему $\Sigma_{f(x_1, \dots, x_n)}$ в базисе B_0 так, что при $n \rightarrow \infty$*

$$\max_{f(x_1, \dots, x_n)} L(\Sigma_{f(x_1, \dots, x_n)}) \lesssim \frac{2^n}{n}.$$

Д о к а з а т е л ь с т в о. Вновь воспользуемся разложением, похожим на (3.3), а именно:

$$\begin{aligned} & f(x_1, \dots, x_n) = \\ & = \bigvee_{(\sigma_{k+1}, \dots, \sigma_n)} x_{k+1}^{\sigma_{k+1}} \& \dots \& x_n^{\sigma_n} \& f(x_1, \dots, x_k, \sigma_{k+1}, \dots, \sigma_n). \end{aligned} \quad (3.5)$$

Для доказательства теоремы нужно подобрать параметр k так, чтобы порядок сложности дешифратора $n - k$ переменных и дизъюнкции 2^{n-k} слагаемых был равен $\bar{o}(\frac{2^n}{n})$ в формуле (3.5), но главное, в отличие от метода Шеннона, одновременно реализовать все подфункции $f(x_1, \dots, x_k, \sigma_{k+1}, \dots, \sigma_n)$ с асимптотической сложностью при $n \rightarrow \infty$, равной $\frac{2^n}{n}$. Воспользуемся представлением функции $f(x_1, \dots, x_n)$ в виде прямоугольной таблицы размера $2^k \times 2^{n-k}$, в которой строки соответствуют значениям первых k переменных, а столбцы — последних $n - k$ (см. таблицу). Прямоугольник разрежем на горизонтальные полосы ширины s (последняя полоса имеет ширину s' , $s' \leq s$). Обозначим через p величину, равную $\lceil \frac{2^k}{s} \rceil$, — количество полос. Значения параметров k и s выберем позднее.

На первом этапе реализуем всевозможные короткие столбцы — функции, совпадающие с $f(x_1, \dots, x_k, \sigma_{k+1}, \dots, \sigma_n)$ на одной из полос и равные нулю вне ее. Для этого используем дешифратор первых k переменных порядка 2^k . Общее количество коротких столбцов не превосходит $p \cdot 2^s$. Реализация каждого требует не более $s - 1$ дизъюнкции. Таким образом, если $2^k = \bar{o}(\frac{2^n}{n})$, мы можем оценить сверху сложность подсхемы, строящейся на первом этапе суммой, $p \cdot s \cdot 2^s$ и бесконечно малой относительно $\frac{2^n}{n}$ величины.

На втором этапе реализуем длинные столбцы, соответствующие функциям $f(x_1, \dots, x_k, \sigma_{k+1}, \dots, \sigma_n)$. Общее количество

	0	σ_{k+1}	1	x_{k+1}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_1 \dots x_k$	0	σ_n	1	x_n
$0 \dots 0$				$\left. \vphantom{\begin{matrix} 0 \dots 0 \\ \vdots \\ 0 \dots 0 \end{matrix}} \right\} s$
				$\left. \vphantom{\begin{matrix} 0 \dots 0 \\ \vdots \\ 0 \dots 0 \end{matrix}} \right\} s$
$\sigma_1 \dots \sigma_k$		$f(\tilde{\sigma})$		$\left. \vphantom{\begin{matrix} 0 \dots 0 \\ \vdots \\ 0 \dots 0 \end{matrix}} \right\} s$
$1 \dots 1$				$\left. \vphantom{\begin{matrix} 0 \dots 0 \\ \vdots \\ 0 \dots 0 \end{matrix}} \right\} s'$

функций равно 2^{n-k} , реализация каждой требует $p - 1$ дизъюнкцию. Общая сложность не превосходит $p \cdot 2^{n-k}$. После этого происходит сборка по формуле (3.5). Если $(1 + \delta) \log_2 n \leq k \leq n - (1 + \delta) \log_2 n$ и $\frac{2^k}{s} \rightarrow \infty$ при $n \rightarrow \infty$, то сложность схемы асимптотически не превосходит

$$p \cdot s \cdot 2^s + p \cdot 2^{n-k} \sim \frac{2^k}{s} (2^s \cdot s + 2^{n-k}) = 2^{k+s} + \frac{2^n}{s}.$$

Если положить $k = \lceil 3 \log_2 n \rceil$, а $s = \lceil n - 5 \log_2 n \rceil$, то второе слагаемое в последней формуле асимптотически равно $\frac{2^n}{n}$, а первое является бесконечно малым относительно этой величины. $\square$

Из этой теоремы и нижней оценки функции Шеннона следует

Т е о р е м а 16. Для функции Шеннона $L_{B_0}(n)$ выполняется соотношение $L_{B_0}(n) \sim \frac{2^n}{n}$ при $n \rightarrow \infty$, причем доля функций, таких, что $L_{B_0}(n) \leq (1 - \delta) \frac{2^n}{n}$, стремится к нулю при $n \rightarrow \infty$ для любого $\delta > 0$.

Вопросы и задания.

1. Сформулируйте следующие теоремы:

- а) об асимптотической оценке функции Шеннона $L_{B_0}(n)$ при $n \rightarrow \infty$ и оценке доли функций, таких, что $L_{B_0}(n) \leq (1 - \delta) \frac{2^n}{n}$ при $n \rightarrow \infty$ для любого $\delta > 0$;

- б) о верхней оценке $\max_{f(x_1, \dots, x_n)} L(\Sigma_{f(x_1, \dots, x_n)})$ при $n \rightarrow \infty$ для любой функции в базисе B_0 .

2. Опишите принцип работы:

- а) метода Шеннона для синтеза СФЭ в базисе $\{\&, \vee, \neg\}$;
б) метода О.Б. Лупанова для синтеза схем в базисе $\{\&, \vee, \neg\}$.

3.5 Контактные схемы. Метод Шеннона

Контактной схемой (КС) назовем мультиграф (неориентированный), ребрам которого приписаны литералы, а часть вершин графа выделена и называется *полюсами*.

Считается, что каждое ребро графа — это контакт, реализующий соответствующую функцию проводимости между инцидентными вершинами (переменную или ее отрицание). Ребра, помеченные переменными, называются *замыкающими* контактами, а отрицаниями переменных — *размыкающими*. Контактная схема реализует симметричную функциональную матрицу. Для каждой пары полюсов рассматриваются пути между ними — простые цепи, и вдоль цепей берется конъюнкция всех функций проводимости. Между полюсами реализуется дизъюнкция по всем цепям соответствующих им конъюнкций (функция проводимости).

Пример.

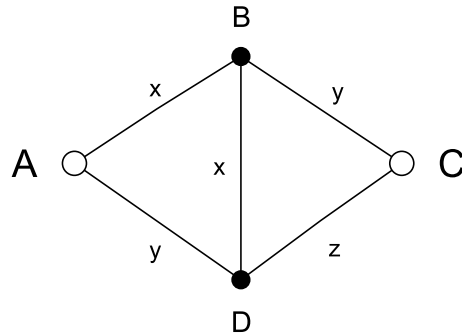


Рис. 3.3

В данном случае имеются два полюса A и C , поэтому можно говорить об одной функции, определяемой путями ABC , ADC , $ABDC$, $ADBC$. При этом реализуется функция

$$xy \vee yz \vee xxz \vee yxy = xy \vee yz \vee xz,$$

медиана переменных x, y, z .

Под сложностью контактной схемы понимается число контактов в ней. Под сложностью функции понимается минимальное количество контактов в реализующей ее схеме. Функция Шеннона определяется как максимум по всем функциям n переменных сложности функций.

Контактным дешифратором порядка n называется контактная схема с $1 + 2^n$ полюсами, в которой между одним полюсом и оставшимися реализуются все конъюнкции n литералов, соответствующих различным переменным. Если проводимость между любыми двумя полюсами из числа 2^n последних

равна нулю, то дешифратор называется *разделительным*.

Рассмотрим корневое бинарное $(k + 1)$ -ярусное дерево. Любая вершина l -го яруса при $l \leq k$ соединена с двумя вершинами $(l + 1)$ -го яруса: с одной — контактом x_l , с другой — контактом $\bar{x}_l$. Полюсами будут корень и все вершины $k + 1$ яруса. Рассматриваются функции проводимости между корнем и листьями.

Л е м м а 5. Построенное дерево сложности $2^{k+1} - 2$ реализует разделительный контактный дешифратор порядка k .

Д о к а з а т е л ь с т в о. Сложность схемы равна $2^k + 2^{k-1} + \dots + 2 = 2^{k+1} - 2$. При прохождении цепи, содержащей два контакта, соединяющих одну вершину с вершинами следующего яруса, получаем нулевую проводимость. Поэтому между корнем и листьями реализуются конъюнкции, причем все. Кроме этого, между листьями проводимость нулевая. $\square$

Назовем *универсальным контактным многополюсником* для множества M контактную схему с k входными и одним выходным полюсами, такую, что для любой функции $f \in M$ найдется такой входной полюс, что между ним и выходным полюсом реализуется эта функция f .

Л е м м а 6. Универсальный контактный многополюсник для функций n переменных имеет сложность не более $2 \cdot 2^{2^n}$.

Д о к а з а т е л ь с т в о. По индукции. Нетрудно построить универсальный многополюсник сложности четыре для функций одной переменной. Воспользуемся разложением по последней переменной.

$$f(x_1, \dots, x_{n+1}) = \bar{x}_{n+1}f(x_1, \dots, x_n, 0) \vee x_{n+1}f(x_1, \dots, x_n, 1).$$

Имея все функции n переменных, при помощи не более двух контактов можно реализовать любую функцию $n + 1$ переменной. $\square$

Т е о р е м а 17. *Для функции Шеннона $L(n)$ сложности реализации булевых функций контактными схемами при $n \rightarrow \infty$ выполняется соотношение $L(n) = O(\frac{2^n}{n})$.*

Д о к а з а т е л ь с т в о. (Метод Шеннона [9].) Для произвольной булевой функции $f(x_1, \dots, x_n)$ воспользуемся ее разложением по первым k переменным:

$$f(x_1, \dots, x_n) = \bigvee_{(\sigma_1, \dots, \sigma_k)} x_1^{\sigma_1} \& \dots \& x_k^{\sigma_k} \& f(\sigma_1, \dots, \sigma_k, x_{k+1}, \dots, x_n).$$

При этом для реализации произвольной функции n переменных достаточно соединить дешифратор k переменных, получаемый по лемме 5, с универсальным многополюсником, отождествив лист дерева дешифратора с проводимостью $x_1^{\sigma_1} \& \dots \& x_k^{\sigma_k}$ с вершиной многополюсника, соответствующего функции $f(\sigma_1, \dots, \sigma_k, x_{k+1}, \dots, x_n)$. Нулевая проводимость между листьями в дереве дешифратора гарантирует отсут-

ствие лишних проводимостей в контактной схеме. При этом

$$L(n) \leq 2^{k+1} - 2 + 2 \cdot 2^{2^{n-k}}. \quad (3.6)$$

Подберем параметр k так, чтобы выполнялось неравенство $2^{2^{n-k}} \leq \frac{2^n}{n^2}$. Положим $k = \lceil n - \log_2(n - 2 \log_2 n) \rceil$. При этом из неравенства (3.6) вытекает утверждение теоремы. $\square$

3.6 Нижняя оценка функции Шеннона для КС

Оценим сверху количество различных двухполюсных контактных схем заданной сложности и тем самым покажем, что при определенных ограничениях схем будет меньше, чем булевых функций n переменных.

Пусть задана сложность L и число переменных n . Обозначим через $N_{\text{КС}}(L, n)$ количество различных схем сложности не выше L от n переменных. Схему можно считать связным графом. Связный граф с L ребрами имеет не более $L + 1$ вершин ([10], с. 29). Мы имеем не более $\frac{(L+1)L}{2}$ способов выбора пар вершин графа — ребер, соответствующих каждому контакту схемы. У нас имеется $\frac{(L+1)L}{2}$ вариантов выбора пары полюсов и $(2n)^L$ вариантов выбора типов контактов (n переменных и 2 варианта: замыкающие и размыкающие). Итого

$$N_{\text{КС}}(L, n) \leq \sum_{l=1}^L (n(l+1)l)^l \cdot (l+1)l.$$

Заметим, что каждое последующее слагаемое этой суммы минимум в 2 раза больше предыдущего, поэтому вся эта сумма оценивается сверху выражением

$$N_{\text{КС}}(L, n) \leq 2 \cdot (n(L+1)L)^L \cdot (L+1)L. \quad (3.7)$$

Т е о р е м а 18. Пусть для некоторого $\delta > 0$ и всех достаточно больших n выполняется соотношение $L \leq \left(\frac{1}{2} - \delta\right) \frac{2^n}{n}$. Тогда при $n \rightarrow \infty$ для любого сколь угодно малого $\varepsilon > 0$, начиная с некоторого n , количество различных контактных схем сложности не выше L от n переменных не превосходит $\varepsilon \cdot 2^{2^n}$.

Д о к а з а т е л ь с т в о. Прологарифмируем выражения (3.7) и $\varepsilon \cdot 2^{2^n}$ по основанию 2. Логарифм выражения (3.7) равен

$$1 + L \log_2 n + L \log_2(L^2 + L) + \log_2(L^2 + L). \quad (3.8)$$

Все слагаемые, кроме третьего, являются бесконечно малыми при $n \rightarrow \infty$ по сравнению с 2^n . Оценим третье слагаемое:

$$\begin{aligned} L \log_2(L^2 + L) &\sim L \log_2(L^2) \leq \\ &\leq \left(\left(\frac{1}{2} - \delta \right) \frac{2^n}{n} \right) \cdot 2 \cdot \left(\log_2 \left(\frac{1}{2} - \delta \right) + n - \log_2 n \right) \lesssim \\ &\lesssim (1 - 2 \cdot \delta) \frac{2^n}{n} n \sim (1 - 2 \cdot \delta) 2^n. \end{aligned}$$

□

Таким образом, доказано, что

- 1) $L(n) \gtrsim \frac{1}{2} \cdot \frac{2^n}{n}$ при $n \rightarrow \infty$;
- 2) какую бы фиксированную долю ε и сколь бы малое число δ мы ни взяли, при достаточно большом n сложности $(\frac{1}{2} - \delta) \frac{2^n}{n}$ не хватит на реализацию ε -й доли всех функций.

Вопросы и задания.

1. Дайте определения следующих понятий:

- а) контактная схема (КС) и её полюс;
- б) сложность контактной схемы;
- в) сложность реализации булевой функции контактными схемами;
- г) функция Шеннона сложности реализации булевой функции контактными схемами;
- д) контактный дешифратор;
- е) разделительный контактный дешифратор.

2. Сформулируйте:

- а) утверждение о верхней оценке сложности универсального контактного многополюсника для функции n переменных;
- б) теорему об асимптотической оценке функции Шеннона сложности реализации булевых функций контактными схемами;

- в) нижнюю оценку функции Шеннона для контактных схем.

3.7 Метод каскадов для КС и СФЭ

Рассмотрим разложение подфункции произвольной функции $f(x_1, \dots, x_n)$ по переменной:

$$\begin{aligned} f(\sigma_1, \dots, \sigma_{i-1}, x_i, \dots, x_n) &= \\ &= \bar{x}_i \cdot f(\sigma_1, \dots, \sigma_{i-1}, 0, x_{i+1}, \dots, x_n) \vee \\ &\vee x_i \cdot f(\sigma_1, \dots, \sigma_{i-1}, 1, x_{i+1}, \dots, x_n). \end{aligned} \quad (3.9)$$

По формулам (3.9) функцию $f(x_1, \dots, x_n)$ можно последовательно строить, имея, например, константы. Соответствующие схемы (как из функциональных элементов, так и контактные) будут делиться на уровни. При этом на нижнем уровне (нулевом снизу, n -м сверху) реализуются, по мере необходимости, константы, на следующем (первом снизу, $(n-1)$ -м сверху) — функции одной переменной x_n , а на самом верхнем — сама функция $f(x_1, \dots, x_n)$. Принципиальной особенностью метода каскадов является синтез лишь тех подфункций, которые используются в разложениях. В случае СФЭ формулы (3.9) дают построение в явном виде, не требующее дополнительных разложений. При построении контактных схем следует отметить отсутствие побочных проводимостей в силу того, что возвратное движение через i -й уровень приводит к образованию цепи с контактами $\bar{x}_{n+1-i}$ и x_{n+1-i} , не являющейся проводящей. Если

мы рассмотрим представление (3.9) и разложение по первым $i - 1$ переменным в методе Шеннона (3.3), то мы убедимся, что верхние оценки метода применимы и к методу каскадов.

З а м е ч а н и е 2. Метод каскадов является практичным в силу того, что возможен выбор произвольного порядка переменных. При этом может существенно различаться сложность получаемых схем.

Рассмотрим для примера функцию $f = (x_1 \oplus x_3)(x_2 \oplus x_4)$. Рассмотрим два порядка переменных: x_1, x_2, x_3, x_4 и x_1, x_3, x_2, x_4 . При этом получатся остаточные функции:

$$\begin{aligned}
 f(0, x_2, x_3, x_4) &= x_3 \cdot (x_2 \oplus x_4), \\
 f(1, x_2, x_3, x_4) &= \bar{x}_3 \cdot (x_2 \oplus x_4), \\
 f(0, 0, x_3, x_4) &= x_3 \cdot x_4, \\
 f(0, 1, x_3, x_4) &= x_3 \cdot \bar{x}_4, \\
 f(1, 0, x_3, x_4) &= \bar{x}_3 \cdot x_4, \\
 f(1, 1, x_3, x_4) &= \bar{x}_3 \cdot \bar{x}_4, \\
 f(0, x_2, 0, x_4) &= f(1, x_2, 1, x_4) = 0, \\
 f(0, x_2, 1, x_4) &= f(1, x_2, 0, x_4) = x_2 \oplus x_4, \\
 f(0, 0, 0, x_4) &= f(0, 1, 0, x_4) = f(1, 0, 1, x_4) = f(1, 1, 1, x_4) = 0, \\
 f(0, 0, 1, x_4) &= f(1, 0, 0, x_4) = x_4, \\
 f(0, 1, 1, x_4) &= f(1, 1, 0, x_4) = \bar{x}_4.
 \end{aligned}$$

Соответствующие наборы контактных схем приведены на рис. 3.4, 3.5.

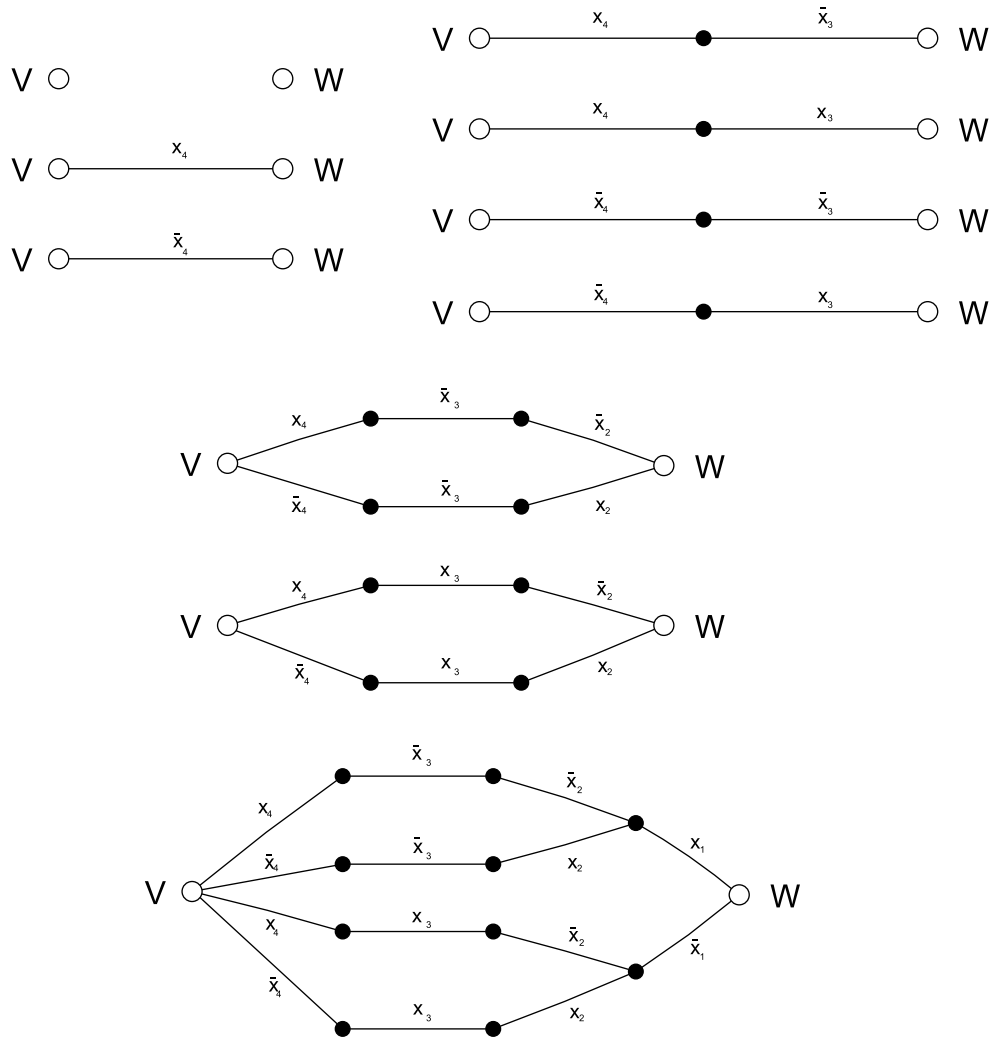


Рис. 3.4

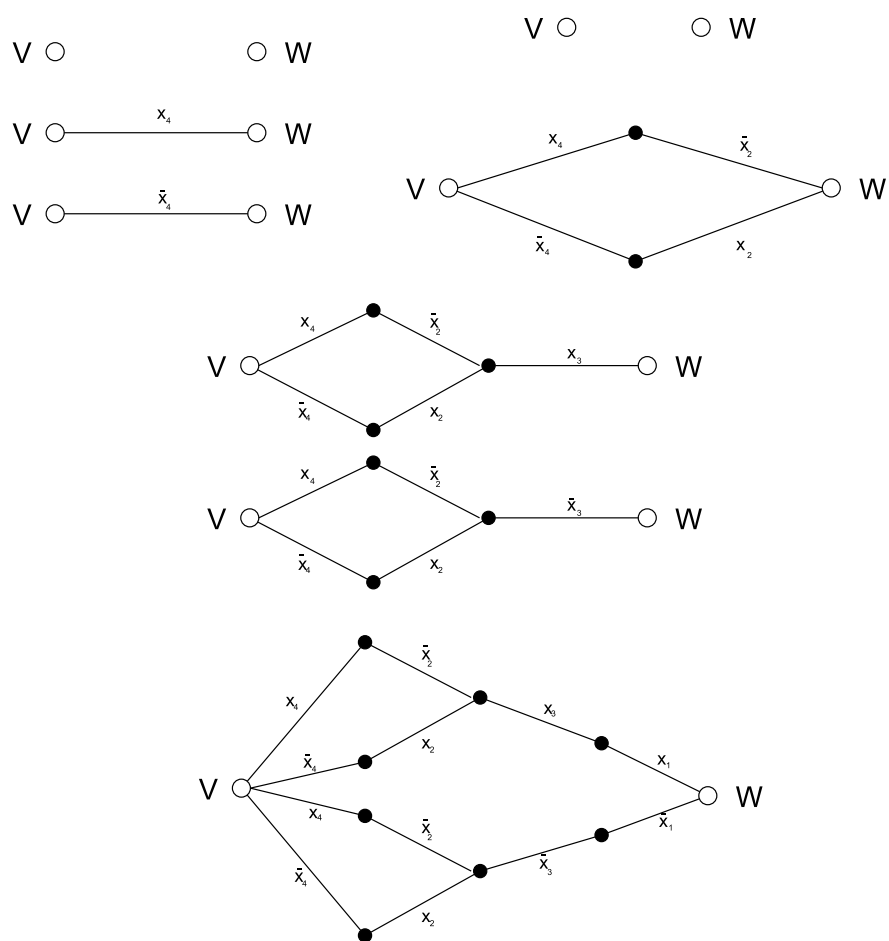


Рис. 3.5

схемы из функциональных элементов показаны на рис. 3.6, 3.7.

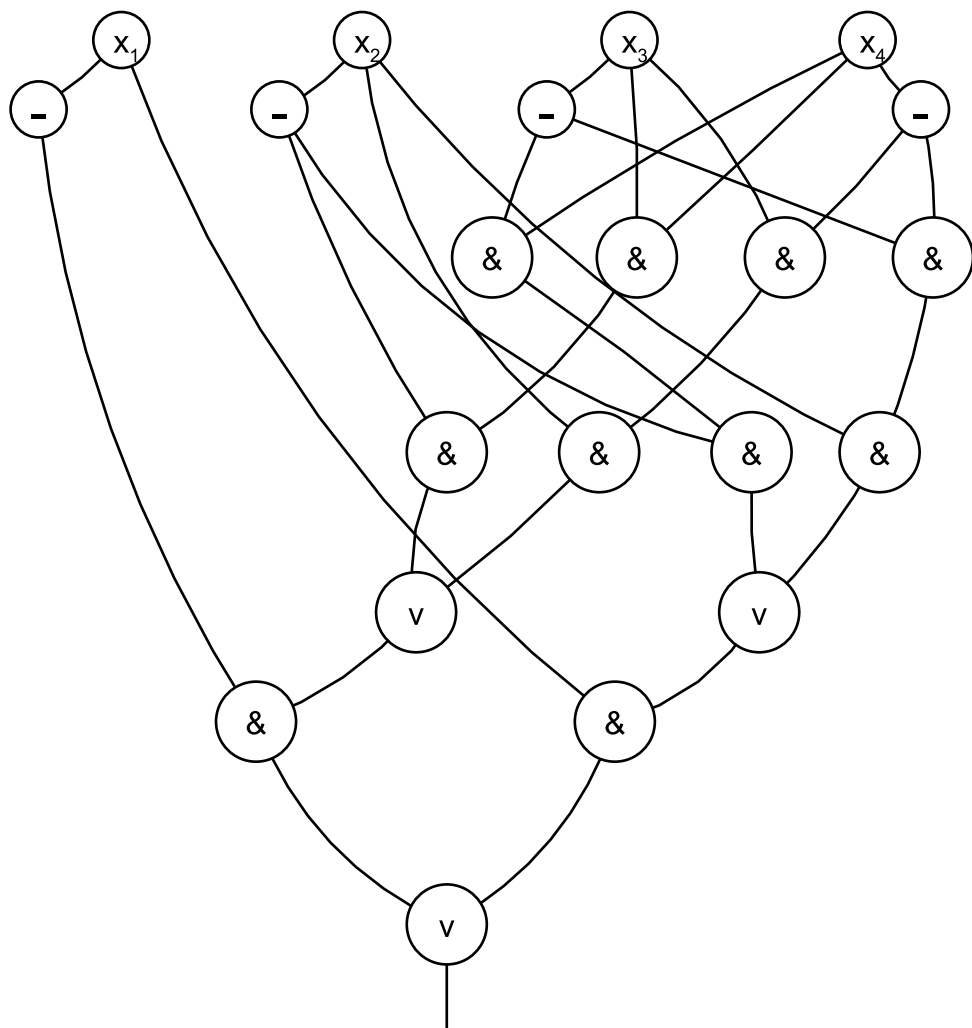


Рис. 3.6

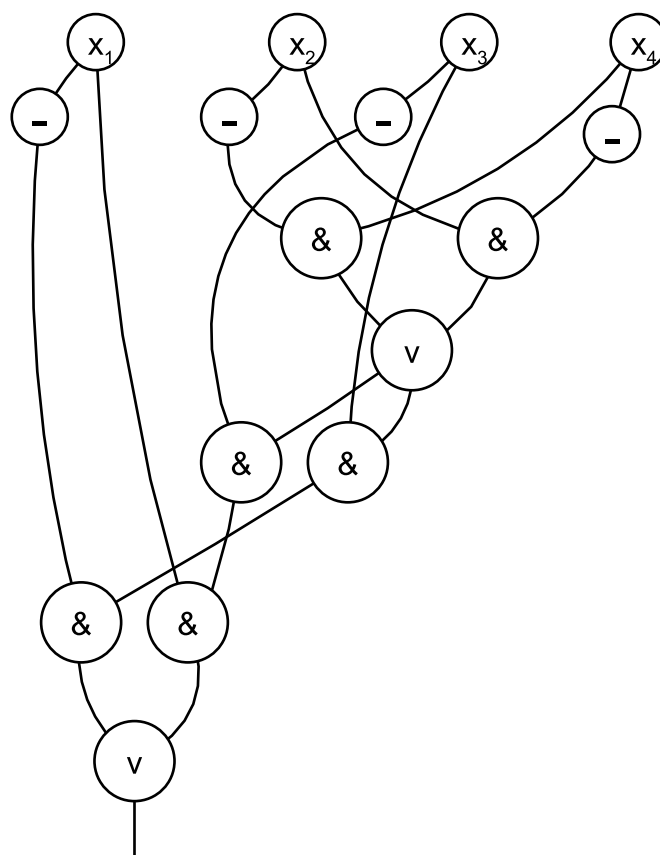


Рис. 3.7

3.8 Самокоррекция контактных схем

Естественной является следующая постановка задачи: построить схему, не меняющую функционирование под действием малого числа неисправностей. Применительно к контактным схемам классическими являются два типа неисправностей: *размыкание* (проводимость контакта становится тож-

дественно равной нулю) и *замыкание* (проводимость контакта становится тождественно равной единице). Наиболее простыми являются задачи построения самокорректирующихся схем относительно *единичного размыкания* (единичного замыкания), т.е. размыкания (замыкания) любого одного контакта. Для решения этой задачи можно применить тривиальное дублирование — заменить каждый контакт на два, помеченных тем же литералом между теми же вершинами (для размыкания) и заменить каждый контакт на два, помеченных тем же литералом и соединяющих цепью через промежуточную вершину те же вершины (для замыкания) (см. рис. 3.8).

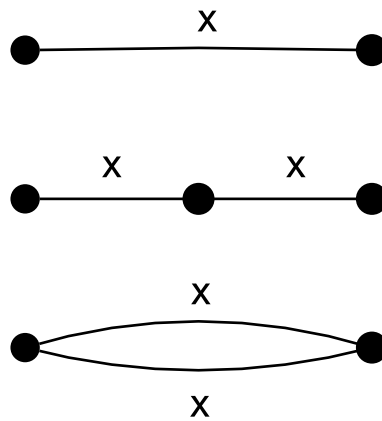


Рис. 3.8

Однако удваивать число контактов необязательно, и за счет этого можно уменьшить сложность самокорректирующихся схем. Следующее утверждение доказывается тривиально.

Л е м м а 7. *Простой цикл контактов x^σ на множестве вершин $V(G)$ реализует функцию проводимости x^σ между всеми вершинами графа G , корректируя единичное размыкание. Граф с множеством вершин, получаемым из $V(G)$ добавлением одной вершины, соединенной контактами x^σ с вершинами множества $V(G)$, корректирует единичное замыкание между вершинами $V(G)$.*

Заметим, что число контактов в самокорректирующейся схеме увеличивается не более чем на единицу, и то, если исходный граф G был деревом.

О.Б.Лупановым был разработан метод синтеза контактных схем, обладающих следующими свойствами.

1. Сложность схем, реализующих булевы функции n переменных, асимптотически не превосходит $\frac{2^n}{n}$ при $n \rightarrow \infty$.
2. Почти все контакты расположены в однородных метелках (деревьях, состоящих из одинаковых контактов, размер которых растет при $n \rightarrow \infty$).

Из свойств 1—2 и леммы 7 вытекает наличие как схем, корректирующих единичное размыкание, так и схем, корректирующих единичное замыкание, асимптотической сложности $\frac{2^n}{n}$.

Заметим, что построение самокорректирующихся схем по оптимальным несамокорректирующимся не всегда является оптимальным. Примером тому могут быть схемы для медианы (см. рис. 3.9 - 3.13). Первая схема (рис. 3.9) реализует медиану

пятью контактами. Вторая (рис. 3.10) корректирует замыкание, а третья (рис. 3.11) — размыкание по лемме 7 за девять контактов. Четвертая (рис. 3.12) и пятая (рис. 3.13) корректируют замыкание (размыкание) за восемь контактов.

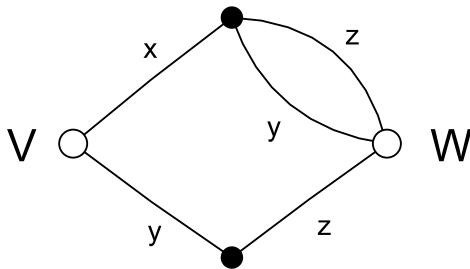


Рис. 3.9

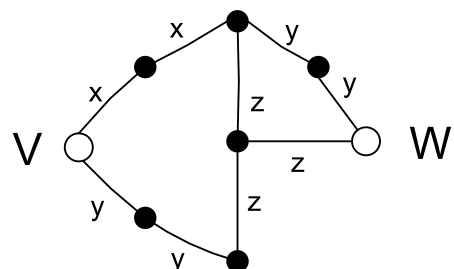


Рис. 3.10

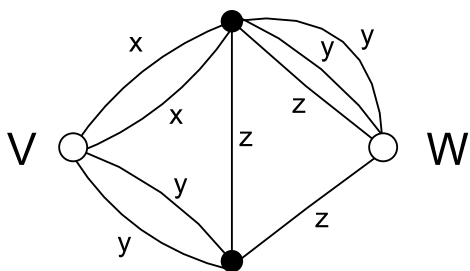


Рис. 3.11

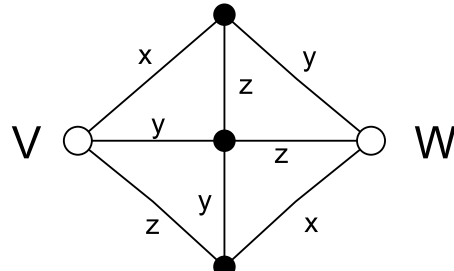


Рис. 3.12

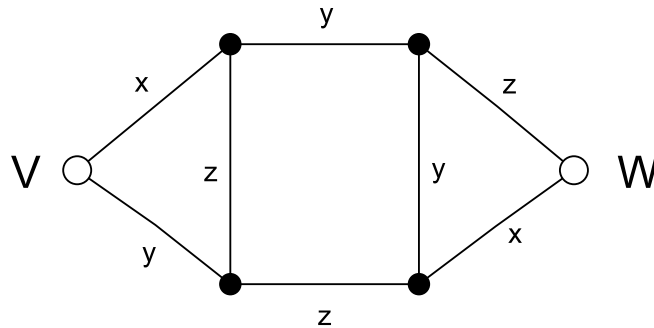


Рис. 3.13

3.9 Схема Кардо. Тестирование на замыкание

Схема Кардо [12] (см. рис. 3.14) реализует функцию $x_1 \oplus x_2 \oplus \dots \oplus x_n$.

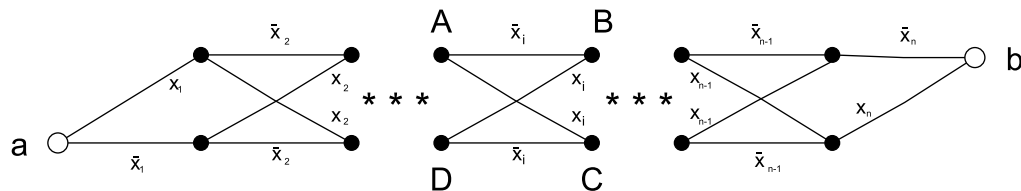


Рис. 3.14

При $n = 2$ замыкание тривиально тестируется на двух наборах: (01) и (10). Разметим схему для построения теста на замыкание (см. рис. 3.14). Пусть f_a — функция проводимости между полюсом a и точкой A , f_b — проводимость между полюсом b и точкой B , f_c — проводимость между полюсом b и точкой C , f_d — проводимость между полюсом a и точкой D .

При этом

$$\begin{aligned}f_a &= x_1 \oplus \dots \oplus x_{i-1}, \\f_b &= x_{i+1} \oplus \dots \oplus x_n \oplus 1, \\f_c &= x_{i+1} \oplus \dots \oplus x_n, \\f_d &= x_1 \oplus \dots \oplus x_{i-1} \oplus 1.\end{aligned}$$

При этом $f_a \oplus f_d = f_b \oplus f_c \equiv 1$. Рассмотрим таблицу проводящих наборов с указанием проводящего контакта по переменной x_i и значения переменной.

Таблица

f_a	f_d	f_b	f_c		x_i
0	1	0	1	DC	0
0	1	1	0	DB	1
1	0	0	1	AC	1
1	0	1	0	AB	0

Множества проводящих наборов $\mathbf{x}$ для всех четырех строк попарно не пересекаются. Наборы $(10\dots 0)$ и $(0\dots 01)$ служат для проверки на размыкание всех размыкающих контактов и контактов x_1 и x_n . При нечетном n для проверки размыканий оставшихся контактов достаточно взять наборы $(1\dots 1)$ и $(01\dots 10)$. При четном — $(01\dots 1)$ и $(1\dots 10)$.

Тестирование схемы Кардо на замыкание. Рассмотрим нулевой набор. На нем функция $x_1 \oplus x_2 \oplus \dots \oplus x_n$ равна нулю, но замкнуты все размыкающие (горизонтальные) кон-

такты. Поэтому при замыкании любого наклонного (замыкающего) контакта возникнет проводимость между полюсами схемы. Пусть n четно. Тогда на единичном наборе в исправной схеме Кардо проводимости между полюсами нет, а замыкание любого размыкающего контакта приводит к появлению такой проводимости. В случае нечетного n к нулевому набору до проверяющего теста достаточно добавить два: $(1 \dots 10)$ и $(01 \dots 1)$. Два набора не могут образовывать проверяющий тест на замыкание, так как для каждой переменной может быть замкнут как замыкающий, так и размыкающий контакт, и функция $x_1 \oplus x_2 \oplus \dots \oplus x_n$ должна быть равна в одном из тестовых наборов нулю, а в другом — единице. Но тогда, если набора два и их длина нечетна, то сумма их будет единичным набором нечетной длины. Последнее означает, что один из наборов имеет нечетное число единиц, что противоречит тому, что он проверяет замыкание.

3.10 Самокорректирующиеся схемы Кардо

Рассмотрим следующую схему, получаемую из схемы Кардо добавлением четырех контактов и двух полюсов (см. рис. 3.15).

Данная схема содержит $4n$ контактов. Склеим пары полюсов c, b и a, d .

Л е м м а 8. *Полученная выше схема реализует функцию $x_1 \oplus x_2 \oplus \dots \oplus x_n$ и корректирует одно размыкание.*

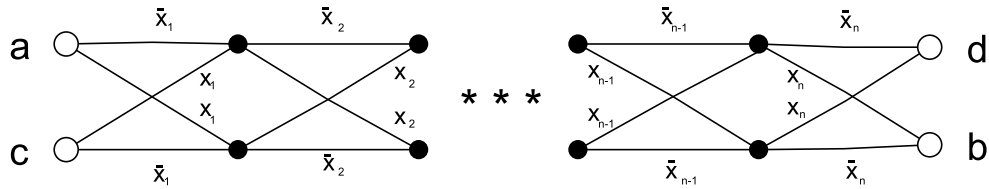


Рис. 3.15

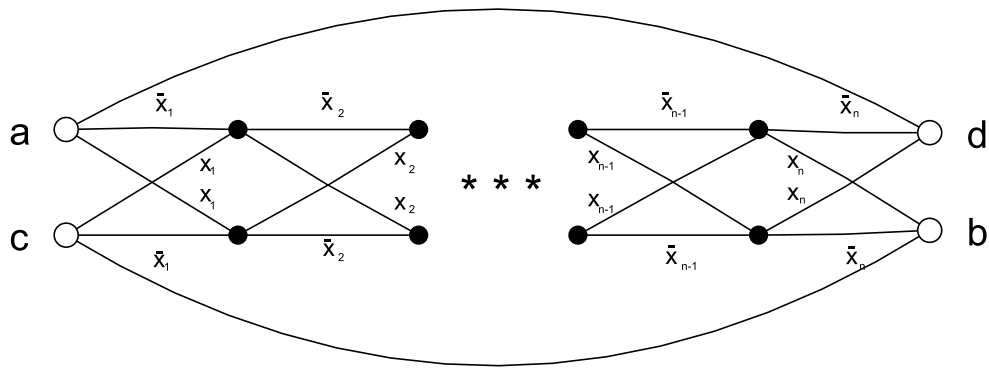


Рис. 3.16

Доказательство. Между полюсами a и b , а также c и d схемы (см. рис. 3.16) реализуется функция $x_1 \oplus x_2 \oplus \dots \oplus x_n$. При этом множества контактов, соответствующих одному и тому же проводящему набору, не пересекаются. Поэтому в схеме, полученной склейкой полюсов, при обрыве одного контакта проводимость сохранится. Отсутствие побочных проводимостей следует из того, что любая смена движения по часовой стрелке на движение в обратном направлении приводит к нулевой проводимости цепи. $\square$

Теорема 19. Минимальная контактная схема для

функции $x_1 \oplus x_2 \oplus \dots \oplus x_n$, самокорректирующаяся относительно одного размыкания, содержит ровно $4n$ контактов.

Д о к а з а т е л ь с т в о. Наличие схемы следует из леммы 8. Замыкающий или размыкающий контакт одной переменной называется *сортом* контакта. Общее количество сортов контактов n переменных равно $2n$. В силу того, что функция $x_1 \oplus x_2 \oplus \dots \oplus x_n$ не монотонна и не антимонотонна ни по одной из переменных, в реализующей ее контактной схеме должны быть контакты всех сортов. Отсюда следует, что самокорректирующаяся схема относительно единичного размыкания должна иметь минимум два контакта каждого сорта, т. е. $4n$ контактов. $\square$

Вопросы и задания.

1. Дайте определения следующих понятий:
 - а) размыкание/замыкание в КС;
 - б) самокорректирующаяся схема;
 - в) схема Кардо.
2. Сформулируйте:
 - а) лемму о коррекции единичного размыкания и замыкания;
 - б) теорему о минимальной КС для функции $x_1 \oplus x_2 \oplus \dots \oplus x_n$, самокорректирующейся относительно одного размыкания.

3. Построить КС для функции $x_1 + \dots + x_n = k$ (арифметическая сумма переменных равна k) методом каскадов.

Глава 4

Эквивалентные преобразования формул

4.1 Эквивалентные преобразования формул в базисе $\{\&, \vee, \neg, 0, 1\}$

Стандартной постановкой задачи эквивалентных преобразований является следующая: требуется построить конечное множество эквивалентных преобразований подформул (т. е. не меняющих реализуемые подфункции), при помощи последовательного применения которых можно преобразовать любую формулу в данном базисе в любую, ей эквивалентную. Такая система называется КПСТ (конечная полная система тождеств). При этом в преобразовании разрешается всюду одновременно заменить переменную на одну и ту же подформулу.

Для решения этой задачи чаще всего выбирают канонический вид формул, а затем описывают преобразования, позво-

ляющие любую формулу привести к каноническому виду. В силу обратимости преобразований этого достаточно. При эквивалентных преобразованиях формул в базисе $\{\&, \vee, \neg, 0, 1\}$ в качестве канонического вида возьмем константу 0 для реализующих ее формул и совершенную ДНФ с упорядоченными в лексикографическом порядке конъюнкциями и упорядоченными литералами внутри конъюнкций для формул, реализующих не равные нулю функции. При этом скобки в конъюнкциях и между дизъюнкциями расставлены так, что операции проводятся слева направо.

Вначале получим систему, позволяющую приводить всевозможные совершенные ДНФ, зависящие от фиксированного набора переменных, друг к другу. Для этого достаточно преобразований, основанных на свойствах ассоциативности и коммутативности конъюнкции и дизъюнкции:

$$\begin{aligned}x \& (y \& z) &= (x \& y) \& z, \\x \vee (y \vee z) &= (x \vee y) \vee z, \\x \& y &= y \& x, \\x \vee y &= y \vee x.\end{aligned}$$

Далее добавим преобразования, позволяющие привести произвольное выражение вида $K_1 \vee \dots \vee K_s$ к совершенной ДНФ (формула “1” также считается конъюнкцией). Рассмотрим причины, в силу которых формула $K_1 \vee \dots \vee K_s$ может не быть совершенной ДНФ.

1. Есть две равные конъюнкции. Тогда добавим преобразо-

вание

$$x \vee x = x.$$

2. В конъюнкции не хватает переменной. Добавим преобразование

$$x = x \& y \vee x \& \bar{y}.$$

3. Одна буква дважды входит в конъюнкцию. Добавим преобразование

$$x \& x = x.$$

4. В конъюнкцию входит переменная вместе с отрицанием. Добавляются преобразования

$$x \& \bar{x} = 0,$$

$$0 \& x = 0,$$

$$0 \vee x = x.$$

5. Одна из конъюнкций является единицей. Тогда после применения преобразования 2 с $x = 1$ применяется преобразование $1 \& x = x$.

Теперь рассмотрим случай, когда формула, отличная от констант 0 и 1, содержит отрицания только над переменными. Используя свойства ассоциативности и коммутативности, можно преобразовать друг к другу все формулы, эквивалентные формуле, представимой в виде дерева A , во внутренних вершинах которого дизъюнкции чередуются с конъюнкциями. При этом внутренние вершины имеют не менее двух сыновей,

а листья представляют собой конъюнкции нескольких переменных. Достаточно добавить преобразование

$$x \& (y \vee z) = x \& y \vee x \& z.$$

При помощи него выражение вида

$$(K_{1,1} \vee \dots \vee K_{1,i_1}) \& \dots \& (K_{m,1} \vee \dots \vee K_{m,i_m})$$

преобразуется к

$$K_{1,1} \& K_{2,1} \& \dots \& K_{m,1} \vee \dots \vee K_{1,i_1} \& K_{2,i_2} \& \dots \& K_{m,i_m},$$

и высота дерева уменьшается.

Пусть теперь допустимы отрицания не только над переменными, но и над подформулами. Их мы можем опустить при помощи преобразований

$$\begin{aligned} \overline{(x \& y)} &= \bar{x} \vee \bar{y}, \\ \overline{(x \vee y)} &= \bar{x} \& \bar{y}, \\ \bar{\bar{x}} &= x. \end{aligned}$$

Наконец, для избавления от констант 0 и 1 достаточно добавить преобразования

$$1 \vee x = 1, \quad \bar{1} = 0, \quad \bar{0} = 1.$$

4.2 Теорема перехода

Пусть $B = \{f_1, \dots, f_m, \dots\}$ — множество функций. Тогда замыканием B будем называть множество функций, реализуемых формулами над B (обозначается через $[B]$; [10], с.14). Если $[B] = B$, то B называется замкнутым классом.

Теорема 20. Пусть $B = \{f_1, \dots, f_m\}$, а $\ddot{B} = \{g_1, \dots, g_s\}$. Пусть при этом $[B] = [\ddot{B}] = A$. Тогда если существует конечная полная система тождеств для формул над B , то существует и конечная полная система тождеств над $\ddot{B}$.

Доказательство. Рассмотрим конечную полную систему тождеств для формул над B :

$$\begin{aligned}\Phi_1(f_1, \dots, f_m) &= \Phi'_1(f_1, \dots, f_m), \\ &\dots \\ \Phi_n(f_1, \dots, f_m) &= \Phi'_n(f_1, \dots, f_m).\end{aligned}$$

При этом, как в тождестве $x \& \bar{x} = 0$, часть переменных может совпадать либо отсутствовать в одной из частей. В силу условия $[B] = [\ddot{B}] = A$ все функции из B выразимы формулами над $\ddot{B}$ и наоборот:

$$\begin{aligned}f_1 &= F_1(g_1, \dots, g_s), \\ &\dots \\ f_m &= F_m(g_1, \dots, g_s)\end{aligned}$$

и

$$\begin{aligned}g_1 &= G_1(f_1, \dots, f_m), \\ &\dots \\ g_s &= G_s(f_1, \dots, f_m).\end{aligned}$$

Покажем, что система тождеств

$$\left\{ \begin{array}{l} \Phi_1(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)) = \\ \quad = \Phi'_1(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)), \\ \quad \dots \\ \Phi_n(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)) = \\ \quad = \Phi'_n(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)), \\ g_1 = G_1(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)), \\ \quad \dots \\ g_s = G_s(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)) \end{array} \right. \quad (4.1)$$

является полной для множества всех формул над $\ddot{B}$. Рассмотрим две произвольные эквивалентные формулы над $\ddot{B}$ — $L'(g_1, \dots, g_s)$ и $L''(g_1, \dots, g_s)$. Если в формулах $L'(g_1, \dots, g_s)$ и $L''(g_1, \dots, g_s)$ выразить функции $\{g_1, \dots, g_s\}$ через $\{f_1, \dots, f_m\}$, то мы получим эквивалентные формулы над B :

$$M'(f_1, \dots, f_m) = L'(G_1(f_1, \dots, f_m), \dots, G_s(f_1, \dots, f_m)),$$

$$M''(f_1, \dots, f_m) = L''(G_1(f_1, \dots, f_m), \dots, G_s(f_1, \dots, f_m)).$$

Для них в силу полноты системы Φ над B существует цепочка эквивалентных преобразований:

$$\begin{aligned} M'(f_1, \dots, f_m) &= L'(G_1(f_1, \dots, f_m), \dots, G_s(f_1, \dots, f_m)) \leftrightarrow \\ &\leftrightarrow \Psi^1(f_1, \dots, f_m) \leftrightarrow \\ &\leftrightarrow \dots \leftrightarrow \\ &\leftrightarrow \Psi^n(f_1, \dots, f_m) \leftrightarrow \\ &\leftrightarrow M''(f_1, \dots, f_m) = L''(G_1(f_1, \dots, f_m), \dots, G_s(f_1, \dots, f_m)). \end{aligned}$$

Существование последней цепочки позволяет построить цепочку эквивалентных преобразований над $\ddot{B}$ при помощи системы (4.1):

$$\begin{aligned}
& L'(g_1, \dots, g_s) \leftrightarrow \\
& \leftrightarrow \dots \leftrightarrow \\
& \leftrightarrow L'(G_1(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)), \dots, \\
& \quad G_s(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s))) \equiv \\
& \equiv M'(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)) \leftrightarrow \\
& \leftrightarrow \Psi^1(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)) \leftrightarrow \\
& \leftrightarrow \dots \leftrightarrow \\
& \leftrightarrow \Psi^n(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)) \leftrightarrow \\
& \leftrightarrow M''(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)) \equiv \\
& \equiv L''(G_1(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s)), \dots, \\
& \quad G_s(F_1(g_1, \dots, g_s), \dots, F_m(g_1, \dots, g_s))) \leftrightarrow \\
& \leftrightarrow \dots \leftrightarrow \\
& \leftrightarrow L''(g_1, \dots, g_s).
\end{aligned}$$

Полнота системы (4.1) доказана.

□

4.3 Теорема Янова

Если $A = [B]$ и при этом для любого C — собственного подмножества B — выполнено $A \neq [C]$, тогда B называется базисом A .

Т е о р е м а 21. (Янов [13]) *В P_k при $k \geq 3$ существует замкнутый класс без базиса.*

Д о к а з а т е л ь с т в о. Пусть функции $f_n(x_1, \dots, x_n)$ равны единице на наборе из всех двоек и нулю — на всех остальных наборах. Рассмотрим множество функций $A = \{f_1, \dots, f_n, \dots\}$ и класс $K = [\{f_1, \dots, f_n, \dots\}]$. Заметим, что все суперпозиции вида $f_m(H_1, \dots, H_m)$ над множеством A равны либо нулю, либо некоторой функции f_j , где $j \leq m$. Таким образом, полная в K система, содержащая не менее двух функций f , не является базисом, так как выполнено соотношение

$$f_m(x_1, \dots, x_m) = f_{m+s}(x_1, \dots, x_m, x_m, \dots, x_m).$$

Базис не может содержать одну функцию f_j , так как с помощью нее нельзя получить функцию f_i при $i > j$. $\square$

4.4 Теорема Мучника

Т е о р е м а 22. (Мучник [13]) *В P_k при $k \geq 3$ существует замкнутый класс со счетным базисом.*

Д о к а з а т е л ь с т в о. Пусть функции $f_n(x_1, \dots, x_n)$ равны единице на наборе из одной единицы и всех остальных двоек и нулю — на всех остальных наборах. Рассмотрим множество функций $A = \{f_2, \dots, f_n, \dots\}$ и класс $K = [\{f_2, \dots, f_n, \dots\}]$. Докажем, что $f_i \notin [\{f_2, \dots, f_{i-1}, f_{i+1}, \dots\}]$. Пусть $f_i = f_j(H_1, \dots, H_j)$ — некоторая суперпозиция над A .

Если $i < j$, то, по крайней мере, одна переменная x_s попадет в две формулы H_p и H_t . При этом выражение справа при $x_s = 1$ заведомо равно нулю. Если же $i > j$, то две переменных x_s и x_t попадут в некую формулу H_p . При этом, если они равны двойке, а некоторая внешняя для H_p переменная равна единице, то функция справа, в отличие от f_i , заведомо равна нулю на этом наборе. Таким образом, множество $\{f_2, \dots, f_n, \dots\}$ образует базис в классе K . $\square$

Следствие. В P_k имеется континуум замкнутых классов, так как все замкнутые классы, порожденные подмножествами функций f , различны.

Вопросы и задания.

1. Дайте определения следующих понятий:

- а) конечная полная система тождеств;
- б) замыкание множества функций;
- в) замкнутый класс;
- г) базис множества A .

2. Сформулируйте:

- а) теорему перехода;
- б) теорему Янова;
- в) теорему Мучника и следствие из нее.

4.5 Пример Линдона. Свойства функции Линдона

Пример Линдона — это пример конечнопорожденного класса в $P_k, k \geq 3$ (конкретно при $k = 7$), для которого не существует конечной полной системы тождеств. Соответствующие примеры были построены и для остальных $k \geq 3$ [20]. Рассмотрим функцию Линдона $\varphi(x, y)$, задаваемую таблицей:

x y	2	3	4
1	1	5	6
5	5	5	5
6	6	6	6

На остальных наборах $\varphi(x, y)$, далее обозначаемая через $x \cdot y$, равна нулю. Функция Линдона обладает следующими свойствами:

$$x \cdot x = 0, \quad 0 \cdot x = x \cdot 0 = 0, \quad x \cdot (y \cdot z) = 0;$$

$$B_i : (\dots (x_1 \cdot x_2) \cdot \dots \cdot x_i) \cdot x_1 = 0, \quad i = 1, 2, \dots, n, \dots;$$

$$C_i : (\dots (x_1 \cdot x_2) \cdot \dots \cdot x_i) \cdot x_2 = (\dots (x_1 \cdot x_2) \cdot \dots \cdot x_i), \quad i = 2, \dots, n, \dots$$

Докажем, что все формулы вида $(\dots (x_{i_1} \cdot x_{i_2}) \cdot \dots \cdot x_{i_k})$, в дальнейшем обозначаемые $x_{i_1} \cdot x_{i_2} \cdot \dots \cdot x_{i_k}$, реализуют разные функции. Если среди переменных $x_{i_1}, x_{i_2}, \dots, x_{i_k}$ нет переменной x_l из отличной от нее формулы, то, положив $x_{i_1} = 1$, $x_{i_2} = \dots = x_{i_k} = 2$, а $x_l = 0$, получим равенство единице нашей формулы и нулю — другой. Рассмотрим случай перестановки переменных $x_{i_1}, x_{i_2}, \dots, x_{i_k}$. Случай несовпадения первых переменных тривиален. Пусть нарушен порядок где-то, начиная со второй переменной. В этом случае пусть переменные x_s и x_t расположены во второй формуле в инверсном порядке. Положим $x_{i_1} = 1, x_s = 3, x_t = 4$, а остальные $x_{i_j} = 2$. Тогда одна из формул реализует константу 5, а другая — константу 6.

4.6 Бесконечная полная система тождеств для класса, порожденного функцией Линдона

$A_1 : x \cdot x = y \cdot y$ — равенство нулей.

$A_2 : (x \cdot x) \cdot y = x \cdot x$ — поглощение нулем второго аргумента.

$A_3 : x \cdot (y \cdot z) = x \cdot x$ — ноль при умножении справа на скобку.

$B_i : (\dots (x_1 \cdot x_2) \cdot \dots \cdot x_i) \cdot x_1 = x_1 \cdot x_1, i = 1, 2, \dots, n, \dots$

$C_i : (\dots (x_1 \cdot x_2) \cdot \dots \cdot x_i) \cdot x_2 = (\dots (x_1 \cdot x_2) \cdot \dots \cdot x_i), i = 2, \dots, n, \dots$

При помощи этих преобразований любая формула, реализующая ноль, приводится к виду $x \cdot x$, а остальные — удалением

лишних переменных при помощи преобразований C_i – к каноническому виду $x_{i_1} \cdot x_{i_2} \cdot \dots \cdot x_{i_k}$.

4.7 Отсутствие КПСТ для замыкания системы из функции Линдона

Т е о р е м а 23. *Для множества функций, порожденных замыканием системы, состоящей из функции Линдона $\varphi(x, y)$, не существует конечной полной системы тождеств.*

Д о к а з а т е л ь с т в о. От противного. Предположим существование конечной полной системы тождеств (КПСТ):

$$D_i : \Sigma_i = \Sigma'_i, \quad i = 1, 2, \dots, m.$$

Тогда при помощи тождеств этой системы можно получить все тождества

$$A_i, \quad i = 1, 2, 3; \quad B_i, \quad i = 1, 2, \dots, n, \dots; \quad C_i, \quad i = 2, \dots, n, \dots \quad (4.2)$$

С другой стороны, при помощи тождеств (4.2) можно получить все тождества $D_i : \Sigma_i = \Sigma'_i, \quad i = 1, 2, \dots, m$. При этом, так как m конечно, для некоторого n достаточно конечной подсистемы тождеств

$$A_i, \quad i = 1, 2, 3; \quad B_i, \quad i = 1, 2, \dots, n; \quad C_i, \quad i = 2, \dots, n, \quad (4.3)$$

которая также будет полной.

Введем в рассмотрение следующее свойство C^n (свойство формул):

1. $x_1, \dots, x_{n+1}$ — множество переменных.
2. Нет подформул-нулей $(x \cdot x)$ и подформул с умножением на скобку слева $(x \cdot (y \cdot z))$.
3. Если переменная встречается дважды, то между ее появлениями в формуле встречаются все остальные переменные.

Преобразования (4.3) сохраняют свойство C^n , а преобразование B_{n+1} — нет. Следовательно, преобразование B_{n+1} не может быть получено при помощи системы преобразований (4.3), что противоречит ее полноте. Полученное противоречие доказывает теорему. $\square$

Вопросы и задания.

1. Дайте описание функции Линдона.
2. Сформулируйте теорему о КПСТ для множества функций, порождаемых замыканием системы, состоящей из функции Линдона.
3. Приведите пример бесконечной ПСТ для замкнутого класса, порожденного функцией Линдона.
- 4*. Приведите пример конечнопорожденного класса без КПСТ для $k = 3$.

Глава 5

Проблема контроля управляющих систем

5.1 Тесты для таблиц. Тривиальные оценки

При решении различных задач контроля управляющих систем возникают задачи, использующие модели, в рамках которых внутри системы проникнуть невозможно. Наиболее известной является модель «черного ящика», на вход которого можно подавать набор аргументов, а на выходе получать значения функции. Целью тестирования является нахождение функции или определение каких-то ее свойств (обычно при наличии дополнительной информации о реализуемой функции). Понятие «черного ящика» естественным образом порождает таблицу, строкам которой соответствуют наборы входных аргументов, а столбцам — выходы, зависящие от реализуемых в «черном ящике» функций. Цель определения функции переходит в определение столбца по значениям в части строк. При

этом возникает естественная мера сложности — необходимое количество строк для решения той или иной задачи.

Введем понятие теста для таблиц (прямоугольных матриц).

Диагностическим тестом для таблицы называется множество ее строк, в котором все столбцы различны. Проверяющим тестом для таблицы называется множество ее строк, в котором все столбцы отличаются от выделенного эталонного столбца. Тесты могут быть условными и безусловными. В случае условного теста разрешается выбирать очередной запрос (строку) в зависимости от ответа на предыдущие. Длиной теста называется количество наборов (строк) в нем. Определим тупиковые тесты (любое собственное подмножество строк не является тестом) и минимальные (среди всех тестов по длине).

Т е о р е м а 24. *Длина любого минимального диагностического теста для матрицы из n столбцов лежит между $\lceil \log_2 n \rceil$ и $(n - 1)$.*

Д о к а з а т е л ь с т в о. Нижняя оценка. Пусть t строк образуют диагностический тест. Тогда все столбцы в них будут различны, при этом $2^t \geq n$, откуда $t \geq \lceil \log_2 n \rceil$.

Верхняя оценка. Предположим, что тест B минимален. Строки разбивают столбцы на классы эквивалентности по значениям в них. Упорядочим строки в тесте B и будем смотреть на изменение количества классов эквивалентности при добавлении очередной строки. Если оно останется прежним, значит, строки, оставшиеся после удаления данной, образуют

тест, и тогда B — не минимальный тест. Добавление строки в тест, если он минимален, должно приводить к дополнительному разбиению классов. Изначально класс один, всего классов не более n , поэтому количество прибавлений числа классов, соответствующих добавлению строк, не превосходит $n - 1$. $\square$

5.2 Верхняя оценка длины диагностического теста для почти всех таблиц

Рассмотрим равновероятностное распределение на множестве $(0, 1)$ -матриц размера $m \times n$. Пусть дана случайная $(0, 1)$ -матрица $A_{m \times n}$. Найдем вероятность $P_{m,n}$ того, что ее строки образуют диагностический тест.

Т е о р е м а 25. Если $m \geq 2 \log_2 n + \phi(n)$, где $\phi(n) \rightarrow \infty$ при $n \rightarrow \infty$, то $P_{m,n} \rightarrow 1$ при $n \rightarrow \infty$.

Д о к а з а т е л ь с т в о. Общее число $(0, 1)$ -матриц размера $m \times n$ равно 2^{mn} . Для того чтобы строки образовывали тест, каждый столбец матрицы должен быть отличен от предыдущих. Таким образом, число матриц, у которых строки образуют тест, равно $2^m \cdot (2^m - 1) \cdot \dots \cdot (2^m - n + 1)$. Оценим снизу вероятность $P_{m,n}$. Имеем.

$$\begin{aligned} P_{m,n} &\geq \frac{2^m \cdot (2^m - 1) \cdot \dots \cdot (2^m - n + 1)}{2^{mn}} \geq \\ &\geq \frac{(2^m - n)^n}{2^{mn}} = \left(1 - \frac{n}{2^m}\right)^n. \end{aligned}$$

Логарифмируя, получим

$$\ln P_{m,n} \geq n \cdot \ln \left(1 - \frac{n}{2^m} \right) \sim -\frac{n^2}{2^m}.$$

Если $m \geq 2 \log_2 n + \phi(n)$ при $n \rightarrow \infty$, то $-\frac{n^2}{2^m} \rightarrow 0$ при $n \rightarrow \infty$, а значит, $P_{m,n} \rightarrow 1$ при $n \rightarrow \infty$. $\square$

5.3 Алгоритм построения всех тупиковых тестов

Рассмотрим две задачи: построение тупиковых проверяющих безусловных тестов и тупиковых диагностических безусловных тестов для таблиц. Пусть требуется построить все тупиковые проверяющие тесты для матрицы A , состоящей из столбцов $\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_n$. Рассмотрим матрицу B , из n столбцов $\mathbf{f}_0 \oplus \mathbf{f}_1, \dots, \mathbf{f}_0 \oplus \mathbf{f}_n$. Пусть найти построить все тупиковые диагностические тесты для матрицы $\tilde{A}$, из столбцов $\mathbf{f}_1, \dots, \mathbf{f}_n$. Рассмотрим матрицу $\tilde{B}$, из $\frac{n(n-1)}{2}$ столбцов $\mathbf{f}_1 \oplus \mathbf{f}_2, \mathbf{f}_1 \oplus \mathbf{f}_3, \dots, \mathbf{f}_1 \oplus \mathbf{f}_n, \mathbf{f}_2 \oplus \mathbf{f}_3, \dots, \mathbf{f}_{n-1} \oplus \mathbf{f}_n$.

Т е о р е м а 26. *Множество строк образует тупиковый тест для матрицы A ($\tilde{A}$) тогда и только тогда, когда оно образует тупиковое покрытие в матрице B ($\tilde{B}$).*

Д о к а з а т е л ь с т в о. Утверждение теоремы следует из того, что элемент $b_{i,j}$ матрицы B (элемент $\tilde{b}_{i,(j,k)}$ матрицы $\tilde{B}$, в которой (j,k) -й столбец — сумма j -го и k -го столбцов матрицы B) равен единице тогда и только тогда, когда в i -й

строке нулевой столбец матрицы B отличается от j -го (в i -й строке j -й и k -й столбцы матрицы $\tilde{B}$ различны). $\square$

Вопросы и задания.

1. Дайте определения следующих понятий:
 - а) диагностический тест;
 - б) проверяющий тест;
 - в) условный/безусловный тест;
 - г) длина теста;
 - д) тупиковый тест;
 - е) минимальный тест.
2. Сформулируйте следующие теоремы:
 - а) о длине любого минимального диагностического теста;
 - б) о верхней оценке длины диагностического теста для почти всех таблиц.
3. Опишите алгоритм построения всех тупиковых тестов.
- 4*. Можно ли существенно улучшить оценку теоремы 25?

Глава 6

Вопросы сложности алгоритмов

6.1 Проблема NP -полноты. Теорема Кука (формулировка)

При решении задач на различных вычислителях естественным является вопрос: насколько растёт сложность (под которой обычно понимают время решения задач) с ростом размера входа. При этом качественное различие возникает между полиномиальной и экспоненциальной функциями роста. Учитывая постоянный рост производительности, в первом случае рост вычислительных возможностей в n раз позволяет решать задачи с ростом размера порядка корня из n степени полинома, а во втором — с ростом лишь на константу. При этом имеется ряд утверждений о том, что вычисления в тех или иных моделях можно свести к вычислениям на машине Тьюринга так, что сложность вырастет лишь полиномиально относительно исходной.

Существует очень много задач, в которых относительно несложно проверить правильный ответ, но непонятно, как построить быстрый алгоритм, выясняющий факт существования такого ответа. Примерами таких задач являются: **существование набора, обращающего КНФ в единицу (выполнимость, ВЫП)**; наличие в графе простого цикла, проходящего через все вершины (гамильтонов цикл, ГЦ); существование вершинной раскраски графа в задаваемое число цветов без смежных вершин, окрашенных в один цвет (РАСКРАСКА), и т. д. Удастся, однако, построить некий формализм, в рамках которого большинство таких задач имеют эквивалентную сложность.

Вначале введем понятие НМТ — недетерминированной машины Тьюринга. В отличие от детерминированной машины Тьюринга МТ ([24], с.178–181), НМТ может иметь более одной команды с одинаковыми текущим состоянием и обозреваемым символом. Если в какой-то момент времени становится возможным выполнение этих команд, то возникает ветвление, и на следующем шаге в каком-то варианте реализуется каждая из команд. При этом считается, что язык распознается за полиномиальное время от длины входа, если существует такой полином от длины входа $p(n)$, что для любого слова из языка длины не более n одна из копий НМТ остановится в принимающем состоянии за время, не большее $p(n)$.

Язык принадлежит классу NP , если существует распознающая его за полиномиальное время от размера входа НМТ.

Язык принадлежит классу P , если существует распознающая его за полиномиальное время от размера входа МТ. Язык L_1 в алфавите A полиномиально сводится к языку L в алфавите B , если существует детерминированная машина Тьюринга, переводящая слова из алфавита A в слова из алфавита B так, что слово принадлежит языку L_1 тогда и только тогда, когда его образ принадлежит языку L , и работающая полиномиальное время от длины входа.

Язык L называется NP -полным, если:

- 1) $L \in NP$;
- 2) любой язык L_1 , принадлежащий NP , полиномиально сводится к L .

Т е о р е м а 27. (Кука [6]). $ВЫП - NP$ -полный язык.

Соотношение $P \subseteq NP$ вытекает непосредственно из определений этих классов, однако доказать равенство или строгое включение до сих пор не удалось. Этот вопрос, называемый проблемой NP -полноты, в 2000 г. получил статус одной из семи проблем тысячелетия.

Вопросы и задания.

1. Дайте определения следующих понятий:

- а) недетерминированная машина Тьюринга (НМТ);
- б) задача выполнимости (ВЫП);
- в) язык принадлежит классу NP ;

- г) язык L_1 в алфавите A полиномиально сводится к языку L в алфавите B ;
- д) язык L называется NP -полным.

2. Сформулируйте теорему Кука.

3*. Докажите, что если $P = NP$, то язык $\{0\}$ в алфавите $\{0, 1\}$ является NP -полным.

6.2 Язык КЛИКА. NP -полнота языка КЛИКА

Язык КЛИКА. Вход — граф G , заданный списком ребер, и число k . Вход принадлежит языку КЛИКА тогда и только тогда, когда в графе G есть полный подграф (клика) на k вершинах. Язык КЛИКА принадлежит NP . В самом деле, НМТ порождает за линейное время все наборы из нулей и единиц, соответствующие характеристическим векторам подмножеств вершин, далее отбирает из них k -вершинные и для каждого выбранного множества за время, ограниченное произведением величины k^2 на время, требующееся на определение наличия конкретного ребра на входе, проверяет, является ли оно кликой.

Т е о р е м а 28. *Язык КЛИКА NP -полный.*

Д о к а з а т е л ь с т в о. Будем полиномиально сводить язык ВЫП к языку КЛИКА. Каждому литералу из КНФ со-

поставим вершину графа G . Соединим все вершины ребрами, за исключением двух ситуаций:

- 1) литералы принадлежат одной скобке;
- 2) литералы являются переменной и ее отрицанием.

В качестве числа k возьмем количество скобок.

Полиномиальность сведения не вызывает сомнений. Докажем корректность. Если в графе G есть клика мощности k , то в КНФ есть k литералов, по одному в каждой скобке, не являющихся отрицаниями друг друга. Такая КНФ выполнима. С другой стороны, если КНФ выполнима, то имеется k одновременно обращающихся в единицу литералов, по одному в каждой скобке. Среди этих литералов нет пары «переменная – ее отрицание», и им соответствует клика размера k в графе. $\square$

6.3 NP -полнота языка 3-ВЫП

Термином 3-КНФ называют КНФ, содержащую в каждой скобке ровно три слагаемых. Язык 3-ВЫП — это язык выполнимых 3-КНФ.

Т е о р е м а 29. Язык 3-ВЫП является NP -полным.

Д о к а з а т е л ь с т в о. Сведем полиномиально язык ВЫП к 3-ВЫП. Для каждой скобки в зависимости от количества литералов в ней сделаем следующие замены:

- 1) $x = (x \vee y_1 \vee y_2)(x \vee y_1 \vee \bar{y}_2)(x \vee \bar{y}_1 \vee y_2)(x \vee \bar{y}_1 \vee \bar{y}_2);$
- 2) $(x_1 \vee x_2) = (x_1 \vee x_2 \vee y_1)(x_1 \vee x_2 \vee \bar{y}_1);$
- 3) $(x_1 \vee x_2 \vee x_3) = (x_1 \vee x_2 \vee x_3);$
- 4) $(x_1 \vee \dots \vee x_n) \leftrightarrow (x_1 \vee x_2 \vee \bar{y}_1)(y_1 \vee x_3 \vee \bar{y}_2) \cdot \dots \cdot (y_{n-4} \vee x_{n-2} \vee \bar{y}_{n-3})(y_{n-3} \vee x_{n-1} \vee x_n), \quad n \geq 4.$

При этом для каждой скобки при замене берутся свои новые переменные y_i .

Первые три замены означают замены слева направо по верному равенству, а четвертая — по одновременной выполнимости. Знак $\leftrightarrow$ означает одновременную выполнимость формул слева и справа. Полиномиальность первых трех преобразований не вызывает сомнений. Замена скобки n литералов на $n-2$ скобки по три в четвертом преобразовании также выполняется за полиномиальное время.

Докажем одновременную выполнимость исходной КНФ и полученной 3-КНФ. Если КНФ выполнима, то имеется набор переменных, обращающих в каждой скобке какой-то литерал в единицу. Если длина исходной скобки была не больше трех, то выполнимость соответствующего куска 3-КНФ легко проверяется по первым трем пунктам. Рассмотрим случай n переменных при $n \geq 4$. Пусть $x_i = 1$. Тогда, положив равными единице все y_j правее скобки, содержащей x_i , и нулю — левее, мы получим выполнимый набор.

Пусть выполнима полученная 3-КНФ. Наличие выполняющего набора исходной КНФ в скобках длины меньше четырех очевидно. В случае $n \geq 4$ достаточно заметить, что если все переменные x_i обращаются в ноль, то не существует набора переменных y , обращающего правую часть формулы пункта 4 в единицу. $\square$

6.4 Полиномиальность языка 2-ВЫП

Язык 2-ВЫП — язык выполнимых 2-КНФ.

Т е о р е м а 30. *Язык 2-ВЫП распознается за полиномиальное время.*

Д о к а з а т е л ь с т в о. Представим КНФ в виде $(x_1 \vee y_1) \dots (x_1 \vee y_s)(\bar{x}_1 \vee z_1) \dots (\bar{x}_1 \vee z_t)K$, где K — конъюнкция, свободная от вхождений переменной x_1 .

$$\begin{aligned} (x_1 \vee y_1) \dots (x_1 \vee y_s)(\bar{x}_1 \vee z_1) \dots (\bar{x}_1 \vee z_t)K &= \\ = (x_1 \vee y_1 \cdot \dots \cdot y_s)(\bar{x}_1 \vee z_1 \cdot \dots \cdot z_t)K & \\ \Leftrightarrow & \\ \&_{i=1}^s \&_{j=1}^t (y_i \vee z_j)K. & \end{aligned}$$

При этом если $s = 0$ или $t = 0$, то

$$(x_1 \vee y_1) \dots (x_1 \vee y_s)(\bar{x}_1 \vee z_1) \dots (\bar{x}_1 \vee z_t)K \Leftrightarrow K.$$

Общее количество скобок в последней формуле не превосходит учетверенного квадрата числа оставшихся переменных,

поэтому мы избавились от одной переменной за полиномиальное время от их общего количества. Последовательно избавляясь от переменных, мы решим задачу за полиномиальное время от длины входа. $\square$

Вопросы и задания.

1. Дайте определения следующих понятий:
 - а) язык КЛИКА / вход принадлежит языку КЛИКА;
 - б) 3-КНФ / язык 3-ВЫП;
 - в) язык 2-ВЫП.
2. Сформулируйте следующие теоремы:
 - а) о NP -полноте языка КЛИКА,
 - б) о NP -полноте языка 3-ВЫП,
 - в) о времени распознавания языка 2-ВЫП.
- 3*. Гамильтонов цикл (ГЦ). Входом является граф, заданный списком ребер. Вход принадлежит языку ГЦ, если в графе есть простой цикл, проходящий через все вершины. Доказать NP -полноту языка ГЦ.
- 4*. Доказать NP -полноту языка 3-ВЫП с дополнительным условием: внутри каждой скобки обращается в единицу ровно один литерал.

6.5 Задача о кратчайшем остовном дереве. Жадный алгоритм

Рассмотрим следующую задачу. Задан связный граф в виде набора ребер с длинами. Требуется найти остовное дерево (подграф исходного графа с тем же множеством вершин, являющийся деревом) с наименьшей суммарной длиной ребер.

Применим следующий алгоритм, называемый жадным [19].

1. Положить граф D пустым (с исходным множеством вершин).
2. Взять кратчайшее из ребер исходного графа, не образующих циклов с ребрами графа D .
3. Если общее число ребер графа D равно $n - 1$, где n — число вершин исходного графа, то D — искомое дерево. Иначе перейти к шагу 2.

Т е о р е м а 31. *Жадный алгоритм строит кратчайшее остовное дерево.*

Д о к а з а т е л ь с т в о. От противного. Пусть алгоритм строит дерево D_{alg} со списком ребер $e_1, \dots, e_{n-1}$. Предположим, что существуют деревья с меньшей суммарной длиной ребер. Среди них есть деревья с наименьшей суммарной длиной ребер — кратчайшие деревья. Тогда среди кратчайших деревьев найдется дерево D'_{opt} , имеющее с D_{alg} наибольшее количество первых общих ребер p , т. е. список ребер дерева D'_{opt}

имеет вид $e_1, \dots, e_p, e'_{p+1}, \dots, e'_{n-1}$. Рассмотрим граф, множество ребер которого состоит из $e_1, \dots, e_p, e_{p+1}, e'_{p+1}, \dots, e'_{n-1}$. Данный граф получен добавлением ребра e_{p+1} к дереву D'_{opt} . Поэтому в нем есть цикл, проходящий через некоторое штрихованное ребро e'_t . Удаляя это ребро, мы получим дерево $\tilde{D}$, суммарная длина ребер которого отличается от суммарной длины ребер дерева D'_{opt} на разность длин ребер e_{p+1} и e'_t — неположительную величину в силу того, что алгоритм на каждом шаге выбирает кратчайшее ребро. Таким образом, мы получили кратчайшее дерево, имеющее с деревом D_{alg} больше общих ребер, чем максимум. Это противоречие доказывает теорему. $\square$

Вопросы и задания.

1. Дайте определение кратчайшего остовного дерева.
2. Сформулируйте теорему о построении кратчайшего остовного дерева жадным алгоритмом.
3. Опишите жадный алгоритм построения кратчайшего остовного дерева.

6.6 Свойства приближенных алгоритмов для задачи МВП

Вершинным покрытием (ВП) графа называется подмножество его вершин, такое, что любое ребро графа инцидентно

хотя бы одной вершине из этого подмножества.

Язык ВП. Входом является граф G и число k . Вход принадлежит языку ВП тогда и только тогда, когда в графе есть вершинное покрытие мощности k .

У т в е р ж д е н и е 2. Язык ВП NP -полон.

Д о к а з а т е л ь с т в о. Построим полиномиальным сведением языка КЛИКА к языку ВП. Рассмотрим язык НМ — независимое множество. Входом является граф и число k . Слово принадлежит языку НМ тогда и только тогда, когда в графе есть независимое множество k вершин (k вершин, не соединенных ребрами). Вход G, k принадлежит языку КЛИКА тогда и только тогда, когда вход $\bar{G}, k$, где $\bar{G}$ — дополнение графа G , принадлежит языку НМ. При этом вход G, k принадлежит языку НМ тогда и только тогда, когда вход $G, n - k$ принадлежит языку ВП, так как наличие независимого множества из k вершин гарантирует, что оставшиеся $n - k$ образуют вершинное покрытие, и наоборот. Принадлежность языка ВП классу NP легко доказывается построением НМТ, сначала строящей все множества вершин мощности k , а затем проверяющей, образует ли каждое из них вершинное покрытие. $\square$

Рассмотрим теперь задачу МВП — построения вершинного покрытия, минимального по числу вершин. Мы рассмотрим два алгоритма для этой задачи и сравним их работу по гаран-

тированному результату.

Если минимизируется функционал качества, то естественным образом вводятся две величины: F_{opt} — оптимальное значение функционала и F_{alg} — значение функционала, достигаемое алгоритмом. Алгоритм называется ε -приближенным, если всегда выполняется неравенство

$$\frac{F_{alg} - F_{opt}}{F_{opt}} \leq \varepsilon.$$

Прямой алгоритм. На каждом шаге выбираем произвольное ребро, добавляем инцидентные ему вершины в покрытие и удаляем все ребра, покрываемые взятой парой вершин.

Т е о р е м а 32. *Прямой алгоритм является 1-приближенным.*

Д о к а з а т е л ь с т в о. Пусть алгоритм выбирает n ребер. Тогда он строит покрытие мощности $2n$. При этом для каждого из n не имеющих общих инцидентных вершин ребер покрытия в него должна входить одна из двух вершин. Поэтому $F_{alg} = 2n$, $F_{opt} \geq n$, откуда следует утверждение теоремы. $\square$

Жадный алгоритм. На каждом шаге выбираем вершину наибольшей степени, добавляем ее в покрытие и удаляем из текущего графа со всеми инцидентными ей ребрами.

Т е о р е м а 33. *Жадный алгоритм не является ε -приближенным ни для какого ε .*

Д о к а з а т е л ь с т в о. Рассмотрим граф с множеством вершин

$$v_{1,1}, \dots, v_{1,n}, v_{2,1}, \dots, v_{2, \lfloor \frac{n}{2} \rfloor}, v_{3,1}, \dots, v_{3, \lfloor \frac{n}{3} \rfloor}, \dots, v_{n-1, \lfloor \frac{n}{n-1} \rfloor}.$$

Вершина $v_{i,j}$ в графе ($i \neq 1$) соединена ребрами с вершинами

$$v_{1, i(j-1)+1}, \dots, v_{1, ij}.$$

Множество вершин $v_{1,1}, \dots, v_{1,n}$ образует покрытие, поэтому $F_{opt} \leq n$. С другой стороны, жадный алгоритм берет все вершины с первым индексом, отличным от единицы, начиная с вершины $v_{n-1, \lfloor \frac{n}{n-1} \rfloor}$. Таким образом,

$$\begin{aligned} F_{alg} &= \lfloor \frac{n}{2} \rfloor + \lfloor \frac{n}{3} \rfloor + \dots + \lfloor \frac{n}{n-1} \rfloor \geq -n + 2 + n \left(\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-1} \right) \geq \\ &\geq -n + 2 + n \cdot \int_2^n \frac{1}{x} dx = -n + 2 + n(\ln n - \ln 2) \geq n(\ln n - \ln 2 - 1). \end{aligned}$$

При этом $\frac{F_{alg} - F_{opt}}{F_{opt}} \geq \ln n - \ln 2 - 2$, т. е. с ростом n ни для какого ε жадный алгоритм не является ε -приближенным. $\square$

Вопросы и задания.

1. Дайте определения следующих понятий:
 - а) вершинное покрытие;
 - б) язык ВП;
 - в) ε -приближенный алгоритм.
2. Сформулируйте:

- а) утверждение о NP -полноте языка ВП;
 - б) теорему о 1-приближении прямого алгоритма построения вершинного покрытия;
 - в) теорему о ε -приближении жадного алгоритма построения вершинного покрытия.
3. Опишите прямой и жадный алгоритмы построения вершинного покрытия.

Глава 7

Дополнительные вопросы

7.1 Сортировка массива. Сортировка слиянием

Задача сортировки массива ставится следующим образом: при помощи сравнения выбираемых по ходу исполнения алгоритма элементов множества требуется упорядочить его по неубыванию (невозрастанию).

Метод сортировки слиянием состоит в последовательном попарном слиянии упорядоченных массивов. Сначала сравниваются наименьшие элементы массивов, и минимальный исключается из одного из старых массивов и присоединяется как наибольший к новому. Процедура повторяется до исчерпания одного из исходных массивов, после чего остаток второго присоединяется к новому массиву в качестве упорядоченного набора наибольших элементов. Сложностью считается количество сравнений. Сложностью сортировки — требуемое ко-

личество сравнений для данного числа элементов в наихудшем случае (сначала фиксируется процедура (алгоритм), затем ищется максимум сложности по всем n -элементным порядкам, а потом — минимум по всем алгоритмам). Процедура сортировки слиянием дает очевидную оценку для сложности сортировки $T(n)$:

$$T(2n) \leq 2T(n) + 2n - 1. \quad (7.1)$$

При четных n мы можем получить оценку

$$T\left(\frac{3}{2}n\right) \leq T(n) + T\left(\frac{n}{2}\right) + \frac{3}{2}n - 1. \quad (7.2)$$

Т е о р е м а 34. $T(n) = \Theta(n \log n)$ при $n \rightarrow \infty$.

Д о к а з а т е л ь с т в о. Из неравенства (7.1) по индукции легко получается соотношение

$$T(n) \leq n \log_2 n + 1 \quad (7.3)$$

для степеней двойки, из которого в силу монотонности функции $T(n)$ вытекает $T(n) \leq 2n \log_2(2n) + 1$ для произвольного n .

С другой стороны, общее количество ответов (отсортированных массивов из n элементов) равно $n!$. Сложность не превосходит двоичного логарифма этой величины, поэтому

$$T(n) \geq \log_2 n! \sim \log_2 \left(\frac{n}{e}\right)^n \sim n \log_2 n, \quad n \rightarrow \infty.$$

Последняя оценка вытекает из формулы Стирлинга $n! \sim \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$. $\square$

Верхнюю оценку теоремы 34 можно улучшить. Справедлива следующая

Л е м м а 9.

$$T(n) \leq \frac{3}{2}n \log_2 n + n + 1.$$

Д о к а з а т е л ь с т в о. Для любого натурального n , не являющегося степенью двойки, при $n \geq 3$ можно найти такие степени двойки m и $2m$, что $m < n < 2m$. При $n \geq \frac{3}{2}m$ из монотонности $T(n)$ и (7.3) следует, что

$$T(n) \leq T(2m) \leq T\left(\frac{4n}{3}\right) \leq \frac{4n}{3} \log_2 \frac{4n}{3} + 1. \quad (7.4)$$

Если же $n < \frac{3}{2}m$, то по неравенству (7.2)

$$T(n) \leq n \log_2 n + 1 + \frac{n}{2} \log_2 \frac{n}{2} + 1 + \frac{3}{2}n - 1.$$

Величина в правой части последнего неравенства в точности равна $\frac{3}{2}n \log_2 n + n + 1$ и превосходит величину в правой части неравенства (7.4). $\square$

Вопросы и задания.

1. Сформулируйте задачу сортировки массива.
2. Опишите метод сортировки слиянием.
3. Сформулируйте:
 - а) теорему об асимптотической сложности метода сортировки слиянием;

- б) лемму о верхней оценке сложности метода сортировки слиянием.

3*. Ответьте на вопрос, можно ли усилить лемму 9.

7.2 Об одновременном нахождении максимума и минимума в массиве

Пусть дан массив $X = \{x_1, \dots, x_n\}$. Требуется при помощи сравнения элементов найти максимальный и минимальный элементы массива. Тривиальный алгоритм, находящий сначала максимум, а затем минимум, требует $2n - 3$ сравнений. Возникает вопрос: есть ли более эффективный алгоритм?

Рассмотрим алгоритм A , добавляющий к массиву с известными максимумом и минимумом два элемента. Для этого требуется три сравнения: элементов между собой, большего с максимумом и меньшего с минимумом. Имеем

$$T_A(n + 2) \leq T_A(n) + 3.$$

Кроме того, $T_A(1) = 0$, $T_A(2) = 1$. Отсюда $T_A(n) \leq \lceil \frac{3n}{2} \rceil - 2$.

Рассмотрим еще один подход. Поделим все элементы, кроме, быть может, одного, на пары. Проведем сравнение внутри пар, затем большие элементы проверим на максимум, меньшие — на минимум. Для данного алгоритма можно получить предыдущее рекурсивное соотношение и точно такой же результат.

Докажем теперь нижнюю оценку $T(n) \geq \lceil \frac{3n}{2} \rceil - 2$. Для этого рассмотрим функцию $u(x_i)$, равную двум, если x_i может быть как минимумом, так и максимумом, единице, если только минимумом либо только максимумом и нулем — иначе. Рассмотрим функцию $U(X) = \sum_{i=1}^n u(x_i)$. В начале работы алгоритма $U(X) = 2n$, в конце — $U(X) = 2$. Величина $U(X)$ будет в худшем случае уменьшаться не более чем на два, и лишь если сравниваются элементы, ранее не сравнивавшиеся ни с одним элементом массива. Отсюда следует нижняя оценка.

Вопросы и задания.

1. Опишите два нетривиальных алгоритма для одновременного нахождения минимума и максимума.
2. Приведите для них нижнюю оценку сложности.

7.3 Нахождение k -го наименьшего элемента в массиве из n элементов

Нашей задачей является нахождение нетривиальной (относительно сложности сортировки) оценки величины $T(n)$ — необходимого количества сравнений для нахождения k -го наименьшего элемента при наихудшем выборе k и наихудших результатах сравнений.

В начале докажем вспомогательное утверждение.

Л е м м а 10. Для сортировки массива из пяти элементов достаточно восьми сравнений.

Д о к а з а т е л ь с т в о. Разобьем массив на два подмножества: из трех и из двух элементов. Первое сортируется за три сравнения, второе — за одно. После этого слияние отсортированных массивов требует не более четырех сравнений.
□

З а м е ч а н и е 3. Оценка леммы не улучшаема, хотя всего имеется 120 перестановок на множестве из пяти элементов и $\lceil \log_2 120 \rceil = 7$, но после любых трех сравнений может возникнуть ситуация с более чем шестнадцатью вариантами ответов.

Л е м м а 11. Для функции $T(n)$ при $n \geq 5$ выполняется рекуррентное неравенство

$$T(n) \leq T(\lfloor \frac{n}{5} \rfloor) + T(\lfloor \frac{7}{10}n + 1.2 \rfloor) + 2n. \quad (7.5)$$

Д о к а з а т е л ь с т в о. Прежде всего заметим, что величина $T(n)$ неотрицательна и не убывает с ростом n . В связи с этим мы можем рассматривать ее как определенную на множестве всех положительных чисел произвольным образом, не противоречащем монотонности. Применим рекурсивный алгоритм.

1. Разбить текущее множество на подмножества по пять элементов.

2. Отсортировать пятерки.
3. Для набора третьих элементов пятерок найти медиану, перейдя к шагу 1.
4. Сравнить все элементы с найденной медианой. Образуются три множества: U — больших медианы элементов, D — меньших медианы элементов и E — равных медиане элементов.
5. Перейти к шагу 1 с множеством D и текущим параметром k либо с множеством U и k , уменьшенным на $|D| + |E|$.

Прокомментируем результаты этапов и оценим количество сравнений.

1. Не более четырех элементов не попадет ни в одно из множеств.
2. По лемме 10 общее число сравнений не превышает $\frac{8n}{5}$.
3. Потребуется не более $T(\lceil \frac{n}{5} \rceil)$ сравнений.
4. Если третий элемент пятерки не превосходит (не меньше) медианы, то первый и второй ее элементы также не превосходят (четвертый и пятый элементы также не меньше) медианы. Отсюда, с учетом того, что вне пятерок расположено не более четырех элементов, величины $|U|$ и $|D|$ оцениваются сверху через $\frac{7}{10}n + 1.2$. Если k -й элемент лежит в множестве E , то он найден, иначе известно, в каком

множестве его искать — U или D . При этом общее число дополнительных сравнений с медианой не превосходит $\frac{2n}{5}$.

5. Сложность не превосходит $T(\lceil \frac{7}{10}n + 1.2 \rceil)$.

Складывая оценки по шагам, получим утверждение леммы.

□

Л е м м а 12. *Если неравенство $T(n) \leq (20 + \delta)n$ справедливо для всех n , меньших $\frac{240}{\delta} + 12$, то оно выполняется для любых n .*

Д о к а з а т е л ь с т в о. Используем верхнюю оценку для величины $T(n)$ из неравенства (7.5):

$$(20 + \delta)\frac{n}{5} + (20 + \delta)(\frac{7}{10} + 1.2) + 2n - (20 + \delta)n = 24 + 1.2\delta + 0.1\delta n.$$

Выражение справа неположительно при $n \geq \frac{240}{\delta} + 12$. □

Т е о р е м а 35. $T(n) \leq 20.2n$.

Д о к а з а т е л ь с т в о. Сортировка массива из n элементов слиянием, согласно лемме 9, требует не более $\frac{3}{2}n \log_2 n + n + 1$ сравнений. При этом $20.2n \geq \frac{3}{2}n \log_2 n + n + 1$ при всех $n < 1250$ и $\frac{240}{0.2} + 12 < 1250$. Таким образом, утверждение теоремы следует из леммы 12. □

Вопросы и задания.

1. Сформулируйте:

- а) лемму о сортировке массива из пяти элементов;
- б) лемму о рекуррентном неравенстве для функции сложности алгоритма $T(n)$;
- б) теорему о верхней оценке для функции сложности алгоритма $T(n)$.

2*. Докажите, что сложность сортировки массива из пяти элементов больше семи.

7.4 Задача тестирования линейности булевой функции

Постановка задачи. Требуется при помощи условного теста выяснить, является ли линейной булева функция в черном ящике.

Т е о р е м а 36. *Сложность тестирования линейности булевой функции n переменных равна 2^n .*

Д о к а з а т е л ь с т в о. Пусть в черном ящике содержится одна из функций 0 и произвольная конъюнкция n литералов $x_1^{\sigma_1} \cdot \dots \cdot x_n^{\sigma_n}$. Тогда для любой последовательности запросов существует конъюнкция, равная нулю на первых $2^n - 1$ наборах, и мы до последнего набора не будем знать, линейная перед нами функция ноль или нелинейная функция — конъюнкция, получая все время ответ "ноль" на наши запросы. $\square$

7.5 Задача доказательства нелинейности булевой функции

Рассмотрим следующую задачу. Пусть нам известна нелинейная булева функция. Требуется сообщить как можно меньше ее значений так, чтобы из них следовала ее нелинейность.

Т е о р е м а 37. *Минимальное количество значений, которое требуется предъявить для доказательства нелинейности булевой функции, равно четырем.*

Д о к а з а т е л ь с т в о. Достаточность. В доказательстве леммы о нелинейной функции ([10], с.20) показывается, что существует константная подстановка на места всех переменных нелинейной функции, кроме двух, сохраняющая нелинейность остаточной функции.

Необходимость. Рассмотрим для функции $f(\mathbf{x})$ три произвольных набора:

$$f(\tilde{\alpha}) = f_0, \quad f(\tilde{\beta}) = f_1, \quad f(\tilde{\gamma}) = f_2.$$

При этом найдутся такие i, j , что $\alpha_i \neq \beta_i, \alpha_i = \gamma_i$, а $\alpha_j = \beta_j, \alpha_j \neq \gamma_j$. Тогда линейная функция

$$f_0 \oplus (f_1 \oplus f_0)(x_i \oplus \alpha_i) \oplus (f_2 \oplus f_0)(x_j \oplus \alpha_j)$$

совпадает с нашей на этих трех наборах. $\square$

Вопросы и задания.

1. Сформулируйте теорему о сложности тестирования линейной функции n переменных.

2. Сформулируйте теорему о минимальном количестве значений для доказательства нелинейности функции.
- 3*. Ответьте на вопрос, что изменится в теореме 37 если рассмотреть не булевы функции, а k -значные (для $k = 3$ и $k = 4$).

7.6 Дешифровка монотонных функций

Цепью в булевом кубе называется возрастающая последовательность соседних наборов. Длиной цепи будем называть количество наборов (обычно ею называется на единицу меньшее количество звеньев цепи). Построим разбиение булева куба на *цепи Анселя*. Разбиение строится последовательным увеличением размерности подкуба. Подкуб размерности 1, состоящий из наборов $(0, \sigma_2, \dots, \sigma_n)$ и $(1, \sigma_2, \dots, \sigma_n)$, покрывается единственной цепью из этих наборов.

Если покрыты подкубы $B_0 = (-\dots - 0, \sigma_{m+2}, \dots, \sigma_n)$ и $B_1 = (-\dots - 1, \sigma_{m+2}, \dots, \sigma_n)$ размерности m , то имеющиеся в них цепи, различающиеся лишь в $(m+1)$ -й компоненте, назовем параллельными. Возьмем все пары параллельных цепей. Внутри каждой пары возьмем максимальный набор в цепи из B_1 и передадим его в параллельную цепь. Получим новое разбиение. При этом из пары параллельных цепей будут получаться две новые цепи (за исключением ситуации, когда обе цепи имели длину 1 — тогда получается одна цепь длины

2).

Алгоритм дешифровки (условного диагностического тестирования) монотонных функций.

1. Взять все цепи Анселя минимальной длины, на которых функция $f(x_1, \dots, x_n)$ еще не определена, и задать ее на них.
2. Перейти к пункту 1, если функция $f(x_1, \dots, x_n)$ еще не определена всюду.

Л е м м а 13. Функция Шеннона длины условного диагностического теста на множестве монотонных функций n переменных не меньше $2 \cdot \binom{n}{\lfloor \frac{n}{2} \rfloor}$ для нечетных n и $\binom{n}{\frac{n}{2}} + \binom{n}{\frac{n}{2}+1}$ для четных n .

Д о к а з а т е л ь с т в о. Пусть у нас имеется априорная информация о том, что функция $f(x_1, \dots, x_{2m+1})$, равная нулю на всех наборах с не более чем m единицами и единице — на остальных, отличается от дешифруемой не более чем на одном наборе с m или $m+1$ единицами. Тогда для расшифровки функции требуется перебрать все такие наборы. Для случая четного $n = 2m$ надо рассмотреть функцию $f(x_1, \dots, x_{2m})$, также равную нулю на всех наборах с не более чем m единицами и единице — на остальных, и функции, отличающиеся от нее не более чем на одном наборе с m или $m+1$ единицами.

□

Л е м м а 14. Если $a < b < c$ — три последовательных набора некоторой цепи Анселя C , причем a и b отличаются только в i -й координате, а b и c — только в j -й, то $i < j$.

Д о к а з а т е л ь с т в о. Цепи растут только присоединением новых наборов сверху, причем присоединяется всегда набор, соседний с уже имеющимся по координате i , где i — номер этапа при построении цепей. Утверждение леммы следует из того, что i увеличивается от этапа к этапу. $\square$

Пусть X — подмножество булева куба. Подкуб, порожденный множеством X , — это наименьший по включению подкуб, содержащий (все элементы) X .

Пусть Y — подмножество булева куба; H — некоторый подкуб. Подкуб H согласован с подмножеством Y , если Y либо целиком лежит в H , либо не пересекается с ним.

Л е м м а 15. Всякий подкуб, порожденный некоторой цепью Анселя, согласован со всеми цепями Анселя.

Д о к а з а т е л ь с т в о. Индукция по построению цепей. Рассмотрим ситуацию, когда объединяются два куба, различающиеся по переменной x_n . Пусть в этих кубах условие леммы выполнено; докажем его для нового куба и цепей Анселя в нем. Рассмотрим произвольную пару параллельных цепей Анселя, C_0 и C_1 , в исходных подкубах. По нашему предположению все новые цепи Анселя, пересекающие эти подкубы, целиком содержатся в них. После удлинения цепь, полученная из C_0 , порождает объединение этих двух подкубов, поэтому для нее

условие леммы выполнено. Рассмотрим теперь укороченную цепь, полученную из C_1 отщеплением верхнего набора: она порождает вдвое меньший подкуб по сравнению с C_1 . Обозначим через H_0 и H_1 соответствующие подкубы (C_1 порождает объединение H_0 и H_1 ; укороченная цепь порождает H_0). По построению (новых) цепей Анселя все старые цепи, лежащие в объединении H_0 и H_1 , подверглись отщеплению своих верхних наборов. Таким образом, нужно доказать, что после этого отщепления каждая (новая) цепь лежит либо целиком в H_0 , либо целиком в H_1 .

Докажем этот факт от противного. Пусть некоторая новая цепь D пересекает и H_0 , и H_1 ; тогда на этой цепи есть два набора, лежащие в H_0 и H_1 соответственно и соседние по некоторой переменной x_j (подкубы H_0 и H_1 отличаются именно по этой переменной). Поскольку от цепи C_1 был отщеплен верхний набор, то по лемме 14 произвольные наборы подкубов H_0 и H_1 могут различаться лишь в значениях переменных x_i , где $i \leq j$. Вернемся к цепи D : до отщепления верхним ее набором был некоторый набор α , соседний с новым верхним α' по некоторой (другой) переменной x_k . Так как $\alpha' \in H_1$, по лемме 14 имеем $k > j$. Но раз оба набора α', α лежат в объединении H_0 и H_1 , то, как мы только что доказали, $k \leq j$. Полученное противоречие доказывает лемму. $\square$

Л е м м а 16. *Для любых максимальной и минимальной вершин любой цепи Анселя весь подкуб, лежащий между ни-*

ми, состоит из точек, лежащих на цепях Анселя меньшей длины.

Доказательство леммы 16 вытекает из леммы 15.

Л е м м а 17. К моменту доопределения монотонной функции на каждой цепи Анселя остаются неопределенными не более двух точек.

Д о к а з а т е л ь с т в о. По лемме 16 весь подкуб B , лежащий между максимальной и минимальной вершинами цепи Анселя A , состоит из точек, лежащих на цепях Анселя меньшей длины. Следовательно, на них монотонная функция f уже определена на момент ее определения на цепи Анселя A . Предположим, что имеются по крайней мере три набора на цепи A , значения функции f на которых не определяются ее значениями на $B \setminus A$. В силу монотонности f эти наборы можно считать подряд идущими на цепи и, не ограничивая общности, отличающимися в последних двух компонентах. Итак, это наборы

$$\mathbf{a}_1 = (\alpha_1, \dots, \alpha_{n-2}, 0, 0),$$

$$\mathbf{a}_2 = (\alpha_1, \dots, \alpha_{n-2}, 1, 0),$$

$$\mathbf{a}_3 = (\alpha_1, \dots, \alpha_{n-2}, 1, 1).$$

Тогда рассмотрим набор $\mathbf{a}_4 = (\alpha_1, \dots, \alpha_{n-2}, 0, 1)$, принадлежащий $B \setminus A$. Если $f(\mathbf{a}_4) = 0$, то по монотонности $f(\mathbf{a}_1) = 0$. Если же $f(\mathbf{a}_4) = 1$, то по монотонности $f(\mathbf{a}_3) = 1$. Таким образом,

значение на одном из этих трех наборов цепи A предопределено. Полученное противоречие доказывает лемму. $\square$

Л е м м а 18. Все цепи Анселя в n -мерном кубе проходят через центр, и средний вес наборов равен $\frac{n}{2}$.

Д о к а з а т е л ь с т в о. Индукцией по размерности куба. Базис (размерность $n = 1$) очевиден.

Шаг индукции. Рассмотрим цепь с минимальной вершиной веса $\frac{n}{2} - \delta$ и максимальной $\frac{n}{2} + \delta$ в подкубе с нулевой $n + 1$ -й компонентой. Ей параллельна цепь с минимальной вершиной веса $\frac{n}{2} - \delta + 1$ и максимальной $\frac{n}{2} + \delta + 1$ в подкубе с нулевой $n + 1$ -й компонентой. После передачи наборов получаются цепи с минимальной вершиной веса $\frac{n}{2} - \delta$ и максимальной $\frac{n}{2} + \delta + 1$ и с минимальной вершиной веса $\frac{n}{2} - \delta + 1$ и максимальной $\frac{n}{2} + \delta$ в кубе размера $n + 1$. $\square$

Л е м м а 19. Количество запросов алгоритма не превосходит $2 \cdot \binom{n}{\lfloor \frac{n}{2} \rfloor}$ для нечетных n и $\binom{n}{\frac{n}{2}} + \binom{n}{\frac{n}{2}+1}$ для четных n .

Д о к а з а т е л ь с т в о. По лемме 17 количество запросов на каждой цепи не превосходит два. Кроме этого, цепочки длины один имеются в среднем слое при четной размерности n в количестве $\binom{n}{\frac{n}{2}} - \binom{n}{\frac{n}{2}+1}$. Утверждение леммы следует из леммы 18. $\square$

Т е о р е м а 38. Функция Шеннона длины условного диагностического теста на множестве монотонных функций

n переменных равна $2 \cdot \binom{n}{\lfloor \frac{n}{2} \rfloor}$ для нечетных n и $\binom{n}{\frac{n}{2}} + \binom{n}{\frac{n}{2}+1}$ для четных n .

Д о к а з а т е л ь с т в о. Верхняя оценка вытекает из леммы 19, нижняя — из леммы 13. $\square$

Вопросы и задания.

1. Дайте определения следующих понятий:

- а) цепь в булевом кубе;
- б) длина цепи;
- в) цепи Анселя.

2. Сформулируйте:

- а) лемму о нижней оценке функции Шеннона длины диагностического теста на множестве монотонных функций n переменных;
- б) лемму о любых максимальной и минимальной вершинах цепи Анселя;
- в) лемму о неопределенных точках на цепях Анселя в момент доопределения монотонной функции;
- д) лемму о прохождении всех цепей Анселя через центр куба и о среднем весе набора.

3. Сформулируйте теорему о функции Шеннона длины условного диагностического теста на множестве монотонных функций n переменных.

7.7 Распознавание монотонности по вектор-столбцу

Рассмотрим следующую задачу. Требуется построить схему (для определенности в базисе всех булевых функций двух переменных B_2), распознающую монотонность подаваемой на вход в виде вектор-столбца длины $2^n = N$ функции $f(x_1, \dots, x_n)$. Для построения такой схемы можно использовать следующую лемму.

Л е м м а 20. *Если функция $f(x_1, \dots, x_n)$ немонотонна, то найдутся два соседних набора, на которых нарушается монотонность.*

Д о к а з а т е л ь с т в о. Нарушение монотонности означает наличие двух наборов $\mathbf{a}^n$ и $\mathbf{b}^n$, таких, что $\mathbf{a}^n < \mathbf{b}^n$, но $f(\mathbf{a}^n) = 1$, а $f(\mathbf{b}^n) = 0$. Соединим наборы $\mathbf{a}^n$ и $\mathbf{b}^n$ возрастающей цепью. На ней обязательно имеются два соседних набора $\mathbf{c}^n$ и $\mathbf{d}^n$, таких, что $\mathbf{c}^n < \mathbf{d}^n$, но $f(\mathbf{c}^n) = 1$, а $f(\mathbf{d}^n) = 0$. $\square$

Из леммы 20 вытекает утверждение.

Л е м м а 21. *Существует последовательность схем сложности $O(N \log N)$, распознающих монотонность функций n переменных.*

Д о к а з а т е л ь с т в о. Схема реализуется по формуле

$$\begin{aligned} & \&_{j=1}^n \&_{\sigma_1, \dots, \sigma_{j-1}, \sigma_{j+1}, \dots, \sigma_n} (f(\sigma_1, \dots, \sigma_{j-1}, 0, \sigma_{j+1}, \dots, \sigma_n) \rightarrow \\ & \rightarrow f(\sigma_1, \dots, \sigma_{j-1}, 1, \sigma_{j+1}, \dots, \sigma_n)). \quad \square \end{aligned}$$

Однако оказывается, что данный результат можно улучшить. Для этого мы будем строить код монотонной функции, фиксируя всевозможные нарушения монотонности. Кодом монотонной функции назовем количество единиц на цепях Анселя. Длина кода будет одинакова для всех монотонных функций m переменных и, так как длина цепей не превосходит $m + 1$, равна $O(\frac{2^m}{\sqrt{m}} \log(m + 1))$ при $m \rightarrow \infty$.

Л е м м а 22. Пусть $g_0(x_1, \dots, x_m)$ и $g_1(x_1, \dots, x_m)$ — монотонные булевы функции, а L — длина их кодов. Тогда существует схема из функциональных элементов, входами которой являются коды этих функций, распознающая монотонность функции $g = \bar{x}_{m+1}g_0 \vee x_{m+1}g_1$ и строящая ее код, сложность которой не превосходит cL , где c — константа, зависящая только от базиса.

Д о к а з а т е л ь с т в о. Если функция g немонотонна, то по лемме 20 монотонность нарушается на паре соседних наборов. В силу монотонности функций g_0 и g_1 эти наборы могут отличаться лишь в $m + 1$ -й компоненте, т. е. g немонотонна тогда и только тогда, когда в m -мерном кубе имеется цепь, на которой число единиц g_0 больше числа единиц g_1 . Сравним эти числа и возьмем конъюнкцию всех результатов сравнений. Количество пар цепей равно $\binom{m}{\lfloor \frac{m}{2} \rfloor}$, разрядность числа нулей не превосходит $\log_2(m + 2)$. Сравнение чисел реализуется схемой линейной сложности относительно длины их записи. Реализация конъюнкции результатов сравнений имеет линейную

сложность относительно числа сравнений.

Для построения кода функции g рассмотрим произвольную пару параллельных цепей Анселя. Если обе цепи пары не были нулевыми и имели длину больше 1, то на длинной цепи количество единиц будет на единицу больше, чем на цепи в коде g_0 , а на короткой — на единицу меньше, чем на цепи в коде g_1 . Из параллельных цепей длины 1 получается одна цепь длины 2. Нетрудно видеть, что сложность схемы, реализующей такое преобразование кодов, вновь ограничена линейной функцией от L . $\square$

Т е о р е м а 39. Пусть $N = 2^n$. Тогда для любого базиса существует последовательность схем $\{S_n\}$ сложности $O(N\sqrt{\log N} \log \log N)$ при $n \rightarrow \infty$, такая, что входом схемы $\{S_n\}$ является вектор-столбец значений булевой функции n переменных, а на выходе схема распознает монотонность входной функции.

Д о к а з а т е л ь с т в о. Пусть на вход схемы подается вектор-столбец произвольной функции $f(x_1, \dots, x_n)$. Схема $\{S_n\}$ будет иметь n уровней. На k -м уровне расположены $\frac{N}{2^k} = 2^{n-k}$ подсхем, строящих код подфункции $f(x_1, \dots, x_k, \sigma k + 1, \dots, \sigma_n)$ по кодам подфункций $f(x_1, \dots, \dots, x_{k-1}, 0, \sigma k + 1, \dots, \sigma_n)$ и $f(x_1, \dots, x_{k-1}, 1, \sigma k + 1, \dots, \sigma_n)$ и проверяющих монотонность получаемых подфункций. В силу леммы 22 сложность каждой подсхемы на k -м уровне равна $O(\frac{2^k}{\sqrt{k}} \log(k+1))$ при $k \rightarrow \infty$. Выходом схемы S_n является

конъюнкция выходов всех подсхем, проверяющих монотонность проверяемых подфункций. Сложность всей схемы S_n не превосходит

$$\sum_{k=1}^{\log N} \frac{N}{2^k} O\left(\frac{2^k}{\sqrt{k}} \log(k+1) + 1\right) = N \sum_{k=1}^N O\left(\frac{\ln k}{\sqrt{k}}\right).$$

Начиная с $n = 8$ положительная последовательность $\left\{\frac{\ln n}{\sqrt{n}}\right\}$ монотонно убывает. Поэтому

$$\sum_{k=1}^N \frac{\ln k}{\sqrt{k}} \leq \sum_{k=1}^8 \frac{\ln k}{\sqrt{k}} + \int_8^{\log N} \frac{\ln x}{\sqrt{x}} dx.$$

При этом

$$\int_8^{\log N} \frac{\ln x}{\sqrt{x}} dx = 2 \ln x \sqrt{x} - 4\sqrt{x} \Big|_8^{\log N} = O(\sqrt{\log N} \log \log N).$$

□

Нетрудно показать, что для предполных классов сохранения констант, линейных и самодвойственных булевых функций существуют схемы линейной сложности, распознающие принадлежность данным классам заданной вектор-столбцом функции. Из этого вытекает

Т е о р е м а 40. *Для распознавания полноты системы булевых функций, заданных вектор-столбцом, можно построить последовательность схем сложности $O(N\sqrt{\log N} \log \log N)$, где N — общая длина входа.*

7.8 Критерий кографа. Кографы и 2-деревья

Граф G называется A , если каждый его порожденный подграф либо несвязен, либо обладает несвязным дополнением. Граф $P_4 = (\{a, b, c, d\}, \{(a, b), (b, c), (c, d)\})$ — цепь из трех звеньев. Граф P_4 не является кографом. Это самодополнительный граф (его дополнением является он сам).

Т е о р е м а 41. *Граф G является кографом тогда и только тогда, когда среди его порожденных подграфов нет подграфа/, изоморфного графу P_4 .*

Д о к а з а т е л ь с т в о. Необходимость. Если граф G содержит подграф, изоморфный P_4 , то этот подграф связан вместе со своим дополнением, и граф G не является кографом по определению.

Достаточность. От противного. Предположим, что некоторый граф, не содержащий подграф, изоморфный P_4 , не является кографом. Будем удалять вершины этого графа так, чтобы он сохранял это свойство, пока это возможно. Полученный минимальный граф обозначим $\tilde{G}$. Назовем вершину графа $\tilde{G}$ черной, если ее удаление делает несвязным получающийся граф, и белой — если ее удаление делает несвязным его дополнение. При этом в силу минимальности графа $\tilde{G}$ любая вершина получит какой-то цвет. В силу того, что либо граф, либо его дополнение связны, любая вершина получит ровно один цвет. Так как у любого связного графа есть остовное

дерево, а удаление листа в дереве оставляет его связным, все вершины не могут быть окрашены в один цвет.

Будем размещать вершины графа $\tilde{G}$ на координатной плоскости. Разместим черную вершину X в точке $(1, 0)$ (см. рис. 7.1). При этом в несвязном графе $\tilde{G} \setminus X$ все вершины можно разбить на два непустых множества так, чтобы между вершинами разных множеств не было ребер. Одно из этих множеств вершин разместим в точках с положительными абсциссами, другое — с отрицательными. Разместим белую вершину Y в точке $(0, 1)$ (см. рис. 7.1). Без ограничения общности можно считать, что ребро (X, Y) принадлежит множеству ребер графа $\tilde{G}$. При этом в графе $\tilde{G} \setminus Y$ с несвязным дополнением все вершины можно разбить на два непустых множества так, чтобы между вершинами разных множеств всегда были ребра. Одно из этих множеств вершин разместим в точках с положительными ординатами, другое — с отрицательными. В нашем разбиении на осях лежат только точки X и Y , обязаны присутствовать точки с отрицательными абсциссами и точки с отрицательными ординатами и не может быть точек в противоположных четвертях (между такими вершинами одновременно должно и не должно быть ребро). Из этих условий следует наличие точки Z (см. рис. 7.2) в третьей четверти.

По построению $(X, Z) \in E(\tilde{G})$, а $(Y, Z) \notin E(\tilde{G})$. При от-

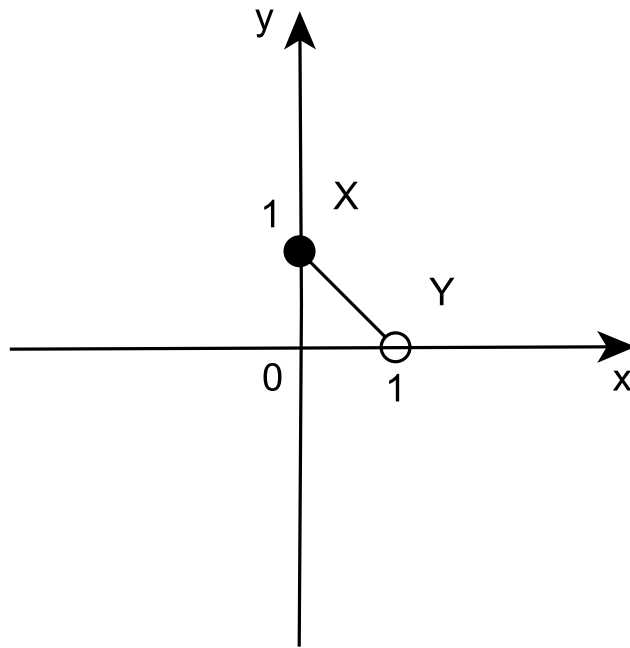


Рис. 7.1

сутствии вершины во второй четверти, не связанной с вершиной X ребром, последняя будет изолированной в графе-дополнении графа $\tilde{G}$, что противоречит связности дополнения графа $\tilde{G}$. Назовем эту вершину W (см. рис. 7.3) и получим подграф $(\{X, Y, Z, W\}, \{(X, Y), (X, Z), (Z, W)\})$ графа $\tilde{G}$, изоморфный P_4 . $\square$

Далее будем для определенности считать корни деревьев

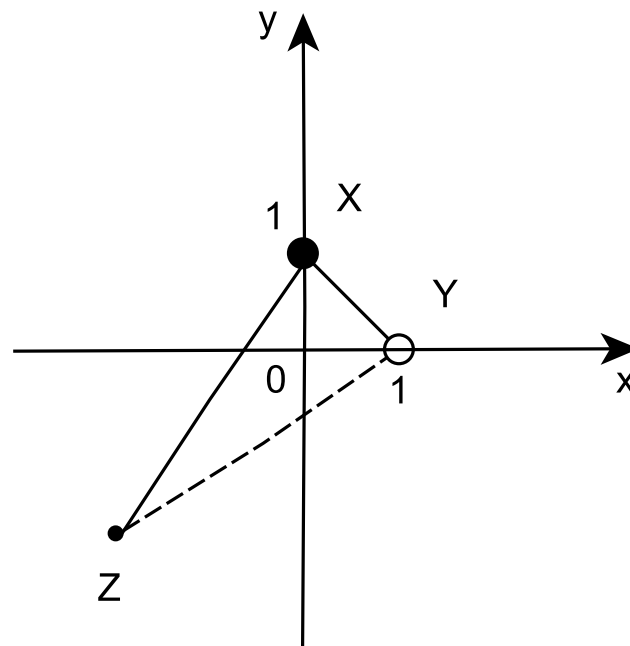


Рис. 7.2

расположенными внизу. Назовем корневое дерево *2-деревом*, если оно обладает следующими свойствами.

1. Все его листья помечены различными символами.
2. Все его внутренние вершины имеют не менее двух сыновей.
3. Все его внутренние вершины правильно раскрашены в два цвета: черный и белый.

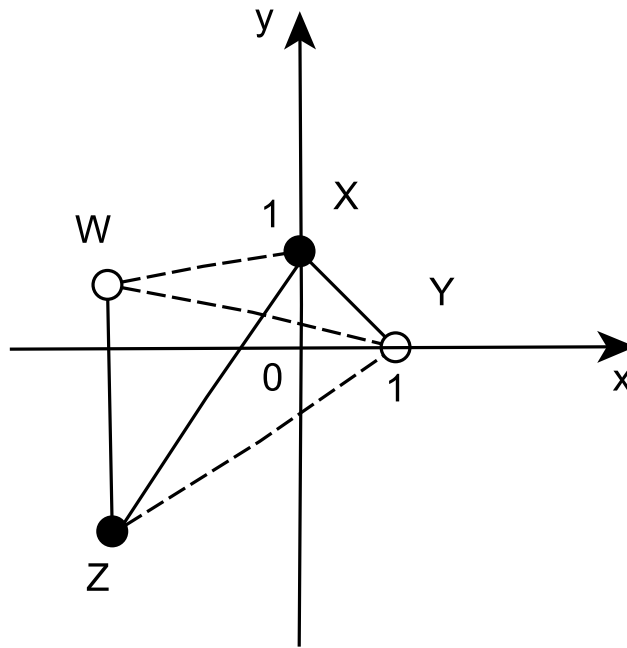


Рис. 7.3

По 2-дереву D можно построить граф $G(D)$ по следующим правилам:

- 1) множество $V = \{v_1, \dots, v_n\}$ вершин графа $G(D)$ – это множество листьев 2-дерева D ;
- 2) ребро (v_i, v_j) принадлежит множеству ребер графа $G(D)$ тогда и только тогда, когда последняя общая вершина на путях из вершин v_i и v_j в корень дерева D покрашена в белый цвет.

Л е м м а 23. $G(D)$ — кограф.

Д о к а з а т е л ь с т в о. Для любого подмножества V' листьев дерева D существует наименьшее (по частичному порядку подграф меньше графа) поддерево, содержащее эти листья. Если корень этого поддерева покрашен в черный цвет, то подграф графа $G(D)$, порожденный множеством вершин V' , несвязен, если в белый, то несвязно дополнение графа $G(D)$. Таким образом, $G(D)$ — кограф по определению. $\square$

Рассмотрим обратную рекурсивную процедуру построения дерева по кографу, используя для удобства обозначение $D(G)$.

На вход процедуры подается кограф и вершина дерева (будущий корень).

1. Если на вход подан связный кограф G' с не менее чем двумя вершинами, то текущая вершина дерева красится в белый цвет. Далее процедура обращается к себе для каждого подграфа графа G' , соответствующего максимальной компоненте связности графа $\overline{G}'$. При этом текущая вершина соединяется ребром с вершиной, подаваемой на вход новой процедуры.
2. Если на вход подан несвязный кограф G' с не менее чем двумя вершинами, то текущая вершина дерева красится в черный цвет. Далее процедура обращается к себе для каждого подграфа графа G' , соответствующего максимальной компоненте связности графа G' . При этом теку-

щая вершина соединяется ребром с вершиной, подаваемой на вход новой процедуры.

3. Если на вход подан граф с одной вершиной, то ее пометка копируется на вершину дерева.

Л е м м а 24. $G(D(G)) = G$.

Д о к а з а т е л ь с т в о. Доказательство следует из того, что ребро (v_i, v_j) одновременно принадлежит множествам ребер графов G и $G(D(G))$. $\square$

Вопросы и задания.

1. Дайте определение кографа.
2. Сформулируйте критерий кографа.
- 3*. Найти все графы, такие, что при удалении из них любой вершины получается кограф.

7.9 Каноническое дерево неповторной функции в элементарном базисе

Функция называется *бесповторной* в базисе B , если выражается в нем формулой, в которую каждая переменная входит не более одного раза. В силу традиций и для удобства зачастую базис считают наследственным, т. е. содержащим

вместе с функциями все получающиеся из них константными подстановками. При этом неповторными в элементарном базисе (далее просто неповторными) становятся константы 0 и 1. Каноническим деревом неповторной функции называется корневое дерево, обладающее следующими свойствами:

- 1) все внутренние вершины дерева имеют не менее двух сыновей;
- 2) все внутренние вершины дерева помечены символами $\&$, $\vee$, причем смежные вершины помечены различными символами;
- 3) листья помечены переменными либо их отрицаниями;
- 4) различные листья помечены литералами, соответствующими разным переменным. Изолированный корень может быть помечен константой 0 или 1.

Функционирование канонического дерева определяется естественным образом от листьев к корню.

Пример канонического дерева для функции

$$f(\tilde{x}^8) = x_1(\bar{x}_2 \vee x_8)(x_3 \vee \bar{x}_4 \vee x_6 \vee x_5\bar{x}_7).$$

Л е м м а 25. Для любой неповторной функции существует каноническое дерево.

Д о к а з а т е л ь с т в о. Рассмотрим произвольную неповторную формулу, реализующую неповторную функцию $f(x_1, \dots, x_n)$, отличную от константы и функции одной

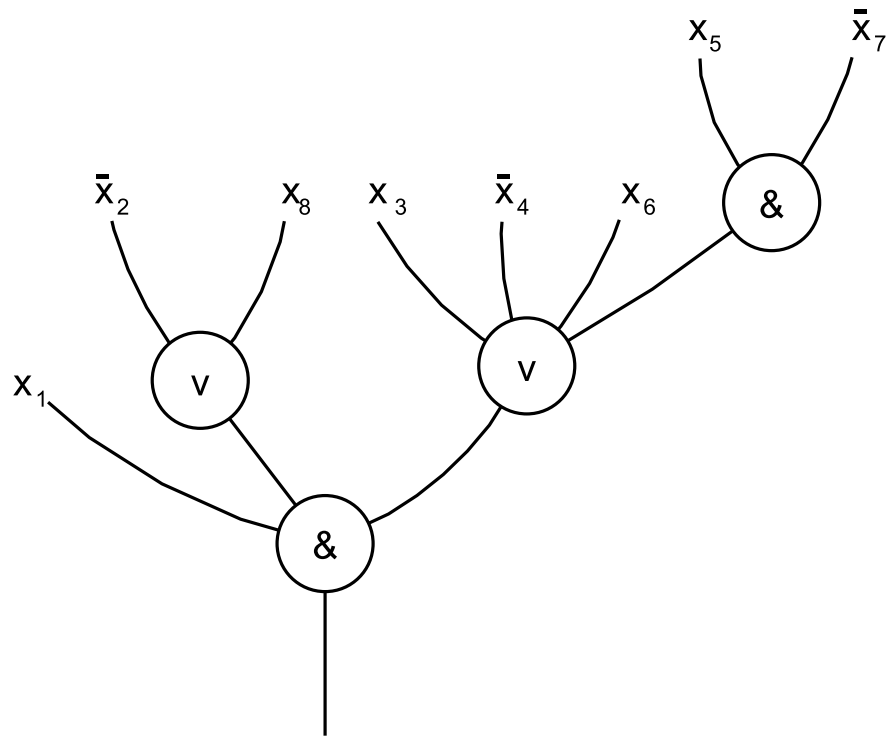


Рис. 7.4

переменной. При помощи правил Де Моргана поднимем отрицания до переменных, после чего склеим в одну вершину конъюнкции и дизъюнкции по ассоциативности. Получим каноническое дерево. $\square$

Л е м м а 26. Каноническое дерево неповторной функции единственно.

Д о к а з а т е л ь с т в о. Любая неповторная функция $f(x_1, \dots, x_n)$ обладает следующим свойством: для любой ее переменной x_i найдется константная подстановка $x_i = \sigma_i$, такая, что функция $f(x_1, \dots, x_{i-1}, \sigma_i, x_{i+1}, \dots, x_n)$ существенно зависит от всех $n - 1$ переменных (*незабывающая подстановка*). Производя забывающие подстановки последовательно, мы можем получить всевозможные подфункции функции $f(x_1, \dots, x_n)$, существенно зависящие от одной и двух переменных. На каноническом дереве функция, существенно зависящая от одной переменной, однозначно определяется литералом листа (переменная с отрицанием или без), а функция двух переменных — их литералами на листьях и последней общей пометкой на путях из корня к этим листьям (конъюнкцией или дизъюнкцией).

Предположим наличие у функции $f(x_1, \dots, x_n)$ двух различных канонических деревьев. Наличие пометки, соответствующей переменной на одном дереве и ее отрицанию на другом, невозможно. Считая, что пометка $\vee$ белого цвета, а пометка $\&$ — черного, мы получим два различных 2-дерева D' и D'' с одинаковыми кографами $G(D')$ и $G(D'')$. Полученное противоречие доказывает лемму. $\square$

Вопросы и задания.

1. Дайте определения следующих понятий:

а) функция неповторна в базисе B ;

б) каноническое дерево бесповторной функции.

2. Сформулируйте леммы:

а) о существовании канонического дерева для бесповторной функции;

б) о единственности канонического дерева для бесповторной функции.

3*. Введите с доказательством единственности понятие канонического дерева для бесповторных функций в базисе B_2 всех двуместных функций.

7.10 Наличие запретных конфигураций у функции Стеценко

Преобразованиями *обобщенной однотипности* называются замена переменной или самой функции на ее отрицание и перестановка переменных. Функции, получаемые друг из друга конечным числом преобразований обобщенной однотипности, называются *обобщенно однотипными*.

Повторная в базисе B функция называется *слабоповторной*, если любая ее подфункция бесповторна в этом базисе. В.А.Стеценко [15] получил все слабоповторные функции в элементарном базисе $B_0 = \{\&, \vee, \neg\}$. С точностью до преобразований обобщенной однотипности они образуют следующие

пять семейств:

$$\begin{aligned}
 f_d^s &= x_1(\bar{x}_2 \vee x_3 x_4 \dots x_s) \vee \bar{x}_2 \bar{x}_3 \dots \bar{x}_s, \quad s \geq 3, \\
 f_t^s &= x_1 x_2 \dots x_s \vee \bar{x}_1 \bar{x}_2 \dots \bar{x}_s, \quad s \geq 2, \\
 f_m^s &= x_1(x_2 \vee \dots \vee x_s) \vee x_2 x_3 \dots x_s, \quad s \geq 3, \\
 f_4 &= x_1(x_2 \vee x_3) \vee x_3 x_4, \\
 f_5 &= x_1(x_2 \vee x_3 x_4) \vee x_5(x_2 x_3 \vee x_4).
 \end{aligned} \tag{7.6}$$

Запретной четверкой назовем две пары соседних наборов, на одной из которых реализуется остаточная функция — переменная, а на другой — ее отрицание. Запретной шестеркой называются шесть наборов, разбитых на две тройки, расположенные на параллельных квадратах. При этом в одной тройке соседними с нулем являются два набора с единицами, а в другой — соседними с единицей являются два набора с нулями. При параллельном наложении квадратов вышеупомянутые ноль и единица занимают противоположные позиции.

Т е о р е м а 42. *Функции Стеценко и функции с запретными четверками и шестерками повторны в базисе $B_0 = \{\&, \vee, \neg\}$.*

Д о к а з а т е л ь с т в о. Воспользуемся тем фактом, что неповторная в базисе $B_0 = \{\&, \vee, \neg\}$ функция имеет равные остаточные функции, существенно зависящие от одного непустого множества переменных:

$$f_t^s \big|_{x_2=\dots=x_s=1} = x_1 x_2 \dots x_s \vee \bar{x}_1 \bar{x}_2 \dots \bar{x}_s \big|_{x_2=\dots=x_s=1} = x_1,$$

$$\begin{aligned}
& f_t^s \mid_{x_2=\dots=x_s=0} = x_1 x_2 \dots x_s \vee \bar{x}_1 \bar{x}_2 \dots \bar{x}_s \mid_{x_2=\dots=x_s=0} = \bar{x}_1, \\
& f_d^s \mid_{x_1=1, x_2=1, x_4=\dots=x_s=1} = \\
& = x_1 (\bar{x}_2 \vee x_3 x_4 \dots x_s) \vee \bar{x}_2 \bar{x}_3 \dots \bar{x}_s \mid_{x_1=1, x_2=1, x_4=\dots=x_s=1} = x_3, \\
& f_d^s \mid_{x_1=0, x_2=0, x_4=\dots=x_s=0} = \\
& = x_1 (\bar{x}_2 \vee x_3 x_4 \dots x_s) \vee \bar{x}_2 \bar{x}_3 \dots \bar{x}_s \mid_{x_1=0, x_2=0, x_4=\dots=x_s=0} = \bar{x}_3, \\
& f_m^s \mid_{x_3=\dots=x_s=0} = \\
& = x_1 (x_2 \vee \dots \vee x_s) \vee x_2 x_3 \dots x_s \mid_{x_3=\dots=x_s=0} = x_1 \cdot x_2, \\
& f_m^s \mid_{x_3=\dots=x_s=1} = \\
& = x_1 (x_2 \vee \dots \vee x_s) \vee x_2 x_3 \dots x_s \mid_{x_3=\dots=x_s=1} = x_1 \vee x_2, \\
& f_4 \mid_{x_2=x_4=0} = x_1 (x_2 \vee x_3) \vee x_3 x_4 \mid_{x_2=x_4=0} = x_1 \cdot x_3, \\
& f_4 \mid_{x_2=x_4=1} = x_1 (x_2 \vee x_3) \vee x_3 x_4 \mid_{x_2=x_4=1} = x_1 \vee x_3, \\
& f_5 \mid_{x_1=1, x_2=0, x_5=0} = \\
& = x_1 (x_2 \vee x_3 x_4) \vee x_5 (x_2 x_3 \vee x_4) \mid_{x_1=1, x_2=0, x_5=0} = x_3 \cdot x_4, \\
& f_5 \mid_{x_1=0, x_2=1, x_5=1} = \\
& = x_1 (x_2 \vee x_3 x_4) \vee x_5 (x_2 x_3 \vee x_4) \mid_{x_1=0, x_2=1, x_5=1} = x_3 \vee x_4.
\end{aligned}$$

Для каждой из функций Стеценко необходимое условие бесповторности нарушается. При наличии запретной четверки мы сразу имеем остаточные функции: переменную и ее отрицание. При наличии запретной шестерки, рассмотрев значения

на четвертых наборах квадратов, мы либо сразу получим остаточные функции вида $x_i \oplus x_j$ или $x_i \sim x_j$, либо получим пару функций $x_i^{\sigma_i} \vee x_j^{\sigma_j}$ и $x_i^{\sigma_i} \& x_j^{\sigma_j}$, которые также не равны. $\square$

Вопросы и задания.

1. Дайте определения следующих понятий:
 - а) преобразования обобщенной однотипности;
 - б) обобщенно однотипные функции;
 - б) слабоповторная функция;
 - б) запретная четверка;
 - б) запретная шестерка.
2. Опишите функции Стеценко.
3. Сформулируйте теорему о повторности функции Стеценко и функции с запретными четверками и шестерками.

7.11 Повторность. Свойство Субботовской

Т е о р е м а 43. Пусть функция $f(x_1, \dots, x_n)$ слабоповторна. Тогда у нее есть забывающая переменная.

Д о к а з а т е л ь с т в о. От противного. Пусть функция $f(x_1, \dots, x_n)$ — минимальная по числу переменных слабоповторная функция, не имеющая забывающей переменной. Построим для этой функции орграф G с множеством вершин

$\{x_1, \dots, x_n\}$ и дугами между теми вершинами x_i и x_j (обозначается $x_i \rightarrow x_j$), для которых существует соответствующая забивающая константа σ , т. е. $f(x_1, \dots, x_n) |_{x_i=\sigma}$ не зависит от x_j . Заметим, что из того, что $x_i \rightarrow x_j$ и $x_j \rightarrow x_k$, следует, что $x_i \rightarrow x_k$, т. е. отношение $\rightarrow$ транзитивно. Отсутствие незабивающей переменной функции $f(x_1, \dots, x_n)$ означает отсутствие вершины со степенью полуисхода ноль у графа G . Последнее означает наличие контура в графе G и наличие (в силу транзитивности) двух вершин в отношении $\rightarrow$. Не ограничивая общности, будем считать их вершинами x_1, x_2 , а соответствующие забивающие константы нулями:

$$\begin{aligned} f(0, x_2, \dots, x_n) &= g_1(x_3, \dots, x_n), \\ f(x_1, 0, x_3, \dots, x_n) &= g_2(x_3, \dots, x_n), \\ f(0, 0, x_3, \dots, x_n) &= g(x_3, \dots, x_n) = \\ &= g_1(x_3, \dots, x_n) = g_2(x_3, \dots, x_n). \end{aligned}$$

Таким образом:

$$\begin{aligned} f(0, 0, x_3, \dots, x_n) &= f(0, 1, x_3, \dots, x_n) = \\ &= f(1, 0, x_3, \dots, x_n) = g(x_3, \dots, x_n). \end{aligned}$$

Последнее означает, что $f(x_1, \dots, x_n) = \varphi(x_1 \& x_2, x_3, \dots, x_n)$. Если $\varphi(y_1, \dots, y_{n-1})$ — неповторная функция, то тогда и $f(x_1, \dots, x_n)$ — неповторная функция. С другой стороны, $f(1, x_2, \dots, x_n) = \varphi(x_2, x_3, \dots, x_n)$, т. е. φ должна быть неповторной в силу слабоповторности f . Полученное противоречие доказывает теорему. $\square$

З а м е ч а н и е 4. Теорема 43 была доказана в работе [14]. Здесь она приведена с учетом понятия слабоповторной функции.

7.12 Наличие запретных конфигураций у функции Субботовской

Рассмотрим слабоповторную функцию $f(x_1, \dots, x_n)$. По теореме Субботовской у этой функции имеется некоторая незабывающая переменная x_i . Рассмотрим бесповторные функции $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$ и $f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ и их канонические деревья D_0 и D_1 . Так как переменная x_i — забывающая, обе подфункции существенно зависят от всех $n - 1$ переменных. Если некоторая переменная x_j встречается в одном дереве с отрицанием, а в другом — без, то это означает наличие запретной четверки. В противном случае мы можем заменой переменных на их отрицания добиться монотонности функций $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$ и $f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$. Если в одном из деревьев D_0 и D_1 последняя общая вершина на пути в корень из некоторых листьев x_j и x_k помечена конъюнкцией, а в другом — дизъюнкцией, то это означает наличие запретной шестерки. Отсутствие такой ситуации означает, что $G(D_0) = G(D_1)$ и, следовательно, $D_0 = D_1$, то есть переменная x_i фиктивна, чего быть не может.

7.13 Теорема В.А. Стеценко

Вслед за В.А.Стеценко [15], пользуясь техникой работы [17], докажем следующее утверждение.

Т е о р е м а 44. С точностью до преобразований обобщенной однотипности существует всего пять семейств слабоповторных функций в элементарном базисе $B_0 = \{\&, \vee, \neg\}$, задаваемых формулами (7.6).

Пусть функция g является слабоповторной. Рассмотрим деревья ее подфункций, полученных константными подстановками на место забывающей переменной w . Возможны два случая.

1. Одно из этих деревьев содержит лист x_i , а другое — лист $\overline{x_i}$.
2. Для двух литералов-листьев u и v одного дерева первая общая вершина на пути в корень помечена конъюнкцией, а другого — дизъюнкцией.

Нетрудно проверить, что если ни один из случаев не выполнен, то деревья совпадут. Дальнейшие рассуждения проводятся методом неподвижной точки: показывается, что если g не является функцией Стеценко, то из нее константной подстановкой можно получить функцию с запретной четверкой или шестеркой и меньшим количеством существенных переменных.

Итак, пусть имеет место первый случай (наличие перемен-

ной и отрицания). Заменой переменных на отрицания всегда можно добиться, чтобы отрицания встречались лишь в дереве функции $g_0 = g|_{w=0}$. Обозначим переменные, встречающиеся при этом с отрицаниями? буквами x , а остальные – буквами y .

Л е м м а 27. Любая переменная y_i в одном из двух деревьев смежна с вершиной, помеченной конъюнкцией, а в другом – дизъюнкцией.

Д о к а з а т е л ь с т в о. В противном случае для переменной y_i найдется забывающая константа, общая для обеих подфункций, а функция g не будет слабоповторной в силу повторности ее подфункции. $\square$

Рассмотрим произвольный литерал, соответствующий переменной x_i в одном из деревьев. Рассмотрим все внутренние вершины дерева на пути из соответствующей ему вершины в корень. Назовем множество всех смежных с ними вершин *главной дорогой*.

Л е м м а 28. Любая переменная в любом из двух деревьев принадлежит всем главным дорогам.

Д о к а з а т е л ь с т в о. Подставим на место произвольной переменной z , лежащей вне главной дороги, определяемой литералом, соответствующим x_i , константу σ , забывающую в другом дереве. У получаемой подфункции переменная x_i останется существенной. При этом подставленная константа в си-

лу предыдущей леммы не будет забивающей для второй подфункции. Таким образом, функция $g|_{z=\sigma}$ повторна, а g — не слабоповторна. $\square$

Л е м м а 29. Никакие две переменные y_i и y_j не могут быть смежны с одной внутренней вершиной в каком-либо из деревьев.

Д о к а з а т е л ь с т в о. Предположим, что эти переменные в дереве 0-компоненты смежны с конъюнкцией. Тогда в дереве 1-компоненты по лемме 27 каждая из них смежна с дизъюнкцией. Если некоторая переменная x_k не смежна ни с одной из вышеупомянутых вершин, то константная подстановка на место x_k оставляет переменные y_i и y_j существенными, а соответствующую подстановке подфункцию — имеющей запретную шестерку. Если переменная x_k смежна с упомянутыми выше вершинами, то в силу противоположности забивающих констант для конъюнкции и дизъюнкции найдется подстановка $x_k = \sigma$, сохраняющая существенность y_i и y_j и соответствующую запретную шестерку.

Таким образом, деревья компонент функции g обязаны иметь форму «зонтика». Перенумеруем в обоих поддеревьях переменные y от вершины зонтика к корню. При этом в деревьях не должно быть пары переменных y_i и y_j , идущих в одинаковом порядке, так как иначе по аналогии с доказательством леммы 29 можно получить запретную шестерку. Если же порядки противоположны и число переменных типа y не

меньше трех, то запретную шестерку можно вновь получить по первой и третьей переменным y .

С учетом обобщенной однотипности мы можем положить функции в верхней вершине равными конъюнкции, а количество внутренних вершин дерева 0-компоненты не меньшим, чем 1-компоненты. При этом получается семь вариантов (по три для двух и одной переменной типа y и один — для нуля). Три варианта противоречат лемме 27. Оставшиеся четыре:

$$\begin{aligned} g_1 &= \bar{w}(y_2(y_1 \vee \bar{x}_1 \& \dots \& \bar{x}_n)) \vee w(y_1(y_2 \vee x_1 \& \dots \& x_n)), \\ g_2 &= \bar{w}(y_2 \vee y_1 \& \bar{x}_1 \& \dots \& \bar{x}_n) \vee w(y_1 \vee y_2 \& x_1 \& \dots \& x_n), \\ g_3 &= \bar{w}(y_1 \vee \bar{x}_1 \& \dots \& \bar{x}_n) \vee w \& y_1 \& x_1 \& \dots \& x_n, \\ g_4 &= \bar{w} \& \bar{x}_1 \& \dots \& \bar{x}_n \vee w \& x_1 \& \dots \& x_n. \end{aligned}$$

Функция g_4 обобщенно однотипна функции семейства f_t^{n+1} , функция g_3 — семейства f_d^{n+2} , а функция g_1 — семейства f_d^{n+3} . При $n = 1$ функция g_2 обобщенно однотипна функции f_d^4 , а при $n \geq 2$ подфункция $g_2|_{x_1=1}$ имеет запретную четверку по переменной w . Лемма полностью доказана. $\square$

Таким образом, рассмотрение первого случая полностью завершено, и мы переходим ко второму в предположении отсутствия запретных четверок у функции g . В случае отсутствия отрицаний у нас имеется запретная шестерка по некоторой паре переменных y_i и y_j . Поскольку функция g поляризуема, мы можем заменой части переменных на отрицания сделать ее монотонной. Справедливо следующее утверждение.

Л е м м а 30. Пусть функция g слабоповторна и моно-

тонна. В любом из деревьев компонент любая внутренняя вершина лежит либо на пути из корня в вершину y_i , либо на пути из корня в вершину y_j .

Д о к а з а т е л ь с т в о. Если некая внутренняя вершина u не лежит ни на одном из этих путей, то в поддереве с корнем u есть некая переменная z . Тогда забывающая для дерева другой компоненты подстановка $z = \sigma$ сохраняет запретную шестерку по переменным y_i и y_j . $\square$

Л е м м а 31. *Пусть функция g слабоповторна и монотонна. Пусть вершина u , отличная от y_i и y_j , смежна с корнем в одном из деревьев. Тогда найдется вершина v (необязательно отличная от y_i и y_j), такая, что имеется запретная шестерка по переменным u и v .*

Д о к а з а т е л ь с т в о. В самом деле, пусть корень, с которым смежна u , помечен конъюнкцией. Тогда если u связана с конъюнкцией и в другом дереве, то подстановка $u = 1$ сохраняет запретную шестерку по переменным y_i и y_j . Иначе имеется запретная шестерка по переменной u и произвольной переменной v , лежащей в поддереве с корнем — дизъюнкцией, смежной с u . $\square$

Назовем дерево прямым, если не более одного сына каждой внутренней вершины является внутренней вершиной. Из лемм 30 и 31 следует

Л е м м а 32. *Пусть функция g слабоповторна и монотонна. Либо оба дерева компонент прямые, либо корни обоих*

деревьев имеют степень два, и оба их поддеревья — прямые.

Л е м м а 33. *Пусть функция g слабоповторна и монотонна. Если корни обоих деревьев связаны с внутренними вершинами и имеют степень два, то функция g обобщенно однотипна с f_5 .*

Д о к а з а т е л ь с т в о. Рассмотрим дерево с корнем, помеченным конъюнкцией. В одном из поддеревьев в силу леммы 30, помимо переменной y_j , будет находиться переменная u , смежная с вершиной, смежной с корнем и помеченной дизъюнкцией, а в другом, помимо переменной y_j , — переменная v , смежная с вершиной, также смежной с корнем и помеченной дизъюнкцией.

Во втором дереве корень помечен дизъюнкцией, вершины u и v смежны с вершинами, помеченными конъюнкцией и находятся (во избежание наличия константной подстановки на место y_i (y_j), сохраняющей запретную шестерку по переменным u и y_j (v и y_i)) в одном поддереве с y_i и y_j соответственно. При этом имеется запретная шестерка по переменным u и v , которая сохраняется подстановкой константы на место одной из переменных y_i или y_j при наличии в деревьях еще хотя бы одной (четвертой) внутренней вершины.

Легко убедиться, что при отсутствии шестой переменной (кроме u, v, y_i, y_j и w), мы имеем функцию, обобщенно однотипную с f_5 . При наличии еще хотя бы одной (четвертой) внутренней вершины существует константная подстановка на ме-

сто переменной y_i или y_j , сохраняющая запретную шестерку по переменным u и v . Пусть внутренних вершин три и шестая переменная t и переменные u и y_j смежны в первом дереве с одной вершиной. Тогда если переменная t во втором дереве смежна с общей вершиной с переменными v и y_j , то подстановка $y_i = 0$ сохраняет запретную шестерку по переменным t и y_j , а если с переменными u и y_i , то подстановка $v = 0$ — по u и t . $\square$

Если корни в обоих деревьях являются единственными внутренними вершинами, то функция g обобщенно однотипна соответствующей функции семейства f_m . Пусть хотя бы в одном дереве имеется две (или более) внутренние вершины. Тогда по леммам 30 и 31 имеется запретная шестерка по переменным y_i и y_j , причем в одном дереве y_i смежна с корнем, а y_j с верхней вершиной, а в другом — наоборот.

Л е м м а 34. Пусть функция g слабоповторна и монотонна. Пусть оба дерева ее компонент — прямые. Тогда если внутренняя вершина не является верхней, то с ней смежна ровно одна переменная.

Д о к а з а т е л ь с т в о. В силу лемм 30 и 31 для любой переменной из пары переменных u, v , смежных с одной вершиной, будет иметься запретная шестерка по паре переменных, ее не содержащая. Поэтому переменные u и v во втором дереве связаны с внутренней вершиной, помеченной другим символом. Тогда имеется запретная шестерка по переменным

u, v , которая сохраняется при незабывающей для второго дерева подстановке на место переменной, связанной с верхней вершиной в первом дереве. $\square$

Л е м м а 35. Пусть функция g слабоповторна и монотонна. Пусть оба дерева ее компонент — прямые. Тогда ни одно из деревьев компонент не может иметь более трех внутренних вершин.

Д о к а з а т е л ь с т в о. От противного. Пусть 0-компонента имеет вид

$$y_1 \& (y_2 \vee y_3 \& (y_4 \vee h(y_5, \dots, y_n))).$$

Тогда в дереве 1-компоненты переменная y_1 связана с верхней вершиной, а переменная y_3 — с дизъюнкцией. Поэтому незабывающая для 1-компоненты подстановка константы на место переменной y_5 сохраняет запретную шестерку по переменным y_1 и y_3 . $\square$

С учетом доказанного теорема 44 будет вытекать из следующего утверждения.

Л е м м а 36. Пусть функция g слабоповторна и монотонна. Пусть оба дерева ее компонент — прямые и число внутренних вершин каждого из них не превосходит трех. Пусть с внутренними вершинами, отличными от верхней, смежна ровно одна переменная. Тогда g принадлежит семейству функций Стеценко.

Д о к а з а т е л ь с т в о. Если разность числа внутренних

вершин деревьев компонент равна единице, то верхние вершины этих деревьев помечены одним символом, и существует подстановка, сохраняющая запретную шестерку. Сохраняющая запретную шестерку подстановка также легко находится при соотношении внутренних вершин 3 : 1. Рассмотрим два оставшихся случая:

$$\begin{aligned} g_{22} &= y_1 \& (y_2 \vee \dots \vee y_n) \vee w(y_2 \vee y_1 \& y_3 \& \dots \& y_n), \quad n \geq 3, \\ g_{33} &= y_1 \& (y_2 \vee y_3 \& \dots \& y_n) \vee w(y_3 \vee y_2 \& (y_1 \vee y_4 \vee \dots \vee y_n)), \\ & \qquad \qquad \qquad n \geq 4. \end{aligned}$$

При $n = 3$ функция g_{22} обобщенно однотипна функции f_4 . При $n \geq 4$

$$g_{22} \mid_{y_n=0} = y_1 \& (y_2 \vee \dots \vee y_{n-1}) \vee w \& y_2.$$

Последняя подфункция имеет запретную шестерку по переменным y_1, y_2 . Функция g_{33} при $n = 4$ обобщенно однотипна функции f_5 . При $n \geq 5$

$$g_{33} \mid_{y_n=0} = y_1 \& y_2 \vee w(y_3 \vee y_2 \& (y_1 \vee y_4 \vee \dots \vee y_{n-1})).$$

Подстановка $y_1 = 0, y_3 = 0, y_4 = 1$ делает последнюю подфункцию равной $w \& y_2$, а подстановка $y_1 = 1, y_3 = 1$ — равной $w \vee y_2$. Таким образом, у подфункции $g_{33} \mid_{y_n=0}$ имеется запретная шестерка по переменным w, y_2 . $\square$

Доказательство леммы 36 полностью исчерпывает рассмотрение всех вариантов и доказательство теоремы 44.

Глава 8

Задачи

1. Для заданной функции построить сокращенную ДНФ.

Пусть задана функция $f(\mathbf{x}^4) = (1101\ 0101\ 0101\ 0111)$.

Применим метод Нельсона. Построим КНФ, покрывая множество нулей как можно меньшим числом граней. Занумеруем нули в таблице от 0 до 15, построим коды граней и дизъюнкции в КНФ.

Наборы	Коды	Скобки
2,6	(0-10)	$(x_1 \vee \bar{x}_3 \vee x_4)$
4,6	(01-0)	$(x_1 \vee \bar{x}_2 \vee x_4)$
2,10	(-010)	$(x_2 \vee \bar{x}_3 \vee x_4)$
8,12	(1-00)	$(\bar{x}_1 \vee x_3 \vee x_4)$

Перемножая первую скобку на вторую, а третью на четвертую, получим $(x_1 \vee \bar{x}_2 \bar{x}_3 \vee x_4)(\bar{x}_1 x_2 \vee \bar{x}_1 \bar{x}_3 \vee x_2 x_3 \vee x_4)$. Раскрывая скобки и поглощая слагаемые, получим $x_1 x_2 x_3 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_4$.

Решим ту же задачу методом карт Карно [22]. Представим

наборы в виде соответствующей таблицы.

		x_3	0	0	1	1
		x_4	0	1	1	0
x_1	x_2					
0	0		1	1	1	0
0	1		0	1	1	0
1	1		0	1	1	1
1	0		0	1	1	0

Конъюнкции x_4 соответствует прямоугольник площади восемь, а конъюнкциям $x_1x_2x_3$ и $\bar{x}_1\bar{x}_2\bar{x}_3$ – два прямоугольника площади два каждый.

Задания для самостоятельной работы.

Для заданной функции построить сокращенную ДНФ:

а) $f(\tilde{x}^4) = (0110 \ 1111 \ 1111 \ 0110);$

б) $f(\tilde{x}^4) = (0001 \ 1001 \ 0111 \ 0010);$

в) $f(\tilde{x}^4) = (0010 \ 0001 \ 0010 \ 0110);$

г) $f(\tilde{x}^4) = (1100 \ 1101 \ 1001 \ 1101);$

д) $f(\tilde{x}^4) = (1011 \ 0110 \ 0111 \ 0101) .$

2. Построить схему минимальной сложности в базисе $\{\&, \vee, \neg\}$.

Пусть задана функция $f(x, y, z) = (0100\ 0111)$. Формула и схема, соответствующие ДНФ Квайна функции $f(x, y, z)$, имеют вид $x \& y \vee \bar{y} \& z$ и сложность четыре.

Докажем минимальность. Функция $f(x, y, z)$ немонотонна, и поэтому схема должна содержать хотя бы одно отрицание. Функция существенно зависит от трех переменных, поэтому схема в силу связности должна содержать не менее двух двухвходовых элементов. Наконец, при наличии ровно одного отрицания и двух элементов из множества $\{\&, \vee\}$ схема и реализуемая ею функция будут неповторными, а функция $f(x, y, z)$ является повторной, так как содержит запретную четверку $f(100) = 0, f(110) = 1; f(001) = 1, f(011) = 0$.

Задания для самостоятельной работы.

Построить схему минимальной сложности для функций:

а) $f(\mathbf{x}^3) = (0001\ 0111);$

б) $f(\mathbf{x}^3) = (0001\ 1110);$

в) $f(\mathbf{x}^3) = (0111\ 1011);$

г) $f(\mathbf{x}^3) = (1001\ 0001);$

д) $f(\mathbf{x}^3) = (0110\ 1001).$

3. Построение тупиковых тестов.

Рассмотрим схему для медианы (см. рис. 3.3). Если неисправностью является замыкание любого одного контакта x , то в обоих случаях функция проводимости равна $y \vee xz$, проверяющий тест состоит из одного набора (010), диагностировать неисправность невозможно. Если неисправностью является размыкание любого одного контакта x , то в обоих случаях функция проводимости равна $y \& (x \vee z)$, проверяющий тест состоит из одного набора (101), диагностировать неисправность невозможно.

При замыкании контакта y между вершинами A и D реализуется функция $z \vee xy$. При замыкании контакта y между вершинами B и C реализуется функция $x \vee yz$. Проверяющим и диагностическим тестом является пара наборов $\{(001), (100)\}$. При размыкании контакта y между вершинами A и D реализуется функция $x \& (y \vee z)$. При замыкании контакта y между вершинами B и C реализуется функция $z \& (x \vee y)$. Проверяющим и диагностическим тестом является пара наборов $\{(011), (110)\}$.

Рассмотрим теперь общий алгоритм построения тупиковых

тестов для матриц. Рассмотрим матрицу

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Согласно алгоритму построения проверяющих тестов (см. п. 5.3), сложим по модулю два все столбцы с первым. Получим матрицу

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Для полученной матрицы решаем задачу о покрытии столбцов по алгоритму из п. 1.12. Получим формальное выражение

$$\begin{aligned} & (1 \vee 5)(1 \vee 2 \vee 3 \vee 6)(2 \vee 4 \vee 6)(3 \vee 4 \vee 5 \vee 6)(1 \vee 2 \vee 6) \times \\ & \times (2 \vee 3 \vee 6)(2 \vee 3 \vee 4 \vee 6) = (13 \vee 2 \vee 6)(23 \vee 25 \vee 4 \vee 6) \times \\ & \times (1 \vee 5) = (6 \vee 23 \vee 24 \vee 25 \vee 134)(1 \vee 5) = \\ & = 16 \vee 56 \vee 123 \vee 124 \vee 25 \vee 134. \end{aligned}$$

Первое преобразование получено перемножением второй, пятой, шестой и седьмой скобок, а также третьей и четвертой

скобок. Последняя запись содержит все шесть тупиковых проверяющих тестов.

Теперь без выписывания двадцативосьмистолбцовой матрицы попарных различий столбцов выпишем выражение для их различий в задаче построения диагностических тестов. Пары столбцов представлены скобками в лексикографическом порядке:

$$\begin{aligned}
 & (1 \vee 5)(1 \vee 2 \vee 3 \vee 6)(2 \vee 4 \vee 6)(3 \vee 4 \vee 5 \vee 6)(1 \vee 2 \vee 6) \times \\
 & \times (2 \vee 3 \vee 6)(2 \vee 3 \vee 4 \vee 6)(2 \vee 3 \vee 5 \vee 6)(1 \vee 2 \vee 4 \vee 5 \vee 6) \times \\
 & \times (1 \vee 3 \vee 4 \vee 6)(2 \vee 5 \vee 6)(1 \vee 2 \vee 3 \vee 5 \vee 6)(1 \vee 2 \vee 3 \vee 4 \vee 5 \vee 6) \times \\
 & \times (1 \vee 3 \vee 4)(1 \vee 2 \vee 4 \vee 5) 31(1 \vee 4) \times \\
 & \times (2 \vee 3 \vee 5)(1 \vee 4)(3 \vee 4) 3 \times \\
 & \times (1 \vee 2 \vee 3 \vee 4 \vee 5)(2 \vee 4 \vee 5)(2 \vee 5) \times \\
 & \times (1 \vee 3)(1 \vee 3 \vee 4) \times \\
 & \times 4.
 \end{aligned}$$

После вынесения выражения 134 и поглощения всех содержащих хотя бы один из этих множителей скобок получим выражение

$$134(2 \vee 5 \vee 6)(2 \vee 5) = 1234 \vee 1345.$$

Таким образом, мы получили два тупиковых диагностических теста.

Задачи для самостоятельной работы.

Построить все тупиковые диагностические тесты для матриц:

а)

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix};$$

б)

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

в)

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix};$$

г)

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix};$$

д)

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

4. Найти число решений уравнения

$$x \cdot y = x \cdot z,$$

где $u \cdot v$ — функция Линдона.

При $x \in \{0, 2, 3, 4\}$ обе части уравнения равны нулю. Среди таких наборов имеется $4 \times 7 \times 7 = 196$ решений. Для $x = 5$ функции равны 5 при втором аргументе из множества $\{2, 3, 4\}$, а иначе — равны нулю. Для $x = 6$ функции равны 6 при втором аргументе из множества $\{2, 3, 4\}$, а иначе — равны нулю.

Эти случаи соответствуют $2 \times (3 \cdot 3 + 4 \cdot 4) = 50$ решениям. При $x = 1$ либо $y, z \in \{0, 1, 5, 6\}$ и обе части уравнения равны нулю, либо $y = z$, что дает $4 \cdot 4 + 3 = 19$ вариантов решений. Итого: $196 + 50 + 19 = 265$.

Ответ: 265.

Задания для самостоятельной работы.

- а) найти число решений уравнения: $(x \cdot y) \cdot z = (z \cdot y) \cdot x$;
- б) найти число решений уравнения: $(x \cdot y) \cdot z = (z \cdot x) \cdot y$;
- в) найти число решений уравнения: $(x \cdot y) \cdot z = x \cdot (y \cdot z)$;
- г) найти число решений уравнения: $(z \cdot x) \cdot y = (z \cdot y) \cdot x$;
- д) найти число решений уравнения: $(x \cdot y) \cdot y = z \cdot (x \cdot y)$.

5. Свести ВЫП к 3-ВЫП.

Преобразуем КНФ

$$(x_1 \vee x_2)(x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4)(\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee \bar{x}_5)$$

по правилам, введенным в доказательстве теоремы 29:

$$\begin{aligned} & (x_1 \vee x_2)(x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4)(\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee \bar{x}_5) \leftrightarrow \\ & \leftrightarrow (x_1 \vee x_2 \vee \bar{y}_1)(x_1 \vee x_2 \vee y_1)(x_1 \vee \bar{x}_2 \vee \bar{y}_2)(y_2 \vee x_3 \vee \bar{x}_4)(\bar{x}_1 \vee \\ & \vee x_2 \vee \bar{y}_3)(y_3 \vee \bar{x}_3 \vee \bar{y}_4)(y_4 \vee x_4 \vee \bar{x}_5). \end{aligned}$$

Задания для самостоятельной работы.

Свести ВЫП к 3-ВЫП:

- а) $(x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2 \vee x_3 \vee x_4 \vee \bar{x}_5);$
 б) $(x_1 \vee \bar{x}_2 \vee x_3 \vee x_4)(\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4);$
 в) $(x_1 \vee x_2)(\bar{x}_1 \vee x_2 \vee \bar{x}_3)(\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4 \vee \bar{x}_5 \vee x_6);$
 г) $\bar{x}_4(\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee x_5 \vee x_6 \vee x_7);$
 д) $(x_1 \vee x_3)(x_1 \vee \bar{x}_2)(x_2 \vee \bar{x}_3)(\bar{x}_1 \vee \bar{x}_2 \vee x_3).$

6. Свести ВЫП к КЛИКА.

Рассмотрим КНФ

$$(x_1 \vee x_2 \vee \bar{x}_3)(\bar{x}_1 \vee x_2)(x_1 \vee \bar{x}_2 \vee x_3 \vee x_4).$$

По теореме 28 задача выполнимости этой КНФ сводится к задаче КЛИКА, для которой граф G на входе имеет множество вершин $V(G) = \{x_1^1, x_2^1, \bar{x}_3^1, \bar{x}_1^2, x_2^2, x_1^3, \bar{x}_2^3, x_3^3, x_4^3\}$ (нижний индекс соответствует номеру переменной, верхний — номеру скобки) и множество ребер

$$\begin{aligned} E(G) = \{ & (x_1^1, x_2^2), (x_1^1, x_1^3), (x_1^1, \bar{x}_2^3), (x_1^1, x_3^3), (x_1^1, x_4^3), \\ & (x_2^1, \bar{x}_1^2), (x_2^1, x_2^2), (x_2^1, x_1^3), (x_2^1, x_3^3), (x_2^1, x_4^3), \\ & (\bar{x}_3^1, \bar{x}_1^2), (\bar{x}_3^1, x_2^2), (\bar{x}_3^1, x_1^3), (\bar{x}_3^1, \bar{x}_2^3), (\bar{x}_3^1, x_4^3), \\ & (\bar{x}_1^2, \bar{x}_2^3), (\bar{x}_1^2, x_3^3), (\bar{x}_1^2, x_4^3), \\ & (x_2^2, x_1^3), (x_2^2, x_3^3), (x_2^2, x_4^3)\}, \end{aligned}$$

а параметр k равен трем.

Задания для самостоятельной работы.

Свести ВЫП к КЛИКА:

$$\text{а)} \quad (x_1 \vee \bar{x}_3 \vee \bar{x}_4)(x_1 \vee \bar{x}_3)(\bar{x}_1 \vee \bar{x}_2)(x_1 \vee x_2 \vee x_3);$$

$$\text{б)} \quad (\bar{x}_1 \vee x_3)(x_3 \vee \bar{x}_4)(x_1 \vee \bar{x}_2 \vee \bar{x}_3)(\bar{x}_1 \vee x_2 \vee \bar{x}_3);$$

$$\text{в)} \quad (x_1 \vee x_3 \vee x_4)(x_2 \vee \bar{x}_1)(x_1 \vee \bar{x}_2 \vee x_4)(x_1 \vee \bar{x}_2 \vee x_3);$$

$$\text{г)} \quad (\bar{x}_2 \vee \bar{x}_3)(x_2 \vee \bar{x}_3)(x_3 \vee \bar{x}_1)(x_1 \vee \bar{x}_2 \vee \bar{x}_3);$$

$$\text{д)} \quad (x_3 \vee \bar{x}_4) \cdot \bar{x}_2 \cdot (x_1 \vee x_2 \vee \bar{x}_4)(\bar{x}_1 \vee x_2 \vee x_3).$$

7. Решить задачу 2–ВЫП последовательным изъятием переменных. Пусть дано 2-КНФ

$$(x_1 \vee x_2)(x_1 \vee \bar{x}_3)(\bar{x}_1 \vee x_5)(x_2 \vee x_3)(\bar{x}_2 \vee x_4)(x_2 \vee \bar{x}_5) \cdot$$

$$\cdot (\bar{x}_3 \vee \bar{x}_4)(x_3 \vee x_5)(x_4 \vee \bar{x}_5).$$

Изымем переменную x_2 . Собирая содержащие x_2 выражения, получим

$$(x_2 \vee x_1 x_3 \bar{x}_5)(\bar{x}_2 \vee x_4) \leftrightarrow (x_1 \vee x_4)(x_3 \vee x_4)(\bar{x}_5 \vee x_4).$$

Получаем следующую 2-КНФ, свободную от переменной x_2 и эквивалентную исходной по выполнимости:

$$(x_1 \vee x_4)(x_3 \vee x_4)(\bar{x}_5 \vee x_4)(x_1 \vee \bar{x}_3)(\bar{x}_1 \vee x_5)(\bar{x}_3 \vee \bar{x}_4)(x_3 \vee x_5)(x_4 \vee \bar{x}_5).$$

Изымем теперь переменную x_4 . Собирая содержащие x_4 выражения, получим

$$(x_4 \vee x_1 x_3 \bar{x}_5)(\bar{x}_4 \vee \bar{x}_3) \leftrightarrow (x_1 \vee \bar{x}_3)(\bar{x}_5 \vee \bar{x}_3).$$

Итого имеем

$$(x_1 \vee \bar{x}_3)(\bar{x}_1 \vee x_5)(x_3 \vee x_5)(x_1 \vee \bar{x}_3)(\bar{x}_5 \vee \bar{x}_3).$$

Изымем переменную x_1 . Собирая содержащие x_1 выражения, получим

$$(x_1 \vee \bar{x}_3)(\bar{x}_1 \vee x_5) \leftrightarrow (\bar{x}_3 \vee x_5).$$

Итого имеем

$$(\bar{x}_3 \vee x_5)(x_3 \vee x_5)(\bar{x}_5 \vee \bar{x}_3).$$

Изымем переменную x_3 . Собирая содержащие x_3 выражения, получим

$$(x_3 \vee x_5)(\bar{x}_3 \vee x_5 \bar{x}_5) = (x_3 \vee x_5)\bar{x}_3,$$

откуда $x_3 = 0$, а $x_5 = 1$ — формула выполнима.

Задания для самостоятельной работы.

Решить задачу 2–ВЫП последовательным изъятием переменных:

- а) $(\bar{x}_3 \vee \bar{x}_4)(x_1 \vee \bar{x}_3)(\bar{x}_1 \vee \bar{x}_2)(x_1 \vee x_3)(\bar{x}_1 \vee \bar{x}_3)(x_2 \vee \bar{x}_3)(\bar{x}_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_3);$
- б) $(\bar{x}_1 \vee x_3)(x_3 \vee \bar{x}_4)(x_1 \vee \bar{x}_2)(x_2 \vee \bar{x}_3)(x_2 \vee \bar{x}_4)(\bar{x}_1 \vee x_4)(\bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2);$
- в) $(x_4 \vee x_3)(x_3 \vee \bar{x}_1)(\bar{x}_2 \vee x_4)(x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2)(x_2 \vee \bar{x}_3)(\bar{x}_1 \vee \bar{x}_4);$
- г) $(\bar{x}_2 \vee \bar{x}_3)(x_2 \vee \bar{x}_3)(x_2 \vee \bar{x}_4)(x_1 \vee \bar{x}_4)(\bar{x}_1 \vee \bar{x}_3)(\bar{x}_2 \vee \bar{x}_4)(x_3 \vee x_4);$

$$\text{д) } (x_1 \vee x_3)(\bar{x}_3 \vee \bar{x}_5)(x_5 \vee x_2)(\bar{x}_2 \vee \bar{x}_4)(\bar{x}_4 \vee x_1)(\bar{x}_1 \vee \bar{x}_2)(x_2 \vee x_3).$$

8. Построить кратчайшее остовное дерево для заданного графа.

Пусть дана матрица смежности графа

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	$\times$	84	12	57	93	43
v_2	84	$\times$	33	17	28	10
v_3	12	33	$\times$	99	76	69
v_4	57	17	99	$\times$	44	38
v_5	93	28	76	44	$\times$	27
v_6	43	10	69	38	27	$\times$

Будем действовать в соответствии с алгоритмом.

1. Берем кратчайшее ребро (v_2, v_6) длины десять и добавляем его к пустому графу.
2. Добавляем следующее по длине (12) ребро (v_1, v_3) .
3. Добавляем третье по длине (17) ребро (v_2, v_4) .
4. Следующим добавляем ребро (v_5, v_6) длины 27.
5. Добавляем ребро (v_2, v_3) веса 33.

Построено кратчайшее дерево $(v_1, v_3), (v_2, v_3), (v_2, v_6), (v_2, v_4), (v_5, v_6)$ с суммарной длиной ребер, равной 99.

Задания для самостоятельной работы.

Построить кратчайшее остовное дерево для заданного графа:

а)

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	×	22	53	95	76	86
v_2	22	×	74	14	80	52
v_3	53	74	×	20	92	41
v_4	95	14	20	×	62	56
v_5	76	80	92	62	×	37
v_6	86	52	41	56	37	×

б)

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	×	38	98	41	81	83	77
v_2	38	×	59	26	89	18	85
v_3	98	59	×	49	76	79	64
v_4	41	26	49	×	71	89	73
v_5	81	89	76	71	×	77	86
v_6	83	18	79	89	77	×	61
v_7	77	85	64	73	86	61	×

B)

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	$\times$	74	83	10	43	73	87	80
v_2	74	$\times$	56	82	58	62	62	39
v_3	83	56	$\times$	89	85	45	91	96
v_4	10	82	89	$\times$	53	11	82	44
v_5	43	58	85	53	$\times$	71	96	47
v_6	73	62	45	11	71	$\times$	91	15
v_7	87	62	91	82	96	91	$\times$	50
v_8	80	39	96	44	47	15	50	$\times$

;

Г)

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
v_1	$\times$	99	27	35	98	79	32	96	71
v_2	99	$\times$	26	17	32	75	92	84	59
v_3	27	26	$\times$	20	75	79	37	73	45
v_4	35	17	20	$\times$	73	36	72	96	86
v_5	98	32	75	73	$\times$	60	52	75	67
v_6	79	75	79	36	60	$\times$	25	81	80
v_7	32	92	37	72	52	25	$\times$	25	93
v_8	96	84	73	96	75	81	25	$\times$	23
v_9	71	59	45	86	67	80	93	23	$\times$

;

д)

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_1	$\times$	25	41	46	34	55	16	17	19	78
v_2	25	$\times$	—	40	79	58	96	53	—	38
v_3	41	—	$\times$	73	33	59	63	—	56	23
v_4	46	40	73	$\times$	—	38	—	68	—	52
v_5	34	79	33	—	$\times$	14	50	66	85	51
v_6	55	58	59	38	14	$\times$	—	14	55	54
v_7	16	96	63	—	50	—	$\times$	64	14	54
v_8	17	53	—	68	66	14	64	$\times$	17	25
v_9	19	—	56	—	85	55	14	17	$\times$	65
v_{10}	78	38	23	52	51	54	54	25	65	$\times$

9. Найти запретную конфигурацию для функции $f = (x_1 \vee \bar{x}_2)(\bar{x}_2 \vee x_3)(x_3 \vee \bar{x}_4)$.

Функция f поляризуема, поэтому надо искать запретную шестерку. Подставляя $x_1 = 0, x_4 = 1$, получим $\bar{x}_2 x_3$, а подставляя $x_1 = 1, x_4 = 0$, получим $\bar{x}_2 \vee x_3$. Таким образом, запретная шестерка наборов имеет вид: $f(0011) = 1, f(0001) = 0, f(0111) = 0$ и $f(1100) = 0, f(1110) = 1, f(1000) = 1$.

Задачи для самостоятельной работы.

Найти запретную конфигурацию для функции f , где

а) $f = x_3(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_4) \vee \bar{x}_1 \bar{x}_2 \bar{x}_4,$

б) $f = (\bar{x}_1 \vee x_2 x_5)(x_2 \vee x_3 x_5)(x_3 \vee \bar{x}_4),$

$$\text{в)} \quad f = (x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee \bar{x}_3),$$

$$\text{г)} \quad f = x_1 \cdot (\bar{x}_2 \vee x_3 \bar{x}_4) \vee \bar{x}_2 \bar{x}_3 x_4,$$

$$\text{д)} \quad f = \bar{x}_2 \cdot (x_1 \vee x_3) \vee x_3 \cdot \bar{x}_4.$$

10. Построение всех тупиковых ДНФ функции по сокращенной.

Рассмотрим функцию $g(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$. Ее сокращенная ДНФ имеет вид

$$x_1 \bar{x}_2 \vee x_2 \bar{x}_3 \vee x_3 \bar{x}_1 \vee x_1 \bar{x}_3 \vee x_3 \bar{x}_2 \vee x_2 \bar{x}_1.$$

Рассмотрим таблицу Квайна:

	(001)	(010)	(011)	(100)	(101)	(110)
$K_1 = x_1 \bar{x}_2$	0	0	0	1	1	0
$K_2 = x_2 \bar{x}_3$	0	1	0	0	0	1
$K_3 = x_3 \bar{x}_1$	1	0	1	0	0	0
$K_4 = x_1 \bar{x}_3$	0	0	0	1	0	1
$K_5 = x_3 \bar{x}_2$	1	0	0	0	1	0
$K_6 = x_2 \bar{x}_1$	0	1	1	0	0	0

Применим алгоритм построения всех тупиковых покрытий данной таблицы. Получим формальное выражение

$$(K_3 \vee K_5)(K_2 \vee K_6)(K_3 \vee K_6)(K_1 \vee K_4)(K_1 \vee K_5)(K_2 \vee K_4).$$

Перемножая скобки: первую на третью, вторую на шестую и

четвертую на пятую, имеем

$$(K_3 \vee K_5 K_6)(K_2 \vee K_4 K_6)(K_1 \vee K_4 K_5).$$

После раскрытия скобок и поглощений получим

$$K_1 K_2 K_3 \vee K_4 K_5 K_6 \vee K_1 K_2 K_5 K_6 \vee K_1 K_3 K_4 K_6 \vee K_2 K_3 K_4 K_5.$$

Слагаемым этой формальной ДНФ соответствуют тупиковые ДНФ функции $g(x_1, x_2, x_3)$ таблицы:

Слагаемое	ДНФ	ДНФ
$K_1 K_2 K_3$	$K_1 \vee K_2 \vee K_3$	$x_1 \bar{x}_2 \vee x_2 \bar{x}_3 \vee x_3 \bar{x}_1$
$K_4 K_5 K_6$	$K_4 \vee K_5 \vee K_6$	$x_1 \bar{x}_3 \vee x_3 \bar{x}_2 \vee x_2 \bar{x}_1$
$K_1 K_2 K_5 K_6$	$K_1 \vee K_2 \vee K_5 \vee K_6$	$x_1 \bar{x}_2 \vee x_2 \bar{x}_3 \vee x_3 \bar{x}_2 \vee x_2 \bar{x}_1$
$K_1 K_3 K_4 K_6$	$K_1 \vee K_3 \vee K_4 \vee K_6$	$x_1 \bar{x}_2 \vee x_3 \bar{x}_1 \vee x_1 \bar{x}_3 \vee x_2 \bar{x}_1$
$K_2 K_3 K_4 K_5$	$K_2 \vee K_3 \vee K_4 \vee K_5$	$x_2 \bar{x}_3 \vee x_3 \bar{x}_1 \vee x_1 \bar{x}_3 \vee x_3 \bar{x}_2$

Таким образом, функция $g(x_1, x_2, x_3)$ имеет пять тупиковых ДНФ, среди которых две минимальных.

Задачи для самостоятельной работы.

Построить все тупиковые ДНФ функции по сокращенной:

- а) $f(\tilde{x}^4) = x_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee \bar{x}_3 x_4 \vee x_3 \bar{x}_4;$
- б) $f(\tilde{x}^4) = \bar{x}_1 x_3 x_4 \vee x_1 x_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 x_4 \vee \bar{x}_2 x_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4;$
- в) $f(\tilde{x}^4) = x_1 x_3 \bar{x}_4 \vee \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4;$
- г) $f(\tilde{x}^4) = \bar{x}_1 \bar{x}_3 \vee x_2 \bar{x}_3 \vee x_2 x_4 \vee \bar{x}_3 \bar{x}_4 \vee x_1 x_3 x_4;$

$$\text{д)} \quad f(\tilde{x}^4) = x_1x_4 \vee \bar{x}_2x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_4 \vee \bar{x}_1x_3\bar{x}_4 \vee x_2\bar{x}_3x_4.$$

Ответы к задачам

1. а) $x_1\bar{x}_2 \vee \bar{x}_1x_2 \vee \bar{x}_3x_4 \vee x_3\bar{x}_4$;
 б) $\bar{x}_1x_3x_4 \vee x_1x_3\bar{x}_4 \vee x_1\bar{x}_2x_3 \vee x_1\bar{x}_2x_4 \vee \bar{x}_2x_3x_4 \vee \bar{x}_1x_2\bar{x}_3\bar{x}_4$;
 в) $x_1x_3\bar{x}_4 \vee \bar{x}_2x_3\bar{x}_4 \vee x_1x_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3x_4$;
 г) $\bar{x}_1\bar{x}_3 \vee x_2\bar{x}_3 \vee x_2x_4 \vee \bar{x}_3\bar{x}_4 \vee x_1x_3x_4$;
 д) $x_1x_4 \vee \bar{x}_2x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_4 \vee \bar{x}_1x_3\bar{x}_4 \vee x_2\bar{x}_3x_4$.
2. а) $x_1 \cdot (x_2 \vee x_3) \vee x_2 \cdot x_3$;
 б) схема $x_1 \oplus x_2 \cdot x_3$, см. рис. 8.1 слева;
 в) $(x_1 \vee x_3) \cdot \overline{(x_1 \wedge x_3)} \vee x_2$;
 г) $x_2 \cdot x_3 \vee \overline{(x_1 \vee x_2 \vee x_3)}$;
 д) схема $x_1 \oplus x_2 \oplus x_3$, см. рис. 8.1 справа;
3. а) $\{1346, 3456\}$;
 б) $\{1234, 1345, 1346, 1456, 2345, 2456\}$;
 в) $\{12345, 23456\}$;
 г) $\{1245, 1345, 1456, 1356, 1234, 1236\}$;
 д) $\{1234, 12456, 1345\}$.
4. а) 289;

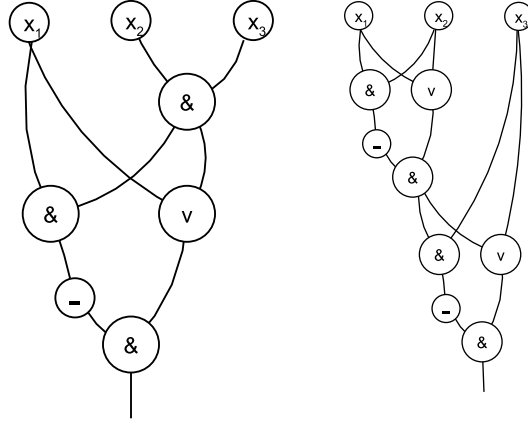


Рис. 8.1 Ответ к задачам 2б, 2д.

- б) 289;
- в) 316;
- г) 342;
- д) 286.
5. а) $(x_1 \vee \bar{x}_2 \vee y_1)(x_1 \vee \bar{x}_2 \vee \bar{y}_1)(\bar{x}_1 \vee x_2 \vee \bar{y}_2)(y_2 \vee x_3 \vee \bar{y}_3)(y_3 \vee \vee x_4 \vee \bar{x}_5);$
- б) $(x_1 \vee \bar{x}_2 \vee \bar{y}_1)(y_1 \vee x_3 \vee x_4)(\bar{x}_1 \vee x_2 \vee \bar{y}_2)(y_2 \vee \bar{x}_3 \vee \bar{x}_4);$
- в) $(x_1 \vee x_2 \vee \bar{y}_1)(x_1 \vee x_2 \vee y_1)(\bar{x}_1 \vee x_2 \vee \bar{x}_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{y}_2)(y_2 \vee \vee x_3 \vee \bar{y}_3)(y_3 \vee \bar{x}_4 \vee \bar{y}_4)(y_4 \vee \bar{x}_5 \vee x_6);$
- г) $(\bar{x}_4 \vee \bar{y}_1 \vee \bar{y}_2)(\bar{x}_4 \vee \bar{y}_1 \vee y_2)(\bar{x}_4 \vee y_1 \vee \bar{y}_2)(\bar{x}_4 \vee y_1 \vee y_2)(\bar{x}_1 \vee \vee x_2 \vee \bar{y}_3)(y_3 \vee \bar{x}_3 \vee \bar{y}_4)(y_4 \vee x_4 \vee \bar{y}_5)(y_5 \vee x_5 \vee \bar{y}_6)(y_6 \vee x_6 \vee x_7);$
- д) $(x_1 \vee x_3 \vee \bar{y}_1)(x_1 \vee x_3 \vee y_1)(x_1 \vee \bar{x}_2 \vee \bar{y}_2)(x_1 \vee \bar{x}_2 \vee y_2)(x_2 \vee \vee \bar{x}_3 \vee \bar{y}_3)(x_2 \vee \bar{x}_3 \vee y_3)(\bar{x}_1 \vee \bar{x}_2 \vee x_3).$

6. а)

$$\begin{aligned}
 E(G) = \{ & (x_1^1, x_1^2), (x_1^1, \bar{x}_3^2), (x_1^1, \bar{x}_2^3), (x_1^1, x_1^4), \\
 & (x_1^1, x_2^4), (x_1^1, x_3^4), (\bar{x}_3^1, x_2^1), (\bar{x}_3^1, \bar{x}_3^2), (\bar{x}_3^1, \bar{x}_1^3), \\
 & (\bar{x}_3^1, \bar{x}_2^3), (\bar{x}_3^1, x_1^4), (\bar{x}_3^1, x_2^4), (\bar{x}_4^1, x_2^1), (\bar{x}_4^1, \bar{x}_3^2), \\
 & (\bar{x}_4^1, \bar{x}_1^3), (\bar{x}_4^1, \bar{x}_2^3), (\bar{x}_4^1, x_1^4), (\bar{x}_4^1, x_2^4), (\bar{x}_4^1, x_3^4), \\
 & (x_1^2, \bar{x}_2^3), (x_1^2, x_1^4), (x_1^2, x_2^4), (x_1^2, x_3^4), (\bar{x}_3^2, \bar{x}_1^3), \\
 & (\bar{x}_3^2, \bar{x}_2^3), (\bar{x}_3^2, x_1^4), (\bar{x}_3^2, x_2^4), (\bar{x}_1^3, x_2^4), (\bar{x}_1^3, x_3^4), \\
 & (\bar{x}_2^3, x_1^4), (\bar{x}_2^3, x_3^4)\},
 \end{aligned}$$

а параметр k равен четырем;

б)

$$\begin{aligned}
 E(G) = \{ & (\bar{x}_1^1, x_3^2), (\bar{x}_1^1, \bar{x}_4^2), (\bar{x}_1^1, \bar{x}_2^3), (\bar{x}_1^1, \bar{x}_3^3), (\bar{x}_1^1, \bar{x}_1^4), \\
 & (\bar{x}_1^1, x_2^4), (\bar{x}_1^1, \bar{x}_3^4), (x_3^1, x_3^2), (x_3^1, \bar{x}_4^2), (x_3^1, x_1^3), (x_3^1, \bar{x}_2^3), \\
 & (x_3^1, \bar{x}_1^4), (x_3^1, x_2^4), (x_3^2, x_1^3), (x_3^2, \bar{x}_2^3), (x_3^2, \bar{x}_1^4), (x_3^2, x_2^4), \\
 & (\bar{x}_4^2, x_1^3), (\bar{x}_4^2, \bar{x}_2^3), (\bar{x}_4^2, \bar{x}_3^3), (\bar{x}_4^2, \bar{x}_1^4), (\bar{x}_4^2, x_2^4), \\
 & (\bar{x}_4^2, \bar{x}_3^4), (x_1^3, x_2^4), (x_1^3, \bar{x}_3^4), (\bar{x}_2^3, \bar{x}_1^4), (\bar{x}_2^3, \bar{x}_3^4), \\
 & (\bar{x}_3^3, \bar{x}_1^4), (\bar{x}_3^3, x_2^4), (\bar{x}_3^3, \bar{x}_1^4)\},
 \end{aligned}$$

а параметр k равен четырем;

в)

$$\begin{aligned}
E(G) = \{ & (x_1^1, x_2^2), (x_1^1, x_3^3), (x_1^1, \bar{x}_2^3), (x_1^1, x_4^4), (x_1^1, x_1^4), \\
& (x_1^1, \bar{x}_2^4), (x_1^1, x_3^4), (x_3^1, x_2^2), (x_3^1, \bar{x}_1^2), (x_3^1, x_1^3), \\
& (x_3^1, \bar{x}_2^3), (x_3^1, x_4^4), (x_3^1, x_1^4), (x_3^1, \bar{x}_2^4), (x_3^1, x_3^4), \\
& (x_4^1, x_2^2), (x_4^1, \bar{x}_1^2), (x_4^1, x_1^3), (x_4^1, \bar{x}_2^3), (x_4^1, x_4^4), \\
& (x_4^1, x_1^4), (x_4^1, \bar{x}_2^4), (x_4^1, x_3^4), (x_2^2, x_1^3), (x_2^2, x_4^4), \\
& (x_2^2, x_1^4), (x_2^2, x_3^4), (\bar{x}_1^2, \bar{x}_2^3), (\bar{x}_1^2, x_4^4), (\bar{x}_1^2, \bar{x}_2^4), \\
& (\bar{x}_1^2, x_3^4), (x_1^3, x_4^4), (x_1^3, \bar{x}_2^4), (x_1^3, x_3^4), (\bar{x}_2^3, x_1^4), \\
& (\bar{x}_2^3, \bar{x}_2^4), (\bar{x}_2^3, x_3^4), (x_4^3, x_1^4), (x_4^3, \bar{x}_2^4), (x_4^3, x_3^4) \},
\end{aligned}$$

а параметр k равен четырем;

г)

$$\begin{aligned}
E(G) = \{ & (\bar{x}_2^1, \bar{x}_3^2), (\bar{x}_2^1, x_3^3), (\bar{x}_2^1, \bar{x}_1^3), (\bar{x}_2^1, x_1^4), (\bar{x}_2^1, \bar{x}_2^4), \\
& (\bar{x}_2^1, \bar{x}_3^4), (\bar{x}_3^1, x_2^2), (\bar{x}_3^1, \bar{x}_2^3), (\bar{x}_3^1, \bar{x}_1^3), (\bar{x}_3^1, x_1^4), (\bar{x}_3^1, \bar{x}_2^4), \\
& (\bar{x}_3^1, \bar{x}_3^4), (x_2^2, x_3^3), (x_2^2, \bar{x}_1^3), (x_2^2, x_1^4), (x_2^2, \bar{x}_3^4), \\
& (\bar{x}_3^2, \bar{x}_1^3), (\bar{x}_3^2, x_1^4), (\bar{x}_3^2, \bar{x}_2^4), (\bar{x}_3^2, \bar{x}_3^4), (x_3^3, x_1^4), \\
& (x_3^3, \bar{x}_2^4), (\bar{x}_1^3, \bar{x}_2^4), (\bar{x}_1^3, \bar{x}_3^4) \},
\end{aligned}$$

а параметр k равен четырем;

д)

$$\begin{aligned}
E(G) = \{ & (x_3^1, \bar{x}_2^2), (x_3^1, x_1^3), (x_3^1, x_2^3), (x_3^1, \bar{x}_4^3), (x_3^1, \bar{x}_1^4), \\
& (x_3^1, x_2^4), (x_3^1, x_3^4), (\bar{x}_4^1, \bar{x}_2^2), (\bar{x}_4^1, x_1^3), (\bar{x}_4^1, x_2^3), \\
& (\bar{x}_4^1, \bar{x}_4^3), (\bar{x}_4^1, \bar{x}_1^4), (\bar{x}_4^1, x_2^4), (\bar{x}_4^1, x_3^4), \\
& (\bar{x}_2^2, x_1^3), (\bar{x}_2^2, \bar{x}_4^3), (\bar{x}_2^2, \bar{x}_1^4), (\bar{x}_2^2, x_3^4), (x_1^3, x_2^4), \\
& (x_1^3, x_3^4), (x_2^3, \bar{x}_1^4), (x_2^3, x_2^4), (x_2^3, x_3^4), \\
& (\bar{x}_4^3, \bar{x}_1^4), (\bar{x}_4^3, x_2^4), (\bar{x}_4^3, x_3^4) \},
\end{aligned}$$

а параметр k равен четырем.

7. а) невыполнима;
 б) выполняема;
 в) выполняема;
 г) невыполнима;
 д) невыполнима.

8. а) $(v_2, v_4), (v_3, v_4), (v_1, v_2), (v_5, v_6), (v_3, v_6)$ — ребра остоного дерева, 134 — суммарная длина ребер кратчайшего остоного дерева;
 б) $(v_2, v_6), (v_2, v_4), (v_1, v_2), (v_3, v_4), (v_6, v_7), (v_4, v_5)$ — ребра остоного дерева, 263 — суммарная длина ребер кратчайшего остоного дерева;
 в) $(v_1, v_4), (v_4, v_6), (v_6, v_8), (v_1, v_5), (v_2, v_8), (v_3, v_6), (v_7, v_8)$ — ребра остоного дерева, 213 — суммарная длина ребер кратчайшего остоного дерева;
 г) $(v_2, v_4), (v_3, v_4), (v_8, v_9), (v_7, v_8), (v_6, v_7), (v_1, v_3), (v_2, v_5), (v_1, v_7)$ — ребра остоного дерева, 201 — суммарная длина ребер кратчайшего остоного дерева;
 д) $(v_5, v_6), (v_6, v_8), (v_7, v_9), (v_1, v_7), (v_1, v_8), (v_3, v_{10}), (v_1, v_2), v_8, v_{10}), (v_2, v_4)$ — ребра остоного дерева, 188 — суммарная длина ребер кратчайшего остоного дерева;

9. а) $\bar{x}_1 \cdot \bar{x}_2 = f(x_1, x_2, 0, 0), \quad \bar{x}_1 \vee \bar{x}_2 = f(x_1, x_2, 1, 1);$

$$\text{б)} \quad x_2 \cdot x_5 = f(1, x_2, 0, 1, x_5), \quad x_2 \vee x_5 = f(0, x_2, 1, 1, x_5);$$

$$\text{в)} \quad x_1 = f(x_1, 1, 0), \quad \bar{x}_1 = f(x_1, 0, 1);$$

$$\text{г)} \quad x_3 = f(1, 1, x_3, 0), \quad \bar{x}_3 = f(0, 0, x_3, 1);$$

$$\text{д)} \quad \bar{x}_2 \cdot x_3 = f(0, x_2, x_3, 1), \quad \bar{x}_2 \vee x_3 = f(1, x_2, x_3, 0).$$

$$10. \text{ а)} \quad x_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee \bar{x}_3 x_4 \vee x_3 \bar{x}_4;$$

$$\text{б)} \quad \bar{x}_1 x_3 x_4 \vee x_1 x_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4;$$

$$\text{в)} \quad x_1 x_3 \bar{x}_4 \vee \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4;$$

$$\text{г)} \quad \bar{x}_1 \bar{x}_3 \vee x_2 x_4 \vee \bar{x}_3 \bar{x}_4 \vee x_1 x_3 x_4;$$

$$\text{д)} \quad x_1 x_4 \vee \bar{x}_2 x_3 \vee \bar{x}_1 \bar{x}_2 \bar{x}_4 \vee \bar{x}_1 x_3 \bar{x}_4 \vee x_2 \bar{x}_3 x_4.$$

Вопросы к зачету по курсу «Основы кибернетики»

(4 курс, осенний семестр 2017–2018 уч. года)

1. Определение дизъюнктивных нормальных форм (ДНФ). Геометрическая интерпретация ДНФ. Совершенная ДНФ.
2. Сложность ДНФ, минимальные ДНФ, кратчайшие ДНФ. Функции Шеннона для ДНФ.
3. Тупиковые ДНФ. Сокращенная ДНФ и методы ее построения.
4. ДНФ типа суммы тупиковых. Критерий вхождения конъюнкции в ДНФ типа суммы тупиковых.
5. Сокращенная ДНФ для монотонных функций.
6. Алгоритм построения всех тупиковых покрытий матрицы.
7. ДНФ суммы минимальных и алгоритмические трудности ее построения.
8. Градиентный алгоритм. Оценка сложности (величины) покрытия, получаемого градиентным алгоритмом.
9. Нижняя оценка функции Шеннона для СФЭ.
10. Метод Шеннона синтеза СФЭ в базисе $\{\&, \vee, \neg\}$.
11. Асимптотически наилучший метод О.Б. Лупанова синтеза СФЭ в базисе $\{\&, \vee, \neg\}$.

12. Эквивалентные преобразования формул в базисе $\{\&, \vee, \neg, x, 0, 1\}$.
13. Теорема перехода.
14. Теоремы Янова–Мучника.
15. Пример Линдона.
16. Проблема контроля управляющих систем. Тесты для таблиц. Тривиальные оценки.
17. Верхняя оценка длины диагностического теста для почти всех таблиц.
18. Алгоритм построения всех тупиковых тестов.
19. Проблема NP-полноты. Теорема Кука (формулировка).
20. NP-полнота языка КЛИКА.
21. NP-полнота языка 3-ВЫП.
22. Полиномиальность языка 2-ВЫП.
23. Задача о кратчайшем остовом дереве. Жадный алгоритм для нее.
24. NP-полнота языка ВП. Свойства жадного алгоритма для задачи МВП.
25. Сложность одновременного нахождения максимума и минимума в массиве.
26. Задача тестирования линейности булевой функции. Задача доказательства нелинейности булевых функций.

Библиографический список

1. Blake A. Canonical expressions in Boolean algebra. Dissertation. Dept. of Mathematics, Univ. of Chicago, 1937. Univ. of Chicago Libraries, 1938.
2. Журавлев Ю. И. Оценки сложности алгоритмов построения минимальных дизъюнктивных нормальных форм для функций алгебры логики // Дискретный анализ. 1964. № 3. С. 41–77.
3. Вороненко А. А. Об универсальных частичных функциях для класса линейных функций // Дискретная математика. 2012. № 3. С. 62–65.
4. Вороненко А. А., Вялый М. Н. Нижняя оценка мощности области определения универсальных функций для класса линейных булевых функций // Дискретная математика. 2016. № 4. С. 50–57.
5. Дискретная математика и математические вопросы кибернетики/ Под ред. С. В. Яблонского, О. Б. Лупанова. Т. 1. М.: Наука, 1974.

6. Cook S. A. The complexity of theorem-proving procedures. 3rd STOC. (PP. 151-158) // New York. – 1970.
7. Вороненко А. А. О функции Шеннона для длины сертификата повторности булевых функций в одном семействе базисов // Вестн. Моск. ун-та. Сер. 15. Вычисл. матем. и киберн. 2013. №4. С. 45–47 (Voronenko A. A. On the Shannon function for read-many certificate length in one base family // Moscow University Computational Mathematics and Cybernetics. 2013. **37**. N 1. P.26–27.)
8. Shannon C. E. A symbolic analysis of relay and switching circuits // Trans. AIEE. 1938. **57**. P. 713–723.
9. Shannon C. E. The syntesis of two-terminal switching circuits // Bell Syst. Techn. J. 1949. **28**. N1. P. 59–98.
(Шеннон К. Работы по теории информации и кибернетике. М.: ИЛ, 1963. С. 59–101.)
10. Алексеев В. Б. Лекции по дискретной математике. Учебное пособие. М.: Изд. отд. ф-та ВМК МГУ им. М. В. Ломоносова, 2004.
11. Поваров Г. Н. Метод синтеза вычислительных и управляющих контактных схем // Автоматика и телемеханика. 1957. **18**. № 2. С. 145–162.
12. Cardot C. Quelques resultats sur l’application de l’algebre de Boole a la synthese des circuits a relais // Annales des Telecommunications. 1952. **7**. N2. P. 75–84.

13. Янов Ю. И., Мучник А. А. О существовании k -значных замкнутых классов, не имеющих конечного базиса // ДАН СССР. 1959. **127**. № 1. С. 144–146.
14. Субботовская Б. А. О сравнении базисов при реализации функций алгебры логики формулами // ДАН СССР. 1963. **149**. № 4. С. 784–787.
15. Стеценко В. А. О предположих базисах в P_2 // Математические вопросы кибернетики. Вып. 4. М.: Физматлит, 1992. С. 139–177.
16. Вороненко А. А., Федорова В. С., Чистиков Д. В. Повторность булевых функций в элементарном базисе // Известия ВУЗов. Математика. 2011. № 11. С. 72–77.
17. Вороненко А. А. Новое доказательство теоремы Стеценко // Вестн. Моск. ун-та. Сер.15. Вычисл. матем. и киберн. 2014. № 2. С. 39–42.
18. M. Karnaugh, The Map Method for Synthesis of Combinational Logic Circuits // AIEE Committee on Technical Operations for Presentation at the AIEE Summer General Meeting. Atlantic City, N. J., 1953. P. 593–599.
19. Joseph B. Kruskal, Jr. Proceedings of the American Mathematical Society Vol. 7, No. 1 (Feb., 1956), pp. 48-50
20. R.C. Lyndon. Identities in finite algebras // Proc. Amer. Math. Soc. 1954. **5**. P. 8–9.

21. Chistikov D., Fedorova V., Voronenko A. Certificates of non-membership for classes of read-once functions // *Fundamenta Informaticae*. 2014. **132**. N 1. P. 63–77.
22. Karnaugh M. The map method for synthesis of combinational logic circuits // *Transactions of the American Institute of Electrical Engineers*. 1953. **72**. Part I. P. 593–598.
23. Quine van Orman W. The Problem of simplifying truth functions // *The American Mathematical Monthly*. 1952. **59**. N8. P. 521–531.
24. Гаврилов Г.П., Сапоженко А.А. Задачи и упражнения по дискретной математике. Учебное пособие. 3-е изд. М.: ФИЗМАЛИТ, 2004.