

Дополнительные главы дискретной математики и кибернетики

Семинар 1

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Множество слов

Определение

- **Алфавит** A — это непустое множество символов.
- A^* — это **множество слов** (конечной длины) в алфавите A , включая пустое слово Λ .
- **Длина** $|w|$ слова $w \in A^*$ — это количество символов в слове w . Длина пустого слова Λ есть нуль.

Определение

- **Конкатенация** слов $u = a_{i_1} \dots a_{i_k} \in A^*$ и $v = b_{j_1} \dots b_{j_l} \in A^*$ — это слово

$$u * v = uv = a_{i_1} \dots a_{i_k} b_{j_1} \dots b_{j_l} \in A^*.$$

При этом для любого $w \in A^*$ определяем $\Lambda w = w \Lambda = w$.

- **Возведение в степень**: $a^n = \underbrace{a * \dots * a}_{n \text{ раз}}$ при $n \in \mathbb{N}$; $a^0 = \Lambda$.

Множество слов

Определение

- Слово $u \in A^*$ является **префиксом** слова $w \in A^*$, если существует слово $v \in A^*$ такое, что $w = uv$.
- Префикс называется **собственным**, если он не равен Λ и не совпадает со всем словом w .

Определение

- Слово $u \in A^*$ является **суффиксом** слова $w \in A^*$, если существует слово $v \in A^*$ такое, что $w = vu$.
- Суффикс называется **собственным**, если он не равен Λ и не совпадает со всем словом w .

Определение

- Слово $u \in A^*$ является **подсловом** слова $w \in A^*$, если существуют слова $v_1, v_2 \in A^*$ такие, что $w = v_1uv_2$.

Конечные автоматы

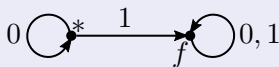
Определение

Конечный автомат (распознаватель) — это (A, Q, g, q_1, F) , где

- $A \neq \emptyset$ — входной алфавит (часто задан заранее и не является частью автомата),
- $Q \neq \emptyset$ — множество состояний,
- $g: A \times Q \rightarrow Q$ — функция переходов,
- $q_1 \in Q$ — начальное состояние,
- $F \subseteq Q$ — множество заключительных состояний.

Пример

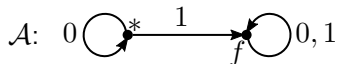
$A = \{0, 1\}$, диаграмма Мура автомата:



Конечные автоматы

Работа автомата

- Автомат работает в дискретном времени. В каждый момент t он находится в некотором состоянии $q(t)$ из множества Q . В начальный момент он находится в состоянии $q(0) = q_1$.
- В каждый момент работы он получает на вход символ $x(t)$ и переходит в следующее состояние: $q(t) = g(x(t), q(t-1))$.
- Когда входные символы закончились, работа автомата завершается. Если он оказался в состоянии из F , автомат принимает слово. Иначе он отвергает слово.
- Каждый автомат $\mathcal{A}$ выделяет множество $D(\mathcal{A}) \subseteq A^*$ слов, которые он принимает.
- Множества, для которых можно построить допускающий их автомат, называются конечно-автоматными.



$$L(\mathcal{A}) = \{w_1 1 w_2 \mid w_1, w_2 \in A^*\}$$

Построение конечного автомата

[1], 1 (1).

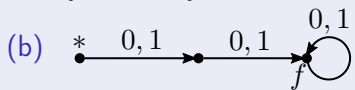
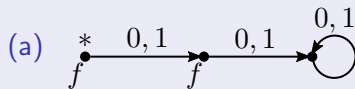
Построить диаграмму Мура для автомата в алфавите $\{1, 0\}$, который допускает следующее множество:

1a) Множество $\{0, 1, \Lambda\}$.

1b) Множество $\{0, 1\}^* \setminus \{0, 1, \Lambda\}$.

Построение конечного автомата

Решение [1], 1 (1).



Построение конечного автомата

[1], 1 (2).

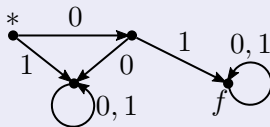
Построить диаграмму Мура для автомата в алфавите $\{1, 0\}$, который допускает следующее множество:

2. Слова, начинающиеся на 01.

Построение конечного автомата

Решение [1], 1 (2).

Автомат для слов, начинающихся на 01:



Построение конечного автомата

[1], 1 (4).

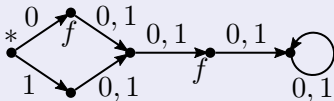
Построить диаграмму Мура для автомата в алфавите $\{1, 0\}$, который допускает следующее множество:

4. Слова длины 3 и слово 0.

Построение конечного автомата

Решение [1], 1 (4).

Автомат для слов длины 3 и слова 0:



Построение конечного автомата

[1], 1 (6).

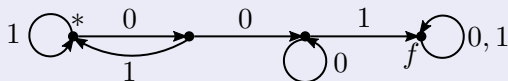
Построить диаграмму Мура для автомата в алфавите $\{1, 0\}$, который допускает следующее множество:

6. Слова, содержащие 001.

Построение конечного автомата

Решение [1], 1 (6).

Автомат для слов, содержащих 001:



Построение конечного автомата

[1], 1 (9).

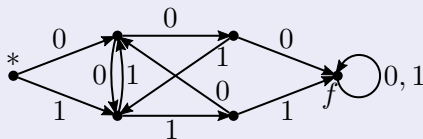
Построить диаграмму Мура для автомата в алфавите $\{1, 0\}$, который допускает следующее множество:

9. Слова, имеющие вхождения 000 и 111.

Построение конечного автомата

Решение [1], 1 (9).

Автомат для слов, имеющих вхождения 000 и 111:



Построение конечного автомата

[1], 2 (1).

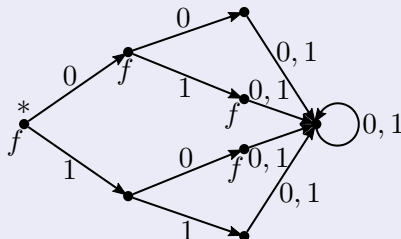
Доказать конечную автоматность множеств:

1. Конечное множество X в алфавите $\{a_1, \dots, a_m\}$; множество $\{a_1, \dots, a_m\}^* \setminus X$;

Построение конечного автомата

Решение [1], 2 (1).

- Конечное множество X в алфавите $\{a_1, \dots, a_m\}$. Пример автомата для $A = \{0, 1\}$, $X = \{\Lambda, 0, 01, 10\}$:



- Для $\{a_1, \dots, a_m\}^* \setminus X$ нужно инвертировать заключительные состояния.

Построение конечного автомата

[1], 2 (3).

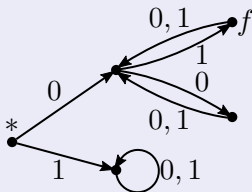
Доказать конечную автоматность множеств:

3. Множество слов в алфавите $\{0, 1\}$, которые имеют чётную длину, начинаются символом 0 и оканчиваются символом 1.

Построение конечного автомата

Решение [1], 2 (3).

Автомат для множества слов в алфавите $\{0, 1\}$, которые имеют чётную длину, начинаются символом 0 и оканчиваются символом 1:



Построение конечного автомата

[1], 2 (4).

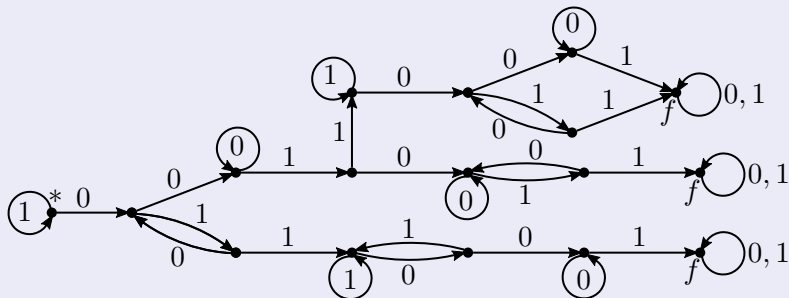
Доказать конечную автоматность множеств:

4. Множество слов в алфавите $\{0, 1\}$, содержащих неперекрывающиеся слова 001, 011.

Построение конечного автомата

Решение [1], 2 (4).

Автомат для множества слов в алфавите $\{0, 1\}$, содержащих неперекрывающиеся слова 001, 011:



- Верхняя ветвь соответствует нахождению слова 0011.
- Средняя ветвь соответствует нахождению слова 001.
- Нижняя ветвь соответствует нахождению слова 011.

Правоинвариантные отношения эквивалентности

Определение

Отношение $\sim \subseteq A^* \times A^*$ — **отношение эквивалентности**, если

- $\forall a \in A^* \quad a \sim a,$
- $\forall a, b \in A^* \quad a \sim b \equiv b \sim a,$
- $\forall a, b, c \in A^* \quad (a \sim b) \& (b \sim c) \rightarrow a \sim c.$

Определение

- Множество A^* разбивается отношением $\sim$ на **классы эквивалентности**: максимальные множества попарно эквивалентных элементов.
- **Индекс отношения эквивалентности** — это число классов эквивалентности.

Правоинвариантные отношения эквивалентности

Определение

- Отношение эквивалентности $\sim \subseteq A^* \times A^*$ является **правоинвариантным**, если

$$\forall a, b, c \in A^* \quad (a \sim b) \rightarrow (ac \sim bc).$$

- Иными словами,

$$\forall a, b \in A^*, c_i \in A \quad (a \sim b) \rightarrow (ac_i \sim bc_i).$$

Теорема

Множество $L \subseteq A^$ конечно-автоматно тогда и только тогда, когда оно является объединением некоторого числа классов эквивалентности правоинвариантного отношения эквивалентности конечного индекса.*

Построение правоинвариантного отношения

[1], 3 (1).

Построить правоинвариантное отношение эквивалентности конечного индекса, объединением классов эквивалентности которого являются множества:

1. $\{\Lambda\}$.

Построение правоинвариантного отношения

Решение [1], 3 (1).

Отношение для $L = \{\Lambda\}$.

Классы эквивалентности: $\{\Lambda\}, A^* \setminus \{\Lambda\}$.

Построение правоинвариантного отношения

[1], 3 (3).

Построить правоинвариантное отношение эквивалентности конечного индекса, объединением классов эквивалентности которого являются множества:

3. $\{\Lambda, 0, 1\}$.

Построение правоинвариантного отношения

Решение [1], 3 (3).

Отношение для $L = \{\Lambda, 0, 1\}$.

Классы эквивалентности: $\{\Lambda\}$, $\{0\}$, $\{1\}$, $A^* \setminus \{\Lambda, 0, 1\}$.

Построение правоинвариантного отношения

[1], 3 (5).

Построить правоинвариантное отношение эквивалентности конечного индекса, объединением классов эквивалентности которого являются множества:

5. $\{0^n 1 : n \geq 0\}.$

Построение правоинвариантного отношения

Решение [1], 3 (5).

Отношение для $L = \{0^n 1 : n \geq 0\}$.

Классы эквивалентности:

$\{0^n 1 : n \geq 0\}$, $\{0^n : n \geq 0\}$, $A^* \setminus (\{0^n 1 : n \geq 0\} \cup \{0^n : n \geq 0\})$.

Построение правоинвариантного отношения

[1], 4.

Для любого $n \geq 2$ определить на $\{0, 1\}$ правоинвариантное отношение эквивалентности индекса n .

Построение правоинвариантного отношения

Решение [1], 4.

Правоинвариантное отношение эквивалентности индекса n .

Классы эквивалентности:

$\{\Lambda\}, \{0\}, \dots, \{0^{n-2}\}, \{0, 1\}^* \setminus \{0^k \mid k = \overline{0, n-2}\}.$

Доказательство неавтоматности множеств

[1], 5 (1).

Пользуясь правоинвариантным отношением эквивалентности, доказать, что множества в алфавите $\{0, 1\}$ НЕ конечно-автоматны:

1. $\{0^n 1^{2n} \mid n \geq 1\}$.

Доказательство неавтоматности множеств

Решение [1], 5 (1).

- Пусть $\{0^n 1^{2n} \mid n \geq 1\}$ — это объединение классов эквивалентности правоинвариантного отношения $\sim$ конечного индекса.
- Выберем $i \neq j$ такие, что $0^i \sim 0^j$.
- Такие i, j существуют, так как иначе все слова 0^k , $k \in \mathbb{N}$ принадлежали бы разным классам, то есть классов было бы бесконечное число.
- Поскольку отношение правоинвариантное, из $0^i \sim 0^j$ следует $0^i 1^{2i} \sim 0^j 1^{2i}$.
- Поскольку весь класс эквивалентности, содержащий $0^i 1^{2i}$, лежит в $\{0^n 1^{2n} \mid n \geq 1\}$, получаем $0^j 1^{2i} \in \{0^n 1^{2n} \mid n \geq 1\}$.
- Противоречие означает, что $\{0^n 1^{2n} \mid n \geq 1\}$ не является объединением классов эквивалентности правоинвариантного отношения конечного индекса. Значит, это множество не конечно-автоматно.

Доказательство неавтоматности множеств

[1], 5 (3).

Пользуясь правоинвариантным отношением эквивалентности, доказать, что множества в алфавите $\{0, 1\}$ НЕ конечно-автоматны:

3. Множество симметричных слов в алфавите A , $|A| \geq 2$.

Доказательство неавтоматности множеств

Решение [1], 5 (3).

- Пусть множество S симметричных слов в алфавите $A = \{0, 1, a_3, \dots, a_n\}$ — это объединение классов эквивалентности правоинвариантного отношения $\sim$ конечного индекса.
- Выберем $i \neq j$ такие, что $0^i 1 \sim 0^j 1$.
- Такие i, j существуют, так как иначе все слова $0^k 1$, $k \in \mathbb{N}$ принадлежали бы разным классам, то есть классов было бы бесконечное число.
- Поскольку отношение правоинвариантное, из $0^i 1 \sim 0^j 1$ следует $0^i 10^i \sim 0^j 10^i$.
- Поскольку весь класс эквивалентности, содержащий $0^i 10^i$, лежит в S , получаем $0^j 10^i \in S$.
- Противоречие означает, что S не является объединением классов эквивалентности правоинвариантного отношения конечного индекса. Значит, это множество не конечно-автоматно.

Домашнее задание

[1], 1 (3, 5, 7, 10).

Построить диаграмму Мура для автомата в алфавите $\{1, 0\}$, который допускает следующее множество:

3. Слова, которые оканчиваются словом 01.
5. Слова длины 3, кроме 110.
7. Слова, которые составлены из «блоков» 011 и 101.
10. Слова, у которых за каждым символом 1 следуют как минимум 2 символа 0.

Домашнее задание

[1], 2 (2, 5).

Построить диаграмму Мура для автомата в алфавите $\{1, 0\}$, который допускает следующее множество:

2. Множество слов вида a_i^n , ($i \in \{1, \dots, m\}$, $n \in \mathbb{N}_0$) в алфавите $\{a_1, \dots, a_m\}$.
5. Множества вида $0^{n_1} 1 0^{n_2} 1 \dots 1 0^{n_k}$, $n_i \geq 1$:
 - a) При каждом фиксированном k .
 - b) При произвольном k .

Домашнее задание

[1], 3 (4, 6).

Построить правоинвариантное отношение эквивалентности конечного индекса, объединением классов эквивалентности которого являются множества:

4. Множество слов вида 0^{3n} , где $n \geq 1$.
6. Слова чётной длины и 1, 111.

Домашнее задание

[1], 5 (2, 4).

Пользуясь правоинвариантным отношением эквивалентности доказать, что множества в алфавите $\{0, 1\}$ НЕ конечно-автоматны:

2. $\{0^n 1 0^n \mid n \geq 1\}$.

4. $\{0^{n^2} \mid n \geq 1\}$.

Список литературы

1. Марченков С. С. Задачи по конечным автоматам-распознавателям (прилагается к плану семинарских занятий по курсу «Дополнительные главы дискретной математики»).

Дополнительные главы дискретной математики
и кибернетики
Семинар 2

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Операции над множествами

Операции над множествами

- Пусть $X, Y \subseteq A^*$.

- Дополнение X :

$$A^* \setminus X.$$

- Объединение X, Y :

$$X \cup Y.$$

- Пересечение X, Y :

$$X \cap Y.$$

- Произведение X, Y :

$$X \cdot Y = \{xy \mid x \in X, y \in Y\}, \quad X \cdot \emptyset = \emptyset \cdot X = \emptyset.$$

- Итерация X :

$$X^* = \{x_1 \dots x_n \mid x_i \in X, i = \overline{1, n}, n \in \mathbb{N}_0\}, \quad \emptyset^* = \emptyset.$$

Прямое произведение автоматов

[1], 8.

Ввести операцию прямого произведения автоматов. Доказать замкнутость конечно-автоматных множеств относительно операций объединения и пересечения.

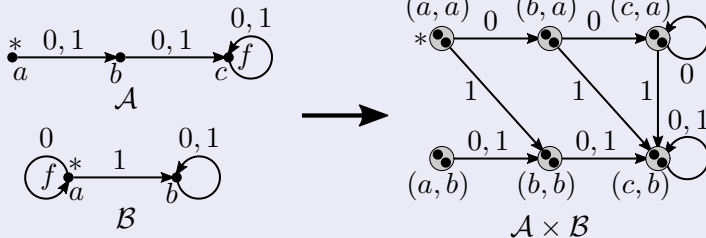
Прямое произведение автоматов

Решение [1], 8.

- Пусть $\mathcal{A}_1 = (A, Q_1, f_1, q_1^1, F_1)$, $\mathcal{A}_2 = (A, Q_2, f_2, q_2^1, F_2)$.
- $\mathcal{A}_1 \times \mathcal{A}_2 = (A, Q_1 \times Q_2, f, (q_1^1, q_2^1), F)$, где

$$f(a, (q^1, q^2)) = (f_1(a, q^1), f_2(a, q^2)), \quad q^1 \in Q_1, \quad q^2 \in Q_2, \quad a \in A.$$

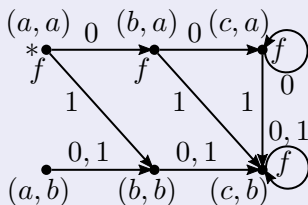
- F выбирается произвольно.



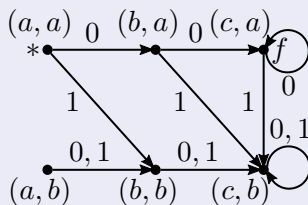
Прямое произведение автоматов

Решение [1], 8 (продолжение).

- Для распознавания объединения множеств $D(\mathcal{A}_1), D(\mathcal{A}_2)$ в F включаются состояния, первые координаты которых входят в F_1 , а также состояния, вторые координаты которых входят в F_2 .
- Для пересечения множеств $D(\mathcal{A}_1), D(\mathcal{A}_2)$ в F включаются состояния, первые координаты которых входят в F_1 , а вторые в F_2 .



Объединение



Пересечение

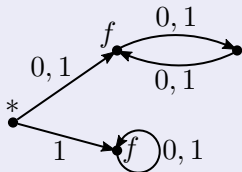
Недетерминированные автоматы

Определение

Недетерминированный конечный автомат — это (A, Q, f, q_1, F) , где

- $A \neq \emptyset$ — входной алфавит,
- $Q \neq \emptyset$ — множество состояний,
- $f: A \times Q \rightarrow 2^Q \setminus \{\emptyset\}$ — функция переходов (по символу и состоянию выбирается подмножество состояний),
- $q_1 \in Q$ — начальное состояние,
- $F \subseteq Q$ — множество заключительных состояний.

Пример



На диаграмме Мура из одного состояния может исходить несколько стрелок с одним и тем же символом.

Недетерминированные автоматы

Работа недетерминированного автомата

- Автомат начинает работу в начальном состоянии и на каждом такте переходит по стрелке с текущим символом.
- Если стрелок с текущим символом из текущего состояния несколько, выбирается любая.
- На одном входе автомат может проводить вычисления разными способами.
- Автомат принимает слово, если он попадает в финальное состояния при хотя бы одном из возможных вычислений.

Недетерминированные автоматы

[1], 10 (1).

Построить диаграммы Мура и описать множества, допускаемые недетерминированными автоматами:

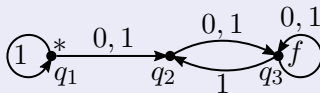
1. Автомат задан соотношениями:

$$\begin{aligned}Q &= \{q_1, q_2, q_3\}, \\f(0, q_1) &= \{q_2\}, \quad f(1, q_1) = \{q_1, q_2\}, \\f(0, q_2) &= \{q_3\}, \quad f(1, q_2) = \{q_3\}, \\f(0, q_3) &= \{q_3\}, \quad f(1, q_3) = \{q_2, q_3\}, \\F &= \{q_3\}.\end{aligned}$$

Недетерминированные автоматы

Решение [1], 10 (1).

- Диаграмма Мура:



- Автомат распознаёт множество слов длины не менее 2.

Недетерминированные автоматы

Процедура детерминизации

- Пусть $\mathcal{A} = (A, Q, f, q_1, F)$ — недетерминированный конечный автомат.
- Строим автомат $\mathcal{B} = (A, 2^Q \setminus \{\emptyset\}, f', \{q_1\}, F')$, где

$$f'(a, X) = \bigcup_{q \in X} f(a, q), \quad F' = \{X \in Q: X \cap F \neq \emptyset\}.$$

- Автомат $\mathcal{B}$ детерминированный и $D(\mathcal{A}) = D(\mathcal{B})$.
- На практике имеет смысл начать строить диаграмму Мура с $\{q_1\}$ и добавлять только достижимые состояния.

Недетерминированные автоматы

[1], 11 (автомат из задачи 10 (1)).

Построить методом детерминизации эквивалентный детерминированный автомат:

$$Q = \{q_1, q_2, q_3\},$$

$$f(0, q_1) = \{q_2\}, \quad f(1, q_1) = \{q_1, q_2\},$$

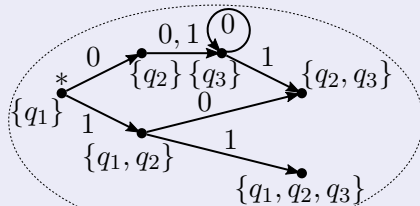
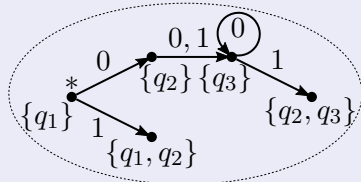
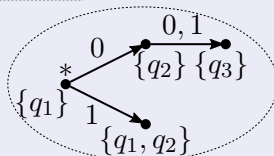
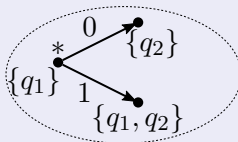
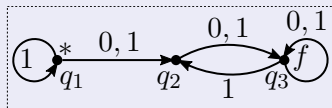
$$f(0, q_2) = \{q_3\}, \quad f(1, q_2) = \{q_3\},$$

$$f(0, q_3) = \{q_3\}, \quad f(1, q_3) = \{q_2, q_3\},$$

$$F = \{q_3\}.$$

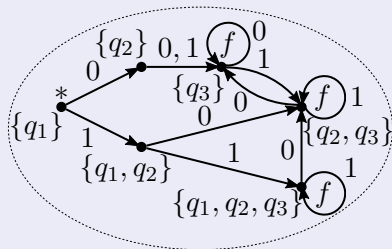
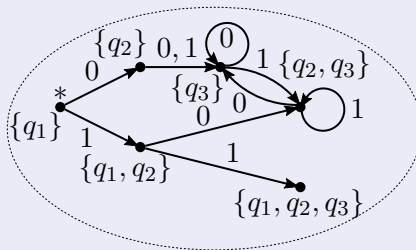
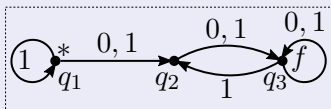
Недетерминированные автоматы

Решение [1], 11 (автомат из задачи 10 (1)).



Недетерминированные автоматы

Решение [1], 11 (автомат из задачи 10 (1)) (продолжение).



Недетерминированные автоматы

[1], 12 (1).

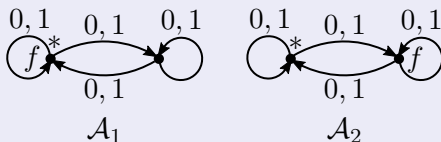
Пусть $\mathcal{A}_i = \{A, Q, f, q_1, F_i\}, i = 1, 2$ (недетерминированные). Верно ли, что

1. Если $F_2 = Q \setminus F_1$, то $D(\mathcal{A}_2) = A^* \setminus D(\mathcal{A}_1)$.

Недетерминированные автоматы

Решение [1], 12 (1).

- Неверно. Рассмотрим следующие автоматы:



- $D(\mathcal{A}_1) = A^*$, $D(\mathcal{A}_2) = A^* \setminus \{\Lambda\}$.

Недетерминированные автоматы

[1], 12 (3).

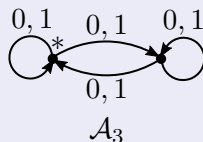
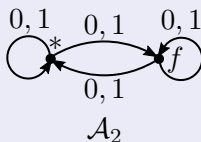
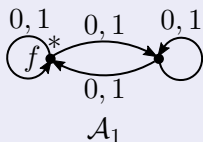
Пусть $\mathcal{A}_i = \{A, Q, f, q_1, F_i\}, i = 1, 2, 3$ (недетерминированные). Верно ли, что

3. Если $F_3 = F_1 \cap F_2$, то $D(\mathcal{A}_3) = D(\mathcal{A}_1) \cap D(\mathcal{A}_2)$.

Недетерминированные автоматы

Решение [1], 12 (3).

- Неверно. Рассмотрим следующие автоматы:



- $D(\mathcal{A}_1) = A^*$, $D(\mathcal{A}_2) = A^* \setminus \{\Lambda\}$, $D(\mathcal{A}_3) = \emptyset$.

Операции над автоматами

Построение автомата для произведения множеств

- Пусть $\mathcal{A}_1 = \{A, Q_1, f_1, q'_s, F_1\}$, $\mathcal{A}_2 = \{A, Q_2, f_2, q''_s, F_2\}$ — недетерминированные автоматы, $Q_1 \cap Q_2 = \emptyset$.
- Строим недетерминированный автомат $\mathcal{A} = \{A, Q, f, q_s, F\}$ такой, что $D(\mathcal{A}) = D(\mathcal{A}_1) \cdot D(\mathcal{A}_2)$.
- Для этого выбираем:

$$Q = Q_1 \cup Q_2,$$

$$f(a, q) = \begin{cases} f_1(a, q), & q \in Q_1 \setminus F_1, \\ f_1(a, q) \cup f_2(a, q''_s), & q \in F_1, \\ f_2(a, q), & q \in Q_2, \end{cases}$$

$$q_s = q'_s,$$

$$F = \begin{cases} F_2, & q''_s \notin F_2, \\ F_1 \cup F_2, & q''_s \in F_2. \end{cases}$$

Операции над автоматами

[1], 23 (1).

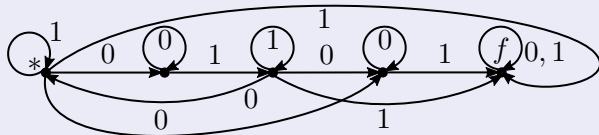
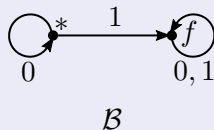
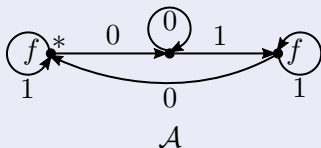
Построить недетерминированный автомат, который допускает произведение автоматных множеств $D(\mathcal{A}) \cdot D(\mathcal{B})$:

$$\begin{aligned}\mathcal{A}: Q &= \{q_1, q_2, q_3\}, \\ f(0, q_1) &= q_2, \quad f(1, q_1) = q_1, \\ f(0, q_2) &= q_2, \quad f(1, q_2) = q_3, \\ f(0, q_3) &= q_1, \quad f(1, q_3) = q_3, \\ F &= \{q_1, q_3\}.\end{aligned}$$

$$\begin{aligned}\mathcal{B}: Q &= \{q_1, q_2\}, \\ f(0, q_1) &= q_1, \quad f(1, q_1) = q_2, \\ f(0, q_2) &= q_2, \quad f(1, q_2) = q_2, \\ F &= \{q_2\}.\end{aligned}$$

Операции над автоматами

Решение [1], 23 (1).



Операции над автоматами

Построение автомата для итерации множества

- Пусть $\mathcal{A}' = \{A, Q', f', q'_s, F'\}$ — недетерминированный автомат, $q_s \notin Q'$.
- Строим недетерминированный автомат $\mathcal{A} = \{A, Q, f, q_s, F\}$ такой, что $D(\mathcal{A}) = (D(\mathcal{A}'))^*$.
- Для этого выбираем:

$$Q = Q' \cup \{q_s\},$$

$$f(a, q) = \begin{cases} f'(a, q), & q \in Q' \setminus F', \\ f'(a, q) \cup f'(a, q'_s), & q \in F', \\ f'(a, q'_s), & q = q_s, \end{cases}$$

$$F = F' \cup \{q_s\}.$$

- Если $q'_s \in F'$, то состояние q_s можно не добавлять, а оставить q'_s начальным.

Операции над автоматами

[1], 24 (1).

Для автомата $\mathcal{A}$ из 23 (1) построить недетерминированный автомат, принимающий $D(\mathcal{A})^*$.

$$\mathcal{A}: Q = \{q_1, q_2, q_3\},$$

$$f(0, q_1) = q_2, \quad f(1, q_1) = q_1,$$

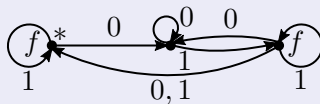
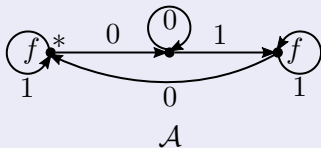
$$f(0, q_2) = q_2, \quad f(1, q_2) = q_3,$$

$$f(0, q_3) = q_1, \quad f(1, q_3) = q_3,$$

$$F = \{q_1, q_3\}.$$

Операции над автоматами

Решение [1], 24 (1).



Операции над множествами

[1], 13.

Построить из множеств $\{0\}$, $\{1\}$ при помощи объединения, произведения и итерации множество всех слов, содержащих подслово 0001.

Операции над множествами

Решение [1], 13.

$$(\{0\} \cup \{1\})^* \cdot \{0\} \cdot \{0\} \cdot \{0\} \cdot \{1\} \cdot (\{0\} \cup \{1\})^*$$

Операции над множествами

[1], 14.

$A = \{a_1, \dots, a_m\}$. Сколько раз нужно применить операцию итерации, чтобы получить множество $A^* \setminus \{\bar{a}\}$ при помощи объединения, произведения и итерации из множеств вида $\{a_i\}$?

Операции над множествами

Решение [1], 14.

- Пусть X — множество слов длины не больше, чем $|\bar{a}|$, исключая само слово $\bar{a}$.
- Пусть Y — множество слов длины $|\bar{a}| + 1$.
- Поскольку множества X и Y конечны, их можно построить без применения итерации.
- Тогда достаточно одного применения итерации:

$$A^* \setminus \{\bar{a}\} = X \cup Y \cdot (\{a_1\} \cup \dots \cup \{a_n\})^*.$$

- Без итерации обойтись невозможно, так как множество $A^* \setminus \{\bar{a}\}$ бесконечно.

Домашнее задание

[1], 9.

Сохраняют ли операции $\cup$, $\cap$, $\cdot$, $*$ класс не конечно-автоматных множеств?

[1], 10 (2).

Построить диаграммы Мура и описать множества, допускаемые недетерминированными автоматами:

$$Q = \{q_1, q_2, q_3\},$$

$$f(0, q_1) = \{q_1, q_2\}, \quad f(1, q_1) = \{q_1\},$$

$$f(0, q_2) = \{q_2\}, \quad f(1, q_2) = \{q_2, q_3\},$$

$$f(0, q_3) = \{q_1, q_3\}, \quad f(1, q_3) = \{q_1, q_3\},$$

$$F = \{q_3\}.$$

Домашнее задание

[1], 11 (автомат из задачи 10 (2)).

Построить методом детерминизации эквивалентный детерминированный автомат:

$$Q = \{q_1, q_2, q_3\},$$

$$f(0, q_1) = \{q_1, q_2\}, \quad f(1, q_1) = \{q_1\},$$

$$f(0, q_2) = \{q_2\}, \quad f(1, q_2) = \{q_2, q_3\},$$

$$f(0, q_3) = \{q_1, q_3\}, \quad f(1, q_3) = \{q_1, q_3\},$$

$$F = \{q_3\}.$$

[1], 12 (2).

Пусть $\mathcal{A}_i = \{A, Q, f, q_1, F_i\}, i = 1, 2, 3$ (недетерминированные). Верно ли, что

2. Если $F_3 = F_1 \cup F_2$, то $D(\mathcal{A}_3) = D(\mathcal{A}_1) \cup D(\mathcal{A}_2)$.

Домашнее задание

[1], 23 (2).

Построить недетерминированный автомат, который допускает произведение автоматных множеств $D(\mathcal{A}) \cdot D(\mathcal{B})$:

$$\begin{aligned}\mathcal{A}: Q &= \{q_1, q_2\}, \\ f(0, q_1) &= q_1, \quad f(1, q_1) = q_2, \\ f(0, q_2) &= q_1, \quad f(1, q_2) = q_2, \\ F &= \{q_2\}.\end{aligned}$$

$$\begin{aligned}\mathcal{B}: Q &= \{q_1, q_2, q_3\}, \\ f(0, q_1) &= q_2, \quad f(1, q_1) = q_3, \\ f(0, q_2) &= q_2, \quad f(1, q_2) = q_2, \\ f(0, q_3) &= q_3, \quad f(1, q_3) = q_1, \\ F &= \{q_2, q_3\}.\end{aligned}$$

Домашнее задание

[1], 24 (2).

Для автомата $\mathcal{B}$ из 23 (2) построить недетерминированный автомат, принимающий $D(\mathcal{A})^*$.

$$\mathcal{B}: Q = \{q_1, q_2, q_3\},$$

$$f(0, q_1) = q_2, \quad f(1, q_1) = q_3,$$

$$f(0, q_2) = q_2, \quad f(1, q_2) = q_2,$$

$$f(0, q_3) = q_3, \quad f(1, q_3) = q_1,$$

$$F = \{q_2, q_3\}.$$

[1], 15.

Пусть множество X состоит из n слов. Может ли множество $X \cdot X$ содержать больше, меньше или в точности n^2 слов?

Список литературы

1. Марченков С. С. Задачи по конечным автоматам-распознавателям (прилагается к плану семинарских занятий по курсу «Дополнительные главы дискретной математики»).

Дополнительные главы дискретной математики и кибернетики

Семинар 3

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Регулярные выражения

Определение

Пусть $A = \{a_1, \dots, a_m\}$ — конечный алфавит.

- $\emptyset, \{\Lambda\}, \{a_i\}, i = \overline{1, m}$ — **регулярные множества**, обозначаемые **регулярными выражениями** $\emptyset, \Lambda, a_i, i = \overline{1, m}$ соответственно.
- Если X, Y — регулярные множества, обозначаемые регулярными выражениями α, β , то $X \cup Y, X \cdot Y, X^*$ — **регулярные множества**, обозначаемые **регулярными выражениями** $(\alpha \cup \beta), (\alpha \cdot \beta), (\alpha)^*$ соответственно.
- Скобки можно опускать с учётом приоритета операций: $*$, $\cdot$, $\cup$ (перечислены в порядке убывания приоритета).
- Знак $\cdot$ можно опускать.

Регулярные выражения

- Регулярное выражение является «шаблоном», показывающим устройство слов в множестве.
- Например, выражение $1^*(010 \cup 0110)1^*$ задаёт слова, в которых сначала присутствует некоторое (возможно, нулевое) количество единиц, далее следует подслово 010 или 0110, после чего снова следует некоторое количество единиц.

Теорема (Клини)

Множество является регулярным тогда и только тогда, когда оно является конечно-автоматным.

Регулярные выражения

Практическое использование

- Во многих текстовых редакторах и файловых менеджерах есть опция поиска/фильтра по регулярным выражениям.
- Эти регулярные выражения основаны на регулярных выражениях Клини, но в них добавлены дополнительные операции для сокращения записи.
- Например, `<[<>]*>` ищет пару угловых скобок с произвольным текстом (не содержащим других угловых скобок) внутри.
- Существует стандартный язык регулярных выражений, который несложно изучить. Он описан, например, в документации языка Python [2] или на Википедии [3].
- Обработчики регулярных выражений иногда поддерживают возможности, выходящие за рамки возможностей регулярных выражений Клини. Но наиболее эффективно реализуемые возможности используют регулярные выражения Клини.

Регулярные выражения

Реализация в программах

- По любому регулярному выражению можно построить конечный автомат, который распознаёт слова, соответствующие данному регулярному выражению
- Конечный автомат работает очень быстро: он проходит по символам текста только один раз, и для каждого символа совершает простую операцию изменения состояния.
- Память автомата конечна: она зависит только от регулярного выражения, но не от текста, по которому идёт поиск. Поэтому автомат может работать с очень большими текстами.
- Чтобы обеспечить быстрый поиск по регулярному выражению программа может построить автомат для этого выражения и применить его к тексту.

Регулярные выражения

[1], 16 (1).

Доказать регулярность множеств слов в алфавите $\{0, 1\}$:

1. Любое конечное множество слов и дополнение к конечному множеству.

Регулярные выражения

Решение [1], 16 (1).

- Конечное множество можно получить с помощью произведения и объединения:

$$\{a_{i_1}^1 \dots a_{i_{r_1}}^1, \dots, a_{i_1}^k \dots a_{i_{r_k}}^k\} = a_{i_1}^1 \cdot \dots \cdot a_{i_{r_1}}^1 \cup \dots \cup a_{i_1}^k \cdot \dots \cdot a_{i_{r_k}}^k.$$

- Построим дополнение к конечному множеству $\{\bar{a}_1, \dots, \bar{a}_k\}$.
- Пусть X — множество слов длины не больше, чем $\max_{i=1, \dots, k} (|\bar{a}_i|)$, исключая сами слова $\bar{a}_1, \dots, \bar{a}_k$.
- Пусть Y — множество слов длины $\max_{i=1, \dots, k} (|\bar{a}_i|) + 1$.
- Множества X и Y конечны, строим их по рассмотренному ранее принципу.
- Тогда

$$A^* \setminus \{\bar{a}_1, \dots, \bar{a}_k\} = X \cup Y \cdot (0 \cup 1)^*.$$

Регулярные выражения

[1], 16 (3).

Доказать регулярность множеств слов в алфавите $\{0, 1\}$:

3. Множество слов, содержащих в качестве под слова одно из слов $x_1, \dots, x_n$.

Регулярные выражения

Решение [1], 16 (3).

- Множество слов, содержащих в качестве подслова одно из слов $x_1, \dots, x_n$.
- Регулярное выражение: $(0 \cup 1)^*(x_1 \cup \dots \cup x_n)(0 \cup 1)^*$.

Регулярные выражения

[1], 16 (5).

Доказать регулярность множеств слов в алфавите $\{0, 1\}$:

5. Множество слов, не содержащих 01.

Регулярные выражения

Решение [1], 16 (5).

- Множество слов, не содержащих 01.
- Регулярное выражение: 1^*0^* .

Регулярные выражения

[1], 17 (1).

Привести пример бесконечного регулярного множества, которое невозможно получить

1. однократным использованием итерации.

Регулярные выражения

Решение [1], 17 (1).

- Рассмотрим множество, задающееся выражением 0^*1^* .
- Пусть 0^*1^* получается в виде выражения Φ над $\cup, \cdot, *, 0, 1, \Lambda, \emptyset$.
- Считаем, что в выражении не встречается символ $\emptyset$ (иначе её можно упростить по правилам $\emptyset \cdot X = \emptyset$, $\emptyset \cup X = X$, $\emptyset^* = \emptyset$).
- В выражении есть итерация, так как множество 0^*1^* бесконечно. Пусть итерация единственная и применяется к множеству X .
- Если в словах X есть и нули, и единицы, то в X^* есть слова со сколь угодно большим числом подслов 01 и 0^*1^* получить нельзя.
- Если в X слова состоят только из нулей, то количество единиц в словах множества 0^*1^* ограничено константой, что неверно.
- Если в X слова состоят только из единиц, то количество нулей в словах множества 0^*1^* ограничено константой, что неверно.
- Значит, итерация не может быть единственной.

Регулярные выражения

[1], 18.

Пусть X — регулярное множество в алфавите $\{a_1, \dots, a_m\}$;
 $Y_1, \dots, Y_m$ — регулярные множества в алфавите $\{b_1, \dots, b_n\}$. Доказать,
что множество $S_{Y_1 \dots Y_m}^{a_1 \dots a_m} X$ слов, получаемых одновременной заменой
букв $a_1, \dots, a_m$ в любом слове из X любыми словами из $Y_1, \dots, Y_m$
соответственно является регулярным.

Регулярные выражения

Решение [1], 18.

- Чтобы получить регулярное выражение множества $S_{Y_1 \dots Y_m}^{a_1 \dots a_m} X$ достаточно подставить регулярные выражения множеств $Y_1, \dots, Y_n$ вместо символов $a_1, \dots, a_m$ в регулярное выражение множества X .

Регулярные выражения

[1], 20.

Пусть X — регулярное множество, $\text{Rev}(X)$ — множество обращений слов из X . Доказать, что оно тоже будет регулярным.

Регулярные выражения

Решение [1], 20.

- Индукция по построению регулярного выражения.
- $\{0\}, \{1\}, \{\Lambda\}, \emptyset$ регулярны, поэтому

$$\text{Rev}(\{0\}) = \{0\}, \text{Rev}(\{1\}) = \{1\}, \text{Rev}(\{\Lambda\}) = \{\Lambda\}, \text{Rev}(\emptyset) = \emptyset.$$

регулярны.

- Пусть X, Y регулярны и уже доказано, что $\text{Rev}(X), \text{Rev}(Y)$ регулярны.
- Если $Z = X \cup Y$, то $\text{Rev}(Z) = \text{Rev}(X) \cup \text{Rev}(Y)$ регулярно.
- Если $Z = X \cdot Y$, то $\text{Rev}(Z) = \text{Rev}(Y) \cdot \text{Rev}(X)$ регулярно.
- Если $Z = X^*$, то $\text{Rev}(Z) = (\text{Rev}(X))^*$ регулярно.

Регулярные выражения

[1], 22.

Пусть X — конечно-автоматное множество в алфавите A , Y — конечно-автоматное множество в однобуквенном алфавите. Обозначим через X/Y множество тех слов из X , длины которых являются длинами слов из Y . Доказать, что множество X/Y конечно-автоматно.

Регулярные выражения

Решение [1], 22.

- Пусть $A = \{a_1, \dots, a_m\}$, а символ алфавита $Y — b$.
- Пусть Y_1 — множество слов в алфавите A , длины которых совпадают с длинами слов из Y . Тогда $Y_1 = S_{\{a_1, \dots, a_m\}}^b Y$ и по номеру 18 Y_1 регулярно.
- Тогда $X/Y = X \cap Y_1$. Поскольку пересечение конечно-автоматных множеств конечно-автоматно, пересечение регулярных множеств регулярно.

Домашнее задание

[1], 16 (2, 4, 6).

Доказать регулярность множеств слов в алфавите $\{0, 1\}$:

2. Множество слов, представимых в виде произведения заданных слов $\bar{a}_1, \dots, \bar{a}_n$.
4. Множество всех слов, длины которых имеют вид $5k + 1$ или $5k + 3$.
6. Множество всех, которые не содержат ни одно из заданных слов $\bar{a}_1, \dots, \bar{a}_n$ (использовать теорему Клини).

[1], 17 (2).

Привести пример бесконечного регулярного множества, которое невозможно получить

2. с n -кратным использованием операции $*$, $n \geq 2$.

Домашнее задание

[1], 19.

Пусть X — множество в алфавите A , B — непустое подмножество A . Обозначим через $\text{Proj}_B(X)$ множество всех слов, которые можно получить из слов множества X вычёркиванием букв множества $A \setminus B$ (если слово не содержит букв из B , то в результате вычёркивания образуется пустое слово). Доказать, что для регулярного множества X множество $\text{Proj}_B(X)$ также регулярно.

[1], 21.

Пусть X, Y — множества слов в алфавите A . Обозначим через $X \times Y$ множество слов в алфавите $A \times A$, которые имеют вид $(a_{i_1} a_{j_1})(a_{i_2} a_{j_2}) \dots (a_{i_n} a_{j_n})$, где $a_{i_1} \dots a_{i_n} \in X$ и $a_{j_1} \dots a_{j_n} \in Y$. С использованием теоремы Клини доказать, что для любых регулярных множеств X, Y множество $X \times Y$ тоже регулярно.

Список литературы

1. Марченков С. С. Задачи по конечным автоматам-распознавателям (прилагается к плану семинарских занятий по курсу «Дополнительные главы дискретной математики»).
2. <https://docs.python.org/3/library/re.html>
3. https://ru.wikipedia.org/wiki/Регулярные_выражения

Дополнительные главы дискретной математики и кибернетики

Семинар 4

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Детерминированные функции

Бесконечные последовательности

Пусть A — непустое множество.

- A^∞ — это множество счётно-бесконечных последовательностей вида $a_{i_1} a_{i_2} \dots$, где $a_{i_n} \in A$ при $n \in \mathbb{N}$.
- Пусть $a = a_{i_1} a_{i_2} \dots \in A^\infty$. Обозначим $a(t) = a_{i_t}$ при $t \in \mathbb{N}$.
- Для введения индексации с нуля пишем $a = a(0)a(1)\dots \in A^\infty$.
- **Конкатенация** слова $u = u_1 \dots u_k \in A^*$ и последовательности $a = a(1)a(2)\dots \in A^\infty$ — это последовательность

$$u * a = ua = u_1 \dots u_k a(1)a(2)\dots \in A^\infty.$$

При этом для любого $a \in A^\infty$ определяем $\Lambda a = a$.

- **Бесконечное повторение** слова $u = u_1 \dots u_k \in A^*$, $u \neq \Lambda$ есть

$$u^\omega = u_1 \dots u_k u_1 \dots u_k \dots \in A^\infty.$$

Детерминированные функции

Определение

Функция $y = f(x): A^\infty \rightarrow B^\infty$ является **детерминированной**, если для каждого $t \geq 1$ существует такая функция $\varphi_t(x_1, \dots, x_t): A^t \rightarrow B$, что

$$y(t) = \varphi_t(x(1), \dots, x(t)).$$

- Можно считать, что функция над бесконечными словами действует не сразу, а растянуто во времени: в каждый момент $t \geq 1$ функция получает на вход символ $x(t)$ и выдаёт символ $y(t)$.
- Тогда детерминированная функция «не может заглядывать в будущее»: её выход в момент t зависит только от входов $x(1), \dots, x(t)$, которые были получены ранее, и не зависит от будущих входов $x(t+1), x(t+2), \dots$

Некоторые автоматные функции

Напоминание о булевых функциях

- $x \oplus y = \overline{x}y \vee x\overline{y}$ — сложение по модулю 2 (исключающее «или»).
- $x \sim y = \overline{x \oplus y} = \overline{x}\overline{y} \vee xy =$ « x истинно **тогда и только тогда, когда** y истинно» — эквивалентность.
- $x \rightarrow y = \overline{x} \vee y =$ «**если** x истинно, **то** y истинно» — импликация.
- $x \mid y = \overline{xy}$ — штрих Шеффера.
- $x \downarrow y = \overline{x \vee y}$ — стрелка Пирса.
- Мнемоническое правило для выражения функций $x \mid y$ и $x \downarrow y$: поворачиваем черту на 90° и кладём сверху в виде отрицания. Внизу остаётся либо часть $\vee$ от стрелки (дизъюнкция), либо ничего (конъюнкция).

Детерминированные функции

[1], глава IV, 1.1 (1), с. 111.

Выяснить, является ли функция $y = f(x)$ детерминированной.

1. $y(1) = x(1), y(t) = x(1) \oplus \dots \oplus x(t), t \geq 2.$

Детерминированные функции

Решение [1], глава IV, 1.1 (1), с. 111.

$$y(1) = x(1), \quad y(t) = x(1) \oplus \dots \oplus x(t), \quad t \geq 2$$

- $y(t) = x(1) \oplus \dots \oplus x(t), \quad t \geq 1$
- Функция детерминированная.

Детерминированные функции

[1], глава IV, 1.1 (3), с. 111.

Выяснить, является ли функция $y = f(x)$ детерминированной.

3. $y(t) = x(1) \cdot \dots \cdot x(t) \cdot x(t+2) \rightarrow x(1), t \geq 1.$

Детерминированные функции

Решение [1], глава IV, 1.1 (3), с. 111.

$$y(t) = x(1) \cdot \dots \cdot x(t) \cdot x(t+2) \rightarrow x(1), t \geq 1$$

- $y(t) = \overline{x(1) \cdot \dots \cdot x(t) \cdot x(t+2)} \vee x(1) = \bar{x}(1) \vee \dots \vee \bar{x}(t+2) \vee x(1) = 1.$
- Функция детерминированная.

Детерминированные функции

[1], глава IV, 1.1 (8), с. 111.

Выяснить, является ли функция $y = f(x)$ детерминированной.

8. $y(1) = 1$, $y(t) = x(2 + x(t))$, $t \geq 2$.

Детерминированные функции

Решение [1], глава IV, 1.1 (8), с. 111.

$$y(1) = 1, \quad y(t) = x(2 + x(t)), \quad t \geq 2$$

- $y(t) = \bar{x}(t)x(2) \vee x(t)x(3), \quad y(2) = x(2)x(3).$
- Поскольку $y(2)$ существенно зависит от $x(3)$, функция недетерминированная.

Детерминированные функции

[1], глава IV, 1.1 (9), с. 111.

Выяснить, является ли функция $y = f(x)$ детерминированной.

9. $y(1) = y(2) = 0$, $y(t) = x(2 + x(t))$, $t \geq 3$.

Детерминированные функции

Решение [1], глава IV, 1.1 (9), с. 111.

$$y(1) = y(2) = 0, \quad y(t) = x(2 + x(t)), \quad t \geq 3$$

- $y(1) = 0, \quad y(2) = 0.$
- При $t \geq 3$ верно $y(t) = \bar{x}(t)x(2) \vee x(t)x(3)$. Зависит только от $x(1), \dots, x(t)$.
- Функция детерминированная.

Детерминированные функции

[1], глава IV, 1.2 (1), с. 111.

Выяснить, является ли функция $y = f(x)$ детерминированной.

$$1. f(\tilde{x}^\omega) = \begin{cases} \tilde{0}^\omega, & \tilde{x}^\omega = \tilde{0}^\omega \\ \tilde{1}^\omega, & \text{иначе.} \end{cases}$$

Детерминированные функции

Решение [1], глава IV, 1.2 (1), с. 111.

$$f(\tilde{x}^\omega) = \begin{cases} \tilde{0}^\omega, & \tilde{x}^\omega = \tilde{0}^\omega \\ \tilde{1}^\omega, & \text{иначе.} \end{cases}$$

- Пусть $y = f(\tilde{x}^\omega)$. Тогда значение $y(1)$ зависит от $x(2), x(3), \dots$
- Функция не детерминированная.

Автоматные функции

Определение

Конечный автомат (преобразователь) $\mathcal{A} = (A, B, Q, F, G, q_0)$:

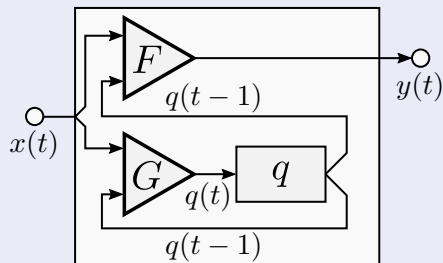
- A — непустое конечное множество (входной алфавит).
- B — непустое конечное множество (выходной алфавит).
- Q — непустое конечное множество (множество состояний).
- $F: A \times Q \rightarrow B$ — функция выхода.
- $G: A \times Q \rightarrow Q$ — функция переходов.
- $q_0 \in Q$ — начальное состояние.

Автомат $\mathcal{A}$ реализует функцию $\varphi: A^\infty \rightarrow B^\infty$: для любого $x \in A^\infty$ верно $\varphi(x) = y \in B^\infty$, где для подходящего $q = q(0)q(1)q(2) \dots \in Q^\infty$

$$\begin{cases} y(t) = F(x(t), q(t-1)), & t \in \mathbb{N}, \\ q(t) = G(x(t), q(t-1)), & t \in \mathbb{N}, \\ q(0) = q_0. \end{cases}$$

Автоматные функции

Схема работы автомата



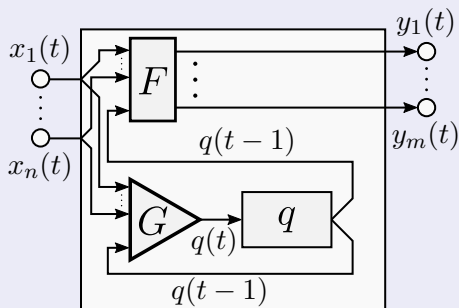
$t = 1, 2, \dots$ — время

В начальный момент $q = q_0$

$$\begin{cases} y(t) = F(x(t), q(t-1)), & t \in \mathbb{N}, \\ q(t) = G(x(t), q(t-1)), & t \in \mathbb{N}, \\ q(0) = q_0. \end{cases}$$

Автоматные функции

Автомат с несколькими входами и выходами



$t = 1, 2, \dots$ — время

В начальный момент $q = q_0$

$$x(t) = (x_1(t), \dots, x_n(t))$$

$$y(t) = (y_1(t), \dots, y_m(t))$$

$$\begin{cases} y(t) = F(x(t), q(t-1)), & t \in \mathbb{N}, \\ q(t) = G(x(t), q(t-1)), & t \in \mathbb{N}, \\ q(0) = q_0. \end{cases}$$

Автоматные функции

Определение

- Функция $\varphi: A^\infty \rightarrow B^\infty$ называется **автоматной** (ограниченно-детерминированной), если существует реализующий её автомат со входным алфавитом A и выходным алфавитом B .
- Любая автоматная функция является детерминированной.
- Детерминированная функция предполагает хранение в памяти всех входных данных $x(1), \dots, x(t)$ с начала работы, эти данные могут использоваться при вычислении $y(t)$.
- Автоматная функция хранит в памяти ограниченное количество информации, не зависящее от текущего такта. Это может быть часть значений $x(1), \dots, x(t)$ или какая-то агрегированная информация.

Автоматные функции

Множества функций

- Для $n, m \in \mathbb{N}$ множество всех автоматных операторов $\Phi: (\{0, 1\}^\infty)^n \rightarrow (\{0, 1\}^\infty)^m$ обозначается $P_{2,\text{од}}^{n,m}$.
- $\hat{P}_{2,\text{од}} = \bigcup_{\substack{n \in \mathbb{N} \\ m \in \mathbb{N}}} P_{2,\text{од}}^{n,m}$.
- $P_{2,\text{од}} = P_{2,\text{од}}^{1,1}$ — множество всех автоматных функций $\varphi: \{0, 1\}^\infty \rightarrow \{0, 1\}^\infty$.

Автоматные функции

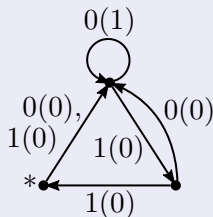
Моделирование реальных систем

- Машина Тьюринга — это модель алгоритма: процесса, который по входным данным за конечное число шагов выдаёт результат.
- Автомат-преобразователь — это модель системы, которая работает неопределённо долгое время, в каждый момент получает определённый входные сигналы и выдаёт некоторые результаты.
- Процессор компьютера является автоматом-преобразователем:
 - ▶ Состояния (конечная память) — регистры.
 - ▶ Входные сигналы — данные из оперативной памяти и с внешних устройств (клавиатуры, мыши).
 - ▶ Выходные сигналы — данные для записи в оперативную память, позиция чтения/записи в оперативной памяти, вывод на внешние устройства (дисплей).
- Автомат — вычислительно слабое устройство, так как имеет лишь конечную память. Компьютер является универсальным за счёт наличия (условно) бесконечной оперативной памяти.

Способы представления автоматов

Диаграмма Мура

- Диаграмма Мура — ориентированный псевдограф с пометками.
- Множество вершин диаграммы — это множество состояний автомата. Вершина начального состояния помечена символом $*$.
- Если автомат, получая на вход $x_1 \dots x_n$, переходит из состояния q_i в состояние q_j и выдаёт $y_1 \dots y_m$, то имеется дуга из q_i в q_j с пометкой $x_1 \dots x_n(y_1 \dots y_m)$.
- Параллельные дуги объединяем в одну с набором пометок.



Способы представления автоматов

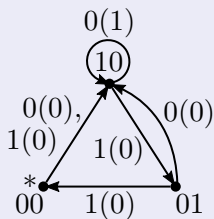
Каноническая таблица

- Кодировем состояния автомата равномерным двоичным кодом длины k . Обычно начальное состояние кодируется нулями.
- Вводим двоичные переменные:
 - ▶ Входы $x_1(t), \dots, x_n(t)$;
 - ▶ Биты кода состояния на прошлом шаге $q_1(t-1), \dots, q_k(t-1)$;
 - ▶ Выходы $y_1(t), \dots, y_m(t)$;
 - ▶ Биты кода нового состояния $q_1(t), \dots, q_k(t)$.
- Записываем в виде таблицы, как переменные $y_i(t)$ и $q_i(t)$ зависят от переменных $x_i(t)$ и $q_i(t-1)$.
- Некоторые наборы $q_1(t-1), \dots, q_k(t-1)$ не задают никакого состояния. На них доопределяем $y_i(t)$ и $q_i(t)$ произвольным образом.
- Помимо самой таблицы, требуется указать биты кода начального состояния $q_1(0), \dots, q_k(0)$.

Способы представления автоматов

Каноническая таблица (пример)

Выделены значения, выбранные произвольно.



$q_1(t-1)$	$q_2(t-1)$	$x(t)$	$q_1(t)$	$q_2(t)$	$y(t)$
0	0	0	1	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	0	1	0

$$q_1(0) = q_2(0) = 0$$

Способы представления автоматов

Канонические уравнения

- Записываем зависимость переменных $y_i(t)$ и $q_i(t)$ от переменных $x_i(t)$ и $q_i(t-1)$, а также начальное состояние, в виде уравнений.
- В примере:

$$\begin{cases} y(t) = q_1(t-1)\bar{x}(t), & t \in \mathbb{N}, \\ q_1(t) = \bar{q}_1(t-1)\overline{q_2(t-1)x(t)} \vee q_1(t-1)\bar{x}(t), & t \in \mathbb{N}, \\ q_2(t) = q_1(t-1)x(t), & t \in \mathbb{N}, \\ q_1(0) = q_2(0) = 0. \end{cases}$$

Построение автомата

[1], глава IV, 2.1 (1), с. 138.

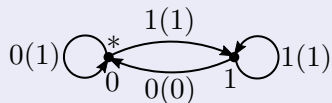
Построить диаграмму Мура, каноническую таблицу и канонические уравнения для функции $f(\tilde{x}^\omega) = y(1)y(2) \dots y(t) \dots$ из $P_{2,\text{од}}^{1,1}$:

- $$y(t) = \begin{cases} 1 & \text{при } t = 1, \\ x(t-1) \rightarrow x(t) & \text{при } t \geq 2. \end{cases}$$

Построение автомата

Решение [1], глава IV, 2.1 (1), с. 138.

Используем 2 состояния, чтобы хранить $x(t-1)$. Начальным является состояние 0, в котором для любого входа будет выдано 1.



$$\begin{cases} y(t) = q(t-1) \rightarrow x(t), \\ q(t) = x(t), \\ q(0) = 0. \end{cases}$$

$q(t-1)$	$x(t)$	$q(t)$	$y(t)$
0	0	0	1
0	1	1	1
1	0	0	0
1	1	1	1

$$q(0) = 0$$

Построение автомата

[1], глава IV, 2.1 (6), с. 138.

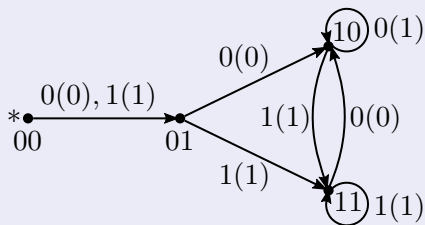
Построить диаграмму Мура, каноническую таблицу и канонические уравнения для функции $f(\tilde{x}^\omega) = y(1)y(2) \dots y(t) \dots$ из $P_{2,\text{од}}^{1,1}$:

$$\bullet y(t) = \begin{cases} x(1) & \text{при } t = 1, \\ x(2) & \text{при } t = 2, \\ \bar{x}(t-1) & \text{при } t \geq 3. \end{cases}$$

Построение автомата

Решение [1], глава IV, 2.1 (6), с. 138.

Используем 2 состояния, чтобы хранить $x(t-1)$, и ещё два состояния для первых двух тактов.



$$\begin{cases} y(t) = \bar{q}_1(t-1)x(t) \vee q_1(t-1)\bar{q}_2(t-1) \\ q_1(t) = q_1(t-1) \vee q_2(t-1), \\ q_2(t) = \bar{q}_1(t-1)\bar{q}_2(t-1) \vee x(t), \\ q_1(0) = q_2(0) = 0. \end{cases}$$

q_1	q_2	x	q_1	q_2	y
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	0

$$q_1(0) = q_2(0) = 0$$

Построение автомата

[1], глава IV, 2.1 (16), с. 138.

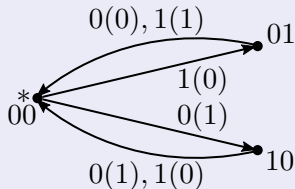
Построить диаграмму Мура, каноническую таблицу и канонические уравнения для функции $f(\tilde{x}^\omega) = y(1)y(2) \dots y(t) \dots$ из $P_{2,\text{од}}^{1,1}$:

- $$y(t) = \begin{cases} \bar{x}(t), & \text{если } t \text{ нечётное,} \\ y(t-1) \oplus x(t), & \text{если } t \text{ чётное.} \end{cases}$$

Построение автомата

Решение [1], глава IV, 2.1 (16), с. 138.

Используем 2 состояния для хранения $\bar{x}(t-1)$ на чётном такте и ещё одно состояние для определения нечётного такта. В таблице выделены значения, выбранные произвольно.



$$\begin{cases} y(t) = q_2(t-1) \sim x(t), \\ q_1(t) = \bar{q}_1(t-1)\bar{q}_2(t-1)\bar{x}(t), \\ q_2(t) = \bar{q}_1(t-1)\bar{q}_2(t-1)x(t), \\ q_1(0) = q_2(0) = 0. \end{cases}$$

q_1	q_2	x	q_1	q_2	y
0	0	0	1	0	1
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	0	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	1

$$q_1(0) = q_2(0) = 0$$

Построение диаграммы Мура по уравнениям

Пример

- Канонические уравнения:

$$\begin{cases} y(t) = \overline{x(t)q(t-1)}(x(t) \vee \bar{q}(t-1)), \\ q(t) = x(t) \vee \bar{q}(t-1), \\ q(0) = 0. \end{cases}$$

- Каноническая таблица:

$q(t-1)$	$x(t)$	$q(t)$	$y(t)$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	0

$$q(0) = 0$$

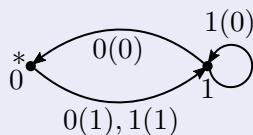
Построение диаграммы Мура по уравнениям

Получение диаграммы Мура

- Каждому из 2^k возможных наборов значений $q_1 \dots q_k$ сопоставляем состояние автомата (вершину диаграммы Мура).
- Начальное состояние соответствует набору $q_1(0) \dots q_k(0)$.
- По каждой строке таблицы добавляем дугу в диаграмму из вершины $q_1(t-1) \dots q_k(t-1)$ в вершину $q_1(t) \dots q_k(t)$ с пометкой $x_1(t) \dots x_n(t) (y_1(t) \dots y_m(t))$.
- Объединяем параллельные дуги в одну с набором пометок.
- Состояния, недостижимые из начального, можно удалить.

$q(t-1)$	$x(t)$	$q(t)$	$y(t)$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	0

$$q(0) = 0$$



Вес автоматной функции

Определение

- Пусть $\mathcal{A} = (A, B, Q, F, G, q_0)$ — автомат, $q_i \in Q$, $u \in A^*$.
- Через $F(u, q_i)$ обозначим слово $v \in B^*$ длины $|u|$, которое выдаст автомат $\mathcal{A}$ во время обработки слова u , если в начале обработки он находится в состоянии q_i .
- Состояния $q_i, q_j \in Q$ автомата $\mathcal{A}$ **отличимы словом** u , если $F(u, q_i) \neq F(u, q_j)$.
- Состояния $q_i, q_j \in Q$ автомата $\mathcal{A}$ **эквивалентны** (неотличимы), если они неотличимы никакими словами.

Теорема (Мур)

Пусть $\mathcal{A} = (A, B, Q, F, G, q_0)$ — автомат, $|Q| = r$, $q_i, q_j \in Q$. Если состояния q_i и q_j автомата $\mathcal{A}$ не эквивалентны, то они отличимы словом длины $r - 1$ или менее.

Вес автоматной функции

Приведённая диаграмма Мура

- Диаграмма Мура называется **приведённой**, если в ней все состояния достижимы и нет эквивалентных состояний.
- **Построение приведённой диаграммы**: удаляем все недостижимые состояния и склеиваем эквивалентные. При склейке все входящие дуги сохраняются, а исходящие дуги берутся от одной (любой) из склеиваемых вершин, исходящие дуги другой вершины удаляются.

Вес функции

- **Вес** автоматной функции — это минимально возможное число состояний в автомате, реализующем эту функцию.
- Вес автоматной функции равен числу вершин в любой приведённой диаграмме Мура, реализующей данную функцию.

Вес автоматной функции

[1], глава IV, 2.4 (2), с. 142.

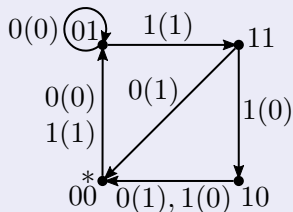
Найти вес о.-д. функции f из $P_{2,\text{од}}^{1,1}$, заданной каноническими уравнениями:

$$\bullet f: \begin{cases} y(t) = x(t) \oplus q_1(t-1), \\ q_1(t) = x(t) \cdot q_2(t-1), \\ q_2(t) = \bar{q}_1(t-1), \\ q_1(0) = q_2(0) = 0. \end{cases}$$

Вес автоматной функции

Решение [1], глава IV, 2.4 (2), с. 142.

- Всего возможны 4 состояния: 00, 01, 10, 11, начальное — 00. Все состояния достижимы.
- Диаграмма Мура:



- Состояние 00 отличимо от 10 и 11 словом 0, а от 01 словом 10. Состояние 01 отличимо от 10 и 11 словом 0. Состояния 10 и 11 отличимы словом 10.
- Вес функции равен 4.

Частичные функции

Частичная функция

Частичная функция — это функция, которая определена только на части элементов данного множества. Считаем, что на остальных элементах выход функции «не имеет значения».

Частичный автомат

- Если в диаграмме Мура простого автомата удалить некоторые дуги (либо не удалять никаких), получим **частичную диаграмму Мура**, которая задаёт **частичный автомат**.
- Частичный автомат может вычислять частичную функцию, если на её области определения автомат не попадает в ситуацию отсутствия нужной дуги.
- Доопределение частичного автомата — это добавление в него недостающих дуг (ведущих в любые вершины и с любыми выходами).

Построение частичной диаграммы Мура

[1], глава IV, 2.5 (2), с. 143.

Для частично определённой д. функции $f(\tilde{x}^\omega): \{0, 1\}^\omega \rightarrow \{0, 1\}^\omega$, отображающей заданные последовательности в заданные, построить диаграмму Мура с возможно меньшим числом вершин; затем полученную диаграмму Мура доопределить до диаграммы Мура всюду определённой о.-д. функции из $P_{2,\text{од}}^{1,1}$, и для этой новой функции построить каноническую таблицу и канонические уравнения:

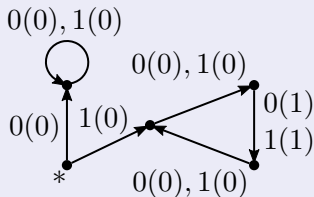
- $f([01]^\omega) = \tilde{0}^\omega$ и $f(1[10]^\omega) = [001]^\omega$.

Построение частичной диаграммы Мура

Решение [1], глава IV, 2.5 (2), с. 143.

$$f([01]^\omega) = \tilde{0}^\omega, \quad f(1[10]^\omega) = [001]^\omega$$

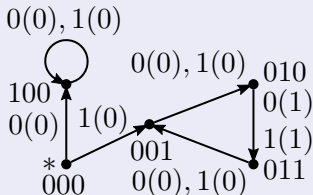
У слов $[01]^\omega$ и $1[10]^\omega$ нет общего префикса, поэтому функция может быть детерминированной. Различаем, что нужно выдавать, по первому символу и реализуем бесконечные повторы слов по отдельности. Диаграмма сразу получается полностью определённой.



Построение частичной диаграммы Мура

Решение [1], глава IV, 2.5 (2), с. 143 (продолжение).

Выделенные значения выбраны произвольно.



$$\begin{cases} y(t) = q_2(t-1)\bar{q}_3(t-1), \\ q_1(t) = \bar{q}_2(t-1)\bar{q}_3(t-1)\bar{x}(t) \vee q_1(t-1), \\ q_2(t) = \bar{q}_1(t-1)(q_2(t-1) \sim q_3(t-1)), \\ q_3(t) = \bar{q}_1(t-1)\bar{q}_3(t-1)x(t) \vee q_2(t-1), \\ q_1(0) = q_2(0) = q_3(0) = 0. \end{cases}$$

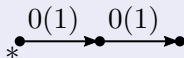
q_1	q_2	q_3	x	q_1	q_2	q_3	y
0	0	0	0	1	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	1	1
1	1	0	1	1	0	1	1
1	1	1	0	1	0	1	0
1	1	1	1	1	0	1	0

$$q_1(0) = q_2(0) = q_3(0) = 0$$

Построение частичной диаграммы Мура

Построение отдельных контуров для последовательностей

- Метод построения частичной диаграммы, применённый в задаче 2.5 (2), позволяет строить диаграммы для любых подобных задач.
- Пусть $f(x) = u(v)^\omega$ и $f(x') = u'(v')^\omega$. Строим цепь из начального состояния для максимального по длине общего префикса x и x' , выдаём на этой цепи префикс $u(v)^\omega$ или $u'(v')^\omega$ той же длины.
- Например, для $f(001[10]^\omega) = 1100[1]^\omega$, $f(0^\omega) = 11[10]^\omega$:

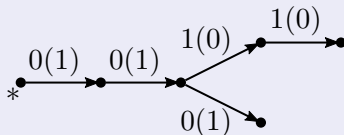


- Если префиксы $u(v)^\omega$ и $u'(v')^\omega$ этой длины различаются, то построение диаграммы невозможно (f не детерминированная).

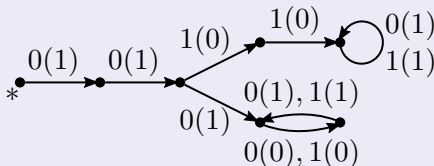
Построение частичной диаграммы Мура

Построение отдельных контуров для последовательностей

- Делаем разветвление по первому отличающемуся символу x и x' и реализуем оставшиеся части u и u' цепями (если требуется).
- Для $f(001[10]^\omega) = 1100[1]^\omega$, $f(0^\omega) = 11[10]^\omega$:



- Реализуем $(v)^\omega$ и $(v')^\omega$ с помощью контуров:



Построение частичной диаграммы Мура

Построение отдельных контуров для последовательностей

- При построении диаграммы этим методом можно заранее вычислить её размер. Пусть $f(x) = u(v)^\omega$ и $f(x') = u'(v')^\omega$, а n — максимальная длина общего префикса x и x' . Тогда число вершин в диаграмме равно

$$n + 1 + (|u| \dot{-} (n + 1)) + (|u'| \dot{-} (n + 1)) + |v| + |v'|,$$

где $a \dot{-} b = \begin{cases} a - b, & \text{если } a \geq b, \\ 0, & \text{если } a < b. \end{cases}$

- Данный метод из информации об x и x' использует только некоторые их префиксы: построение контуров диаграммы от x и x' не зависит.
- Другие методы, использующие информацию об x и x' , позволяют строить диаграммы меньшего размера.

Построение частичной диаграммы Мура

Перебор построений: отдельное построение

- Пусть $f(x) = y$ и $f(x') = y'$, где

$$x = w(z)^\omega, \quad x' = w'(z')^\omega, \quad y = u(v)^\omega, \quad y' = u'(v')^\omega.$$

- Пусть n — максимальная длина общего префикса x и x' . Если префиксы длины n слов y и y' различаются, то диаграмму построить невозможно (f не детерминированная).
- Строим диаграмму, добавляя дуги по одной. По мере построения диаграмма будет реализовывать префиксы последовательностей всё большей длины.
- Начинаем с единственного начального состояния, которое считаем текущим для обеих последовательностей (x и x'). Текущий символ в обеих последовательностях — первый.

Построение частичной диаграммы Мура

Перебор построений: шаг отдельного построения

- Произвольно выбираем последовательность (x или x'), для которой строим очередную дугу.
- Дуга будет исходить из текущего состояния выбранной последовательности.
- Дуга будет помечена текущим символом выбранной последовательности и соответствующим ему по условию задачи выходным символом (то есть пометка имеет вид $x(i)(y(i))$ для первой последовательности и $x'(j)(y'(j))$ для второй).
- Произвольно выбираем состояние, в которое входит дуга: либо берём одно из разрешённых существующих, либо создаём новое состояние и ведём дугу в него.

Построение частичной диаграммы Мура

Перебор построений: шаг отдельного построения

- Проводить дугу в существующее состояние запрещено, если выполняется одно из условий:
 - ▶ В случае добавления дуги диаграмма на некотором префиксе x будет выдавать не начальную часть слова y , либо на некотором префиксе x' будет выдавать не начальную часть слова y' .
 - ▶ В случае добавления дуги диаграмма после реализации максимального по длине префикса x и реализации максимального по длине префикса x' придёт в одно и то же состояние, причём оставшиеся нереализованными части x и x' будут иметь общий префикс и на этом префиксе диаграмма (в случае дальнейшего построения) должна будет выдавать для x и x' разные выходы.
- Реализацией максимального по длине префикса x считаем обход диаграммы из начального состояния в соответствии со словом x до тех пор, пока в диаграмме присутствуют нужные дуги.

Построение частичной диаграммы Мура

Перебор построений: шаг отдельного построения

- После добавления дуги изменяем текущие состояния и текущие символы: продолжаем обход диаграммы в соответствии с x и x' , пока не остановимся из-за отсутствия дуги. Состояние и символ последовательности, на котором мы остановились, становятся текущими для данной последовательности.
- Второе ограничение на предыдущем слайде можно переформулировать: нельзя, чтобы после проведения дуги одна и та же вершина стала текущей для обеих последовательностей, если продолжения последовательностей из этой вершины имеют общий префикс, на котором f должна давать разные выходы.
- Если после добавления дуги диаграмма реализует последовательность x или x' (то есть обход диаграммы в соответствии с этой последовательностью будет бесконечным), то продолжаем строить только оставшуюся последовательность.

Построение частичной диаграммы Мура

Перебор построений

- Перебирая различные способы выбора вершины, в которую идёт каждая новая дуга, можно получить все возможные диаграммы Мура, реализующие требуемую функцию. Среди них есть и диаграмма с наименьшим числом вершин.
- Для полного перебора следует зафиксировать принцип выбора последовательности на каждом шаге (например, сначала строить только первую последовательность, а вторую начинать только после полной реализации первой).
- Перебор следует начинать с «жадного» построения: на каждом шаге, когда возможно, проводить дуги в существующие вершины.
- Далее нужно отсекаать те ветви перебора, в которых количество вершин оказывается больше, чем в «жадном» варианте (не достраивая такие диаграммы). Это сделает перебор конечным.

Построение частичной диаграммы Мура

Перебор построений

- «Жадное» построение диаграммы тоже определяется неоднозначно.
- Вместо полного перебора можно попытаться угадать удачный способ выбора вершин и построить диаграмму не слишком большого размера. В частности, этим вариантом может оказаться одно из «жадных» построений.
- Для такого угадывания может быть удобно на каждом шаге индивидуально выбирать, для какой последовательности проводить новую дугу.

Построение частичной диаграммы Мура

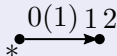
«Жадное» построение: пример

$$f(001[10]^\omega) = 1100[1]^\omega, \quad f(0^\omega) = 11[10]^\omega$$

- Текущие состояния последовательностей $x = 001[10]^\omega$ и $x' = 0^\omega$ будем отмечать цифрами 1 и 2. Сначала полностью реализуем последовательность x , затем переходим к реализации x' .
- Начинаем с единственного начального состояния и без дуг:



- Дуга $0(1)$. Если провести её в виде петли начального состояния, то на слове 0000 диаграмма будет давать 1111 , что противоречит условию. Добавляем новое состояние, оно становится текущим для обеих последовательностей:

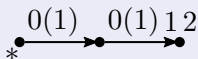


Построение частичной диаграммы Мура

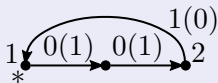
«Жадное» построение: пример

$$f(001[10]^\omega) = 1100[1]^\omega, \quad f(0^\omega) = 11[10]^\omega$$

- Дуга $0(1)$. Если провести её в любое из существующих состояний, то на слове 0000 диаграмма будет давать 1111 , что противоречит условию. Добавляем новое состояние, оно становится текущим для обеих последовательностей:



- Дуга $1(0)$. Проведение её в начальную вершину не нарушает никаких ограничений:

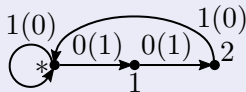


Построение частичной диаграммы Мура

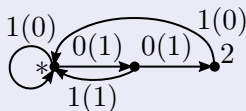
«Жадное» построение: пример

$$f(001[10]^\omega) = 1100[1]^\omega, \quad f(0^\omega) = 11[10]^\omega$$

- Дуга $1(0)$. Проведение её в виде петли не нарушает никаких ограничений. Текущей для x станет вторая добавленная вершина:



- Дуга $1(1)$. Её можно провести в начальную вершину. После этого последовательность x окажется полностью реализованной:

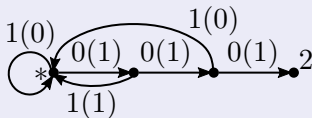


Построение частичной диаграммы Мура

«Жадное» построение: пример

$$f(001[10]^\omega) = 1100[1]^\omega, \quad f(0^\omega) = 11[10]^\omega$$

- Начинаем реализацию последовательности x' . Префикс 00 уже реализован. Далее идёт дуга 0(1).
- Если дугу 0(1) провести в любое из существующих состояний, то на слове 0000 диаграмма будет давать 1111, что противоречит условию. Добавляем новое состояние, оно становится текущим для x' :

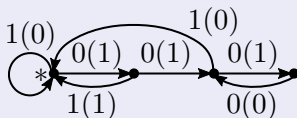


Построение частичной диаграммы Мура

«Жадное» построение: пример

$$f(001[10]^\omega) = 1100[1]^\omega, \quad f(0^\omega) = 11[10]^\omega$$

- Дуга $0(0)$. Проведение её в предыдущую вершину не нарушает никаких ограничений. После этого вторая последовательность оказывается реализованной:



- Получили требуемую диаграмму. Она имеет 4 вершины. Для сравнения, диаграмма, в которой те же последовательности реализовывались отдельными контурами, имела 7 вершин.

Домашнее задание

[1], глава IV, 1.1 (4, 6, 11), с. 111.

Выяснить, является ли функция $y = f(x)$ детерминированной.

4. $y(t) = x(\lfloor \log_2 t \rfloor + 1), t \geq 1.$

6. $y(1) = y(2) = 1, y(t) = x(2^{t-1} - t), t \geq 3.$

13. $y(t) = \begin{cases} x(20t - t^2 - 90), & 7 \leq t \leq 13, \\ 0, & \text{иначе.} \end{cases}$

[1], глава IV, 1.2 (2), с. 111.

Выяснить, является ли функция $y = f(x)$ детерминированной.

2. $f(\tilde{x}^\omega) = \begin{cases} \tilde{1}^\omega, & \tilde{x}^\omega = \tilde{0}^\omega \\ \bar{x}(1)\bar{x}(2)\dots, & \text{иначе.} \end{cases}$

Домашнее задание

[1], глава IV, 2.1 (3, 7), с. 138.

Построить диаграмму Мура, каноническую таблицу и канонические уравнения для функции.

$$3. \ y(t) = \begin{cases} 1 & \text{при } t = 1, \\ y(t-1) \cdot x(t) & \text{при } t \geq 2. \end{cases}$$

$$7. \ y(t) = \begin{cases} x(2), & \text{если } t = 2, \\ 1 & \text{в остальных случаях.} \end{cases}$$

Домашнее задание

[1], глава IV, 2.4 (1, 3), с. 142.

Найти вес функции, заданной каноническими уравнениями.

$$1. \quad f: \begin{cases} y(t) = \bar{x}(t) \rightarrow q_1(t-1) \cdot q_2(t-1), \\ q_1(t) = \bar{q}_1(t-1) \rightarrow q_2(t-1), \\ q_2(t) = \bar{q}_2(t-1) \rightarrow q_1(t-1), \\ q_1(0) = q_2(0) = 1. \end{cases}$$

$$3. \quad f: \begin{cases} y(t) = x(t) \vee q_1(t-1) \cdot q_2(t-1), \\ q_1(t) = q_1(t-1) \vee q_2(t-1), \\ q_2(t) = q_2(t-1) \vee q_1(t-1), \\ q_1(0) = q_2(0) = 1. \end{cases}$$

Домашнее задание

[1], глава IV, 2.5 (3), с. 143.

Для частично определённой д. функции $f(\tilde{x}^\omega): \{0, 1\}^\omega \rightarrow \{0, 1\}^\omega$, отображающей заданные последовательности в заданные, построить диаграмму Мура с возможно меньшим числом вершин; затем полученную диаграмму Мура доопределить до диаграммы Мура всюду определённой о.-д. функции из $P_{2,\text{од}}^{1,1}$, и для этой новой функции построить каноническую таблицу и канонические уравнения:

- $f([101]^\omega) = [110]^\omega$, $f([010]^\omega) = [1011]^\omega$.

Список литературы

1. Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по дискретной математике. — М.: Физматлит, 2005. — 416 с.

Дополнительные главы дискретной математики
и кибернетики
Семинар 5

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Операция суперпозиции

Вектор-функции

- Вектор-функции вида $f: (\{0, 1\}^\omega)^n \rightarrow (\{0, 1\}^\omega)^m$ рассматриваем как вектора $f = (f_1, \dots, f_m)$ скалярных функций $f_i: (\{0, 1\}^\omega)^n \rightarrow \{0, 1\}^\omega$, $i = \overline{1, m}$.

Суперпозиция для вектор-функций

- Считаем, что операция **суперпозиции** включает в себя
 - подстановку скалярной функции вместо переменной,
 - перестановку и отождествление переменных,
 - добавление/удаление фиктивных переменных,
 - группировку нескольких скалярных функций или вектор-функций (от одинакового числа переменных) в одну вектор-функцию,
 - удаление части функций из вектор-функции.
- В [1] суперпозиции соответствуют операции O_1, O_2, O_4, S .

Операция суперпозиции

[1], глава IV, 2.8 (3), с. 154.

Для суперпозиции $f = f_1(f_2)$ о.-д. функций f_1 и f_2 из $\hat{P}_{2,\text{од}}$ построить канонические уравнения и приведённую диаграмму Мура:

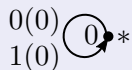
$$3. \quad f_1: \begin{cases} y_1(t) = \bar{x}_1(t), \\ q_1(t) = \bar{q}_1(t-1), \\ q_1(0) = 1, \end{cases} \quad f_2: \begin{cases} y_2(t) = \bar{q}_2(t-1), \\ q_2(t) = \bar{x}_2(t) \cdot q_2(t-1), \\ q_2(0) = 0. \end{cases}$$

Операция суперпозиции

Решение [1], глава IV, 2.8 (3), с. 154.

$$f_1: \begin{cases} y_1(t) = \bar{x}_1(t), \\ q_1(t) = \bar{q}_1(t-1), \\ q_1(0) = 1, \end{cases} \quad f_2: \begin{cases} y_2(t) = \bar{q}_2(t-1), \\ q_2(t) = \bar{x}_2(t) \cdot q_2(t-1), \\ q_2(0) = 0. \end{cases}$$

- Легко видеть, что $f_1(x) = \bar{x}$, $f_2(x) = 1^\omega$.
- Поэтому $f(x) = 0^\omega$. Это функция веса 1:



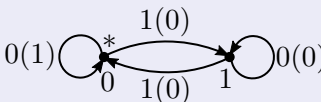
- Канонические уравнения:

$$f: \begin{cases} y(t) = 0, \\ q(t) = 0, \\ q(0) = 0. \end{cases}$$

Операция суперпозиции

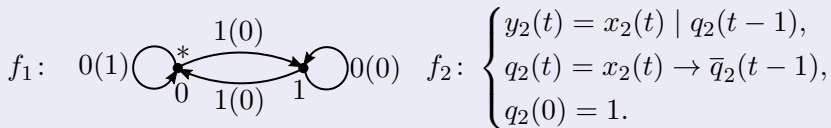
[1], глава IV, 2.8 (6), с. 154.

Для суперпозиции $f = f_1(f_2)$ о.-д. функций f_1 и f_2 из $\hat{P}_{2,од}$ построить канонические уравнения и приведённую диаграмму Мура:

6. f_1 :  f_2 :
$$\begin{cases} y_2(t) = x_2(t) \mid q_2(t-1), \\ q_2(t) = x_2(t) \rightarrow \bar{q}_2(t-1), \\ q_2(0) = 1. \end{cases}$$

Операция суперпозиции

Решение [1], глава IV, 2.8 (6), с. 154.



- $f_1: \begin{cases} y_1(t) = \bar{x}_1(t)\bar{q}_1(t-1), \\ q_1(t) = x_1(t) \oplus q_1(t-1), \\ q_1(0) = 0, \end{cases}$

- Объединяем канонические уравнения и подставляем y_2 вместо x_1 :

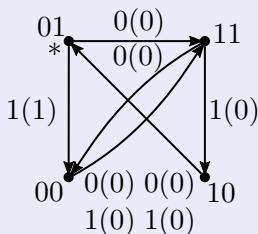
$$f: \begin{cases} y_1(t) = x_2(t)q_2(t-1)\bar{q}_1(t-1), \\ q_1(t) = \overline{(x_2(t)q_2(t-1))} \oplus q_1(t-1), \\ q_2(t) = \bar{x}_2(t) \vee \bar{q}_2(t-1), \\ q_1(0) = 0, \quad q_2(0) = 1. \end{cases}$$

Операция суперпозиции

Решение [1], глава IV, 2.8 (6), с. 154 (продолжение).

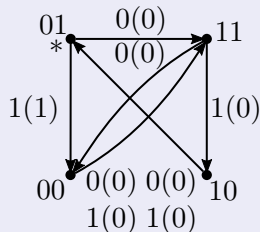
$$f: \begin{cases} y(t) = x(t)\bar{q}_1(t-1)q_2(t-1), \\ q_1(t) = \overline{(x(t)q_2(t-1))} \oplus q_1(t-1), \\ q_2(t) = \bar{x}_2(t) \vee \bar{q}_2(t-1), \\ q_1(0) = 0, \quad q_2(0) = 1. \end{cases}$$

- Всего возможны 4 состояния: 00, 01, 10, 11, начальное — 00. Все состояния достижимы.



Операция суперпозиции

Решение [1], глава IV, 2.8 (6), с. 154 (продолжение).



- Состояние 01 отличимо от всех остальных словом 1. Состояние 10 Отличимо от 00 и 11 словом 01. Состояния 00 и 11 отличимы словом 101.
- Поэтому указанная диаграмма является приведённой.

Введение обратной связи

Определение

У детерминированной функции $(y_1, \bar{y}) = f(x_1, \bar{x})$ выход y_1 **зависит** от x_1 **с запаздыванием**, если $y_1(t)$ не зависит от $x_1(t)$.

Введение обратной связи

- Пусть у $(y_1, \bar{y}) = f(x_1, \bar{x})$ выход y_1 зависит от x_1 с запаздыванием.
- Подадим сигнал с выхода y_1 на вход x_1 : в канонических уравнениях подставим выражение для $y_1(t)$ вместо $x_1(t)$.
- Удалим сам выход y_1 .
- В силу зависимости с запаздыванием в получившихся канонических уравнениях вход x_1 исчезает.
- Получим функцию $\bar{y} = g(\bar{x})$.
- Введение обратной связи сохраняет автоматность функции.
- В [1] введению обратной связи соответствует операция O_3 .

Введение обратной связи

[1], глава IV, 2.9 (4), с. 155.

Построить канонические уравнения и приведённую диаграмму Мура о.-д. функции, получающейся из функции f введением обратной связи по переменным x_i, y_j :

$$4. f: \begin{cases} y_1(t) = x_1(t) \cdot q(t-1) \vee x_2(t), \\ y_2(t) = x_2(t), \\ q(t) = (x_1(t) \rightarrow x_2(t)) \vee q(t-1), \\ q(0) = 0, \end{cases} \quad i = 1, j = 2.$$

Введение обратной связи

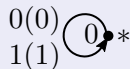
Решение [1], глава IV, 2.9 (4), с. 155.

$$f: \begin{cases} y_1(t) = x_1(t) \cdot q(t-1) \vee x_2(t), \\ y_2(t) = x_2(t), \\ q(t) = (x_1(t) \rightarrow x_2(t)) \vee q(t-1), \\ q(0) = 0, \end{cases} \quad x_1 := y_2.$$

- Подставляем выход $y_2(t) = x_2(t)$ вместо входа $x_1(t)$.

$$f: \begin{cases} y_1(t) = x_2(t) \cdot q(t-1) \vee x_2(t), \\ q(t) = (x_2(t) \rightarrow x_2(t)) \vee q(t-1), \\ q(0) = 0, \end{cases} \quad f: \begin{cases} y(t) = x(t), \\ q(t) = 1, \\ q(0) = 0, \end{cases}$$

- Получили $f(x) = x$. Это функция веса 1:



Некоторые автоматные функции

Истинностные функции

- Пусть $f: \{0, 1\}^n \rightarrow \{0, 1\}$ — булева функция. Ей соответствует **истинностная функция** $f': (\{0, 1\}^\infty)^n \rightarrow \{0, 1\}^\infty$ такая, что

$$f'(x_1, \dots, x_n) = f(x_1(1), \dots, x_n(1))f(x_1(2), \dots, x_n(2)) \dots$$

Она реализуется автоматом с одним состоянием ($Q = \{0\}$):

$$\begin{cases} y(t) = f(x_1(t), \dots, x_n(t)), & t \in \mathbb{N}, \\ q(t) = 0, & t \in \mathbb{N} \cup \{0\}. \end{cases}$$

Для f' может использоваться то же обозначение, что и для f :

$$\begin{aligned} \neg x &= \overline{x}(1)\overline{x}(2) \dots, & \neg[01]^\omega &= [10]^\omega; \\ x \&y &= (x(1) \& y(1))(x(2) \& y(2)) \dots, & [01]^\omega \& [10]^\omega &= 0^\omega. \end{aligned}$$

Некоторые автоматные функции

Единичная задержка

- **Единичная задержка** $z: \{0, 1\}^\infty \rightarrow \{0, 1\}^\infty$, $z(x) = 0x$ реализуется автоматом с двумя состояниями ($Q = \{0, 1\}$):

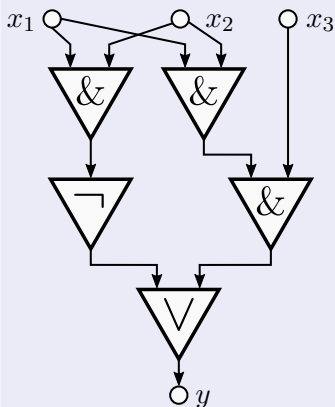
$$\begin{cases} y(t) = q(t-1), & t \in \mathbb{N}, \\ q(t) = x(t), & t \in \mathbb{N}, \\ q(0) = 0. \end{cases}$$

- Другой вариант задержки $z_1: \{0, 1\}^\infty \rightarrow \{0, 1\}^\infty$, $z_1(x) = 1x$ реализуется автоматом с двумя состояниями ($Q = \{0, 1\}$):

$$\begin{cases} y(t) = q(t-1), & t \in \mathbb{N}, \\ q(t) = x(t), & t \in \mathbb{N}, \\ q(0) = 1. \end{cases}$$

- Можно выразить z_1 через z : $z_1(x) = \neg z(\neg x)$.

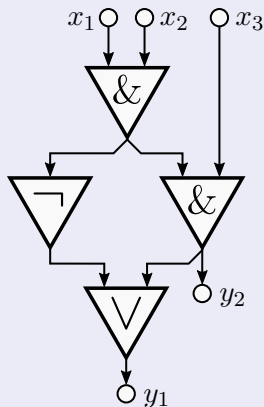
Формула



$$y = \overline{x_1 x_2} \vee (x_1 x_2) x_3$$

- Число входных полюсов равно числу различных символов переменных.
- Во входные полюса не входят дуги, но из них может исходить любое число дуг.
- Выходной полюс один, в него входит одна дуга, и из него не исходит дуг.
- В функциональный элемент входит по одной дуге на каждый аргумент его функции, а исходит ровно одна дуга.
- Формула без входных полюсов представляет собой дерево.
- Формула реализует булеву функцию.

Схема из функциональных элементов (СФЭ)



$$y_1 = \overline{x_1 x_2} \vee (x_1 x_2) x_3$$

$$y_2 = x_1 x_2 x_3$$

Отличия от формулы:

- Из каждого функционального элемента может исходить любое число дуг.
- СФЭ может не являться деревом, но в ней нет контуров.
- Выходных полюсов может быть несколько.
- СФЭ реализует булев оператор

$$F: \{0, 1\}^n \rightarrow \{0, 1\}^m.$$

- В примере $F = (\overline{x_1 x_2} \vee x_1 x_2 x_3, x_1 x_2 x_3)$.

Схема из функциональных элементов и элементов задержки

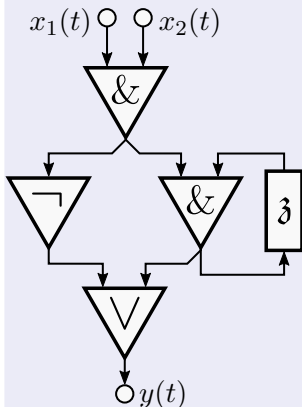
Отличия от СФЭ:

- Допускаются элементы, реализующие функцию единичной задержки.
- Допускаются контуры, но каждый контур должен содержать элемент задержки.
- Схема реализует автоматный оператор

$$F: (\{0, 1\}^\infty)^n \rightarrow (\{0, 1\}^\infty)^m.$$

- В примере

$$\begin{cases} y(t) = \overline{x_1(t)x_2(t)} \vee x_1(t)x_2(t)q(t-1), \\ q(t) = x_1(t)x_2(t)q(t-1), & t \in \mathbb{N}, \\ q(0) = 0. \end{cases}$$



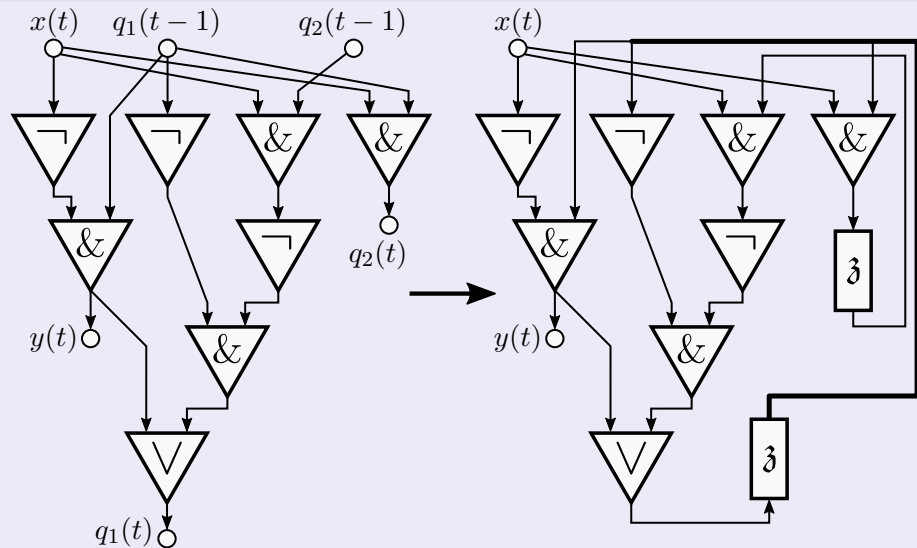
Построение схемы автомата

- По каноническим уравнениям строим СФЭ со входами $x_i(t)$, $q_i(t-1)$ и выходами $y_i(t)$, $q_i(t)$, реализующую соответствующие функции.
- Если $q_i(0) = 0$, соединяем выход $q_i(t-1)$ со входом $q_i(t)$ через элемент задержки z .
- Если $q_i(0) = 1$, соединяем выход $q_i(t-1)$ со входом $q_i(t)$ через z_1 , реализуя её с помощью элемента задержки и двух элементов отрицания: $z_1 = \neg z \neg$.
- Пример схемы для данных уравнений на следующем слайде:

$$\begin{cases} y(t) = q_1(t-1)\bar{x}(t), & t \in \mathbb{N}, \\ q_1(t) = \bar{q}_1(t-1)\overline{q_2(t-1)x(t)} \vee q_1(t-1)\bar{x}(t), & t \in \mathbb{N}, \\ q_2(t) = q_1(t-1)x(t), & t \in \mathbb{N}, \\ q_1(0) = q_2(0) = 0. \end{cases}$$

Схемы

Построение схемы автомата (пример)



Построение схемы автомата

- Построение СФЭ осуществляется при помощи операции суперпозиции, применяемой к истинностным функциям.
- Добавление задержек на места выходов $q_i(t)$ осуществляется при помощи операции суперпозиции, применяемой к истинностным функциям и функции задержки.
- Выходы задержек после этого зависят с запаздыванием от любых входных переменных.
- Соединение выходов задержек со входами $q_i(t - 1)$ производится при помощи операции обратной связи.
- Таким образом, любую автоматную функцию можно получить из функций $\{\&, \vee, \neg, \exists\}$ с помощью операций суперпозиции и обратной связи.

Построение схем автоматов

[1], глава IV, 2.13 (1), с. 167.

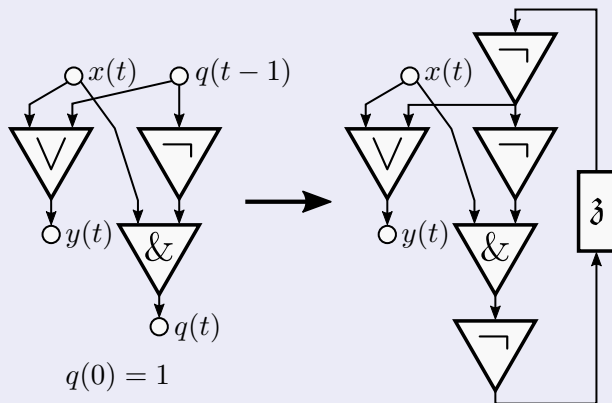
Для функции f из $P_{2,\text{од}}$ построить схему над множеством, состоящим из элемента единичной задержки и функций, порождённых дизъюнкцией, конъюнкцией и отрицанием.

$$\bullet f: \begin{cases} y(t) = x(t) \vee q(t-1), \\ q(t) = x(t) \cdot \bar{q}(t-1), \\ q(0) = 1. \end{cases}$$

Построение схем автоматов

Решение [1], глава IV, 2.13 (1), с. 167.

Поскольку $q(0) = 1$, используем $\mathfrak{z}_1 = \neg \mathfrak{z} \neg$.



Построение схем автоматов

[1], глава IV, 2.13 (4), с. 167.

Для функции f из $P_{2,\text{од}}$ построить схему над множеством, состоящим из элемента единичной задержки и функций, порождённых дизъюнкцией, конъюнкцией и отрицанием.

- Функция f задаётся $q(0) = 0$ и канонической таблицей:

$x(t)$	$q(t - 1)$	$y(t)$	$q(t)$
0	0	1	1
0	1	0	1
1	0	0	0
1	1	1	0

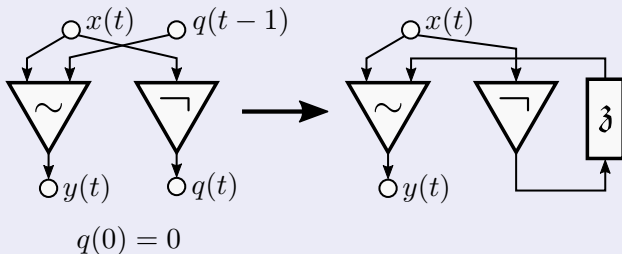
Построение схем автоматов

Решение [1], глава IV, 2.13 (4), с. 167.

- Канонические уравнения:

$$f: \begin{cases} y(t) = x(t) \sim q(t-1), \\ q(t) = \overline{x(t)}, \\ q(0) = 0. \end{cases}$$

- Схема:



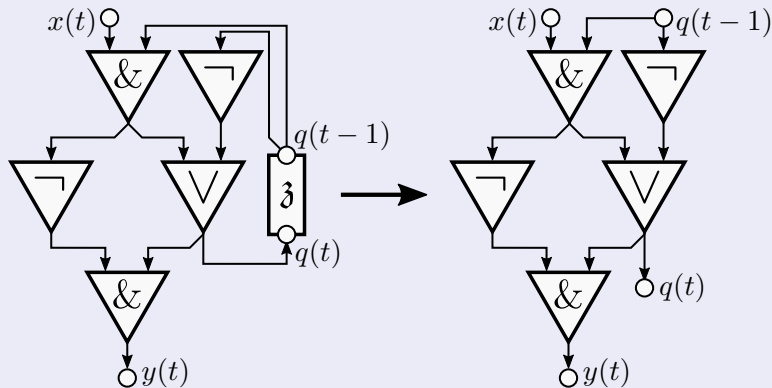
Построение автомата по схеме

Получение канонических уравнений

- Сопоставляем каждому элементу задержки схемы разряд состояния q_i .
- Заменяем каждый элемент задержки на пару полюсов: входной полюс $q_i(t-1)$, из которого идут исходящие дуги элемента, и выходной полюс $q_i(t)$, в который идут заходящие дуги элемента.
- По СФЭ выписываем зависимость $y_1(t), \dots, y_m(t), q_1(t), \dots, q_k(t)$ от $x_1(t), \dots, x_n(t), q_1(t-1), \dots, q_k(t-1)$ в виде уравнений.
- Добавляем условие $q_1(0) = \dots = q_k(0) = 0$.
Получаем канонические уравнения.

Построение автомата по схеме

Пример

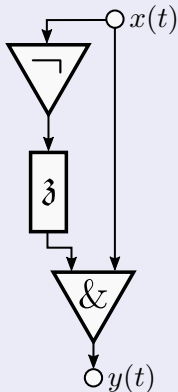


$$\begin{cases} y(t) = \overline{x(t)q(t-1)}(x(t) \vee \bar{q}(t-1)), \\ q(t) = x(t) \vee \bar{q}(t-1), \\ q(0) = 0. \end{cases}$$

Построение автомата по схеме

[1], глава IV, 2.14 (1), с. 168.

По схеме, реализующей функцию из $\hat{P}_{2,\text{од}}$, построить канонические уравнения, каноническую таблицу и диаграмму Мура:



Построение автомата по схеме

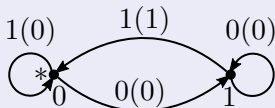
Решение [1], глава IV, 2.14 (1), с. 168.

- Задержке присваиваем бит q .
- Канонические уравнения и каноническая таблица ($q(0) = 0$):

$$\begin{cases} y(t) = x(t)q(t-1), \\ q(t) = \bar{x}(t), \\ q(0) = 0. \end{cases}$$

$q(t-1)$	$x(t)$	$q(t)$	$y(t)$
0	0	1	0
0	1	0	0
1	0	1	0
1	1	0	1

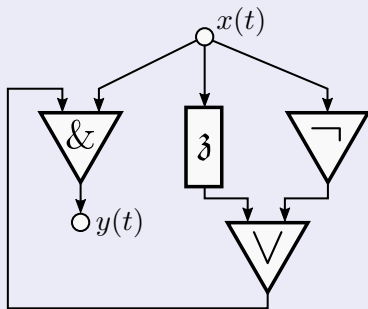
- Диаграмма Мура:



Построение автомата по схеме

[1], глава IV, 2.14 (4), с. 168.

По схеме, реализующей функцию из $\hat{P}_{2,\text{од}}$, построить канонические уравнения, каноническую таблицу и диаграмму Мура:



Построение автомата по схеме

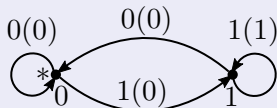
Решение [1], глава IV, 2.14 (4), с. 168.

- Задержке присваиваем бит q .
- Канонические уравнения и каноническая таблица ($q(0) = 0$):

$$\begin{cases} y(t) = x(t)(q(t-1) \vee \bar{x}(t)), \\ q(t) = x(t), \\ q(0) = 0. \end{cases}$$

$q(t-1)$	$x(t)$	$q(t)$	$y(t)$
0	0	0	0
0	1	1	0
1	0	0	0
1	1	1	1

- Диаграмма Мура:



Полнота в классе автоматных функций

Определение

Система функций F называется **полной** в классе функций A относительно набора операций $\mathcal{O}$, если класс A совпадает с классом всех функций, которые можно получить из F с помощью операций $\mathcal{O}$.

Полные системы

- Система булевых функций $\{\&, \vee, \neg\}$ полна в классе всех булевых функций P_2 относительно операции суперпозиции.
- Из построения схемы для автоматной функции следует, что система $\{\&, \vee, \neg, \exists\}$ полна в классе конечно-автоматных функций относительно операций суперпозиции и введения обратной связи.
- Функции $\&, \vee, \neg$ можно заменить на любую систему истинностных функций, соответствующую полной в P_2 системе. Например, на $\{\&, \neg\}$, $\{\vee, \neg\}$, $\{|\}$ или $\{\downarrow\}$.

Полнота в классе автоматных функций

[1], глава IV, 2.17 (1), с. 174.

Доказать полноту системы $\{f_1, f_2\}$ в $\hat{P}_{2,\text{од}}$ относительно операций суперпозиции и обратной связи.

$$\begin{aligned} 1. \quad & f_1: y(t) = \bar{x}_1(t) \cdot \bar{x}_2(t), \quad t \geq 1, \\ & f_2: \begin{cases} y(t) = x_1(t) \cdot \bar{x}_2(t) \vee q(t-1), \\ q(t) = x_1(t) \vee x_2(t), \\ q(0) = 0. \end{cases} \end{aligned}$$

Полнота в классе автоматных функций

Решение [1], глава IV, 2.17 (1), с. 174.

$$f_1: y(t) = \bar{x}_1(t) \cdot \bar{x}_2(t), \quad f_2: \begin{cases} y(t) = x_1(t) \cdot \bar{x}_2(t) \vee q(t-1), \\ q(t) = x_1(t) \vee x_2(t), \\ q(0) = 0. \end{cases}$$

- $f_1: \bar{x}_1(t)\bar{x}_2(t) = x(t) \downarrow y(t)$ — стрелка Пирса. Это полная в P_2 система.
- Отождествляем в f_2 переменные x_1 и x_2 . Получаем $\mathfrak{z}$:

$$\begin{cases} y(t) = x_1(t) \cdot \bar{x}_1(t) \vee q(t-1) = q(t-1), \\ q(t) = x_1(t) \vee x_1(t) = x_1(t), \\ q(0) = 0. \end{cases}$$

- Система $\{\downarrow, \mathfrak{z}\}$ полна в $\hat{P}_{2,\text{од}}$ относительно суперпозиции и обратной связи, поэтому полна и система $\{f_1, f_2\}$.

Полнота в классе автоматных функций

[1], глава IV, 2.17 (4), с. 174.

Доказать полноту системы $\{f_1, f_2\}$ в $\hat{P}_{2,\text{од}}$ относительно операций суперпозиции и обратной связи.

$$\begin{aligned} 4. \quad & f_1: y(t) = \bar{x}(t), \quad t \geq 1 \\ & f_2: \begin{cases} y(t) = x_1(t) \cdot x_2(t) \vee x_3(t) \cdot q(t-1), \\ q(t) = x_2(t) \cdot x_3(t), \\ q(0) = 0. \end{cases} \end{aligned}$$

Полнота в классе автоматных функций

Решение [1], глава IV, 2.17 (4), с. 174.

$$f_1: y(t) = \bar{x}(t), \quad f_2: \begin{cases} y(t) = x_1(t) \cdot x_2(t) \vee x_3(t) \cdot q(t-1), \\ q(t) = x_2(t) \cdot x_3(t), \\ q(0) = 0. \end{cases}$$

- $f_2(x, \bar{x}, x) = 0^\omega. 1^\omega = \overline{0^\omega}$.
- $\mathfrak{z}(x) = f_2(0^\omega, x, 1^\omega)$.
- $xy = f_2(x, y, 0^\omega)$,
- Система $\{xy, \bar{x}, \mathfrak{z}\}$ полна в $\hat{P}_{2, \text{од}}$ относительно суперпозиции и обратной связи, поэтому полна и система $\{f_1, f_2\}$.

Домашнее задание

[1], глава IV, 2.8 (5, 8), с. 154.

Для суперпозиции $f = f_1(f_2)$ о.-д. функций f_1 и f_2 из $\hat{P}_{2,\text{од}}$ построить канонические уравнения и приведённую диаграмму Мура:

$$5. f_1: \begin{cases} y_1(1) = 0, \\ y_1(t) = x_1(t-1) \rightarrow y_1(t-1), t \geq 2, \end{cases}$$

$$f_2: \begin{cases} y_2(t) = q_2(t-1) \rightarrow x_2(t), \\ q_2(t) = x_2(t), \\ q_2(0) = 1. \end{cases}$$

$$8. f_1: \begin{cases} y_1(1) = y_1(2) = 0, \\ y_1(t) = y_1(t-1) \rightarrow y_1(t-2), t \geq 2, \end{cases}$$

$$f_2: \begin{cases} y_2(1) = 0, y_2(2) = 1, \\ y_2(t) = x_2(t) \rightarrow y_2(t-2), t \geq 2. \end{cases}$$

Домашнее задание

[1], глава IV, 2.9 (5), с. 155.

Построить канонические уравнения и приведённую диаграмму Мура о.-д. функции, получающейся из функции f введением обратной связи по переменным x_i, y_j :

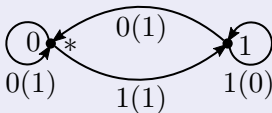
$$5. f: \begin{cases} y_1(t) = q(t-1), \\ y_2(t) = x_1(t) \oplus (x_2(t) \vee q(t-1)), \\ q(t) = q(t-1) \rightarrow x_1(t)x_2(t), \\ q(0) = 0. \end{cases}$$

Домашнее задание

[1], глава IV, 2.13 (6, 11), с. 167.

Построить канонические уравнения и приведённую диаграмму Мура о.-д. функции, получающейся из функции f введением обратной связи по переменным x_i, y_j :

6. Функция f задаётся диаграммой Мура:

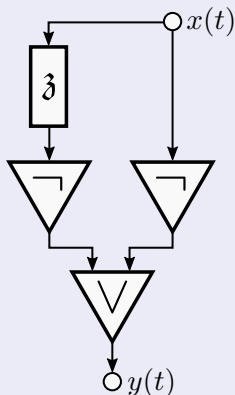


11. $f: \begin{cases} y(1) = 0, \\ y(t) = x(t-1) \vee y(t-1), t \geq 2, \end{cases}$

Домашнее задание

[1], глава IV, 2.14 (2), с. 168.

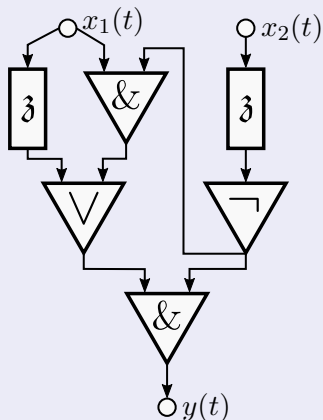
По схеме, реализующей функцию из $\hat{P}_{2,\text{од}}$, построить канонические уравнения, каноническую таблицу и диаграмму Мура:



Домашнее задание

[1], глава IV, 2.14 (5), с. 168.

По схеме, реализующей функцию из $\hat{P}_{2,од}$, построить канонические уравнения, каноническую таблицу и диаграмму Мура:



Домашнее задание

[1], глава IV, 2.17 (2, 5), с. 174.

Доказать полноту системы $\{f_1, f_2\}$ в $\hat{P}_{2,\text{од}}$ относительно операций суперпозиции и обратной связи.

2. $f_1: y(t) = x_1(t) \rightarrow \bar{x}_2(t), t \geq 1,$

$$f_2: \begin{cases} y(t) = (x_1(t) \rightarrow x_2(t)) \cdot q(t-1), \\ q(t) = x_1(t) \cdot x_2(t), \\ q(0) = 0. \end{cases}$$

5. $f_1: y(t) = \bar{x}_1(t) \cdot \bar{x}_2(t) \vee \bar{x}_1(t) \cdot \bar{x}_3(t) \vee \bar{x}_2(t) \cdot \bar{x}_3(t), t \geq 1,$

$$f_2: \begin{cases} y(t) = q(t-1), \\ q(t) = x_1(t) \oplus x_2(t), \\ q(0) = 0. \end{cases}$$

Список литературы

1. Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по дискретной математике. — М.: Физматлит, 2005. — 416 с.

Дополнительные главы дискретной математики и кибернетики

Семинар 6

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Машина Тьюринга

Конструкция машины

- **Машина Тьюринга** состоит из бесконечной в обе стороны ленты, разделённой на ячейки; считывающе-записывающей головки; множества состояний; и программы.
- В каждой ячейке ленты записан символ алфавита A . Один из символов этого алфавита считаем пустым символом. Только конечное число ячеек могут быть непустыми.
- Далее, если не оговорено иное, считаем $A = \{0, 1\}$ и 0 пустым символом.
- Множество состояний Q машины Тьюринга конечно и содержит начальное состояние q_1 и заключительное состояние q_0 .
- Программа машины — это отображение $f: A \times Q \rightarrow A \times \{L, R, S\} \times Q$.

Машина Тьюринга

Работа машины

- В каждый момент времени головка машины обозревает некоторую ячейку ленты и машина находится в одном из состояний Q .
- В начальный момент времени машина находится в состоянии q_1 .
- В каждый момент времени машина считывает символ a из обозреваемой головкой ячейки. По этому символу и текущему состоянию q_i машина с помощью своей программы получает набор $f(a, q_i) = (b, D, q_j) \in A \times \{R, S, L\} \times Q$.
- После этого машина записывает в текущую ячейку символ b , передвигает головку на другую ячейку ленты (L — на 1 влево, R — на 1 вправо, S — не передвигает) и переходит в состояние q_j .
- Машина останавливается при переходе в состояние q_0 . Если этого не происходит, машина работает бесконечно.

Машина Тьюринга

Запись программ машин Тьюринга

- Записываем программы машин Тьюринга в виде таблиц:

	1*	2	3	4
0	0R1	0L3		0R q_0
1	1R2	1R2	0L4	1L1

- Состояния (кроме q_0) обозначаем цифрами или другим удобным образом. Начальное состояние (если оно не обозначено q_1) отмечаем звёздочкой.
- Ячейки таблицы, которые не могут выполняться, оставляем пустыми. Для определённости считаем, что указанная там команда не меняет символ в ячейке, не двигает головку и переходит в q_0 .

Машина Тьюринга

[1], глава V, 1.8 (1), с. 185.

Построить машину Тьюринга, которая:

1. Начав работу с последней единицы массива из единиц, машина «сдвигает» его на одну ячейку влево, не изменяя «остального содержимого» ленты. Головка останавливается на первой единице «перенесённого» массива.

Машина Тьюринга

Решение [1], глава V, 1.8 (1), с. 185.

	1*	2
0		$1Sq_0$
1	$0L2$	$1L2$

- В состоянии 1 машина стирает последнюю единицу массива.
- В состоянии 2 она движется по массиву влево, записывает единицу слева от массива, и останавливается.

Машина Тьюринга

[1], глава V, 1.8 (3), с. 185.

Построить машину Тьюринга, которая:

3. Машина начинает работу с самой левой единицы на ленте и отыскивает единицу, примыкающую с левой стороны к первому слева массиву из трех нулей («окаймлённому» единицами). Головка останавливается на найденной единице (если такая есть). Содержимое ленты не меняется.

Машина Тьюринга

Решение [1], глава V, 1.8 (3), с. 185.

	1*	2	3	4	5	6
0	0R2	0R3	0R4	0R5	0R5	0L6
1	1R1	1R1	1R1	1L6	1R1	1S _{q₀}

- В состояниях 1, 2, 3 машина идёт вправо по единицам и отсчитывает 3 нуля подряд.
- В состоянии 4 машина отслеживает единицу после трёх нулей. Если единицы нет, она пропускает текущий массив нулей и возвращается в состояние 1.
- Если единица есть, то машина в состоянии 6 возвращается по трём нулям назад и останавливается на примыкающей к ним слева единице.

Машина Тьюринга

Конфигурация машины Тьюринга

- **Конфигурация** машины Тьюринга — это набор (v, q, w) , где
 - ▶ v — слово левее текущей позиции головки машины (начиная с первого непустого символа);
 - ▶ q — текущее состояние машины;
 - ▶ w — слова справа от текущей позиции головки машины (включая текущий обозреваемый символ, заканчивая последним непустым символом).
- Конфигурация машины Тьюринга полностью описывает текущее состояние ленты и головки машины.

Машина Тьюринга

[1], глава V, 1.4 (2), с. 184.

Построить машину Тьюринга, переводящую конфигурацию $q_1 0^n 1^n$ в $q_0 [01]^n$, $n \geq 1$.

Машина Тьюринга

Решение [1], глава V, 1.4 (2), с. 184.

Идём по ленте и пишем 01, пока не встретим 0 после единицы, затем возвращаемся.

	$[00]^*$	$[01]$	$[10]$	$[11]$	2	3
0	$0R[01]$	$1R[00]$	$0L2$	$0L2$	$0Rq_0$	$0L2$
1	$0R[11]$	$1R[10]$	$0R[11]$	$1R[10]$	$1L3$	

- Машина переключается между состояниями $[00]$, $[01]$, чтобы чередовать запись нулей и единиц. Когда машина встретит единицу, она начинает чередовать состояния $[10]$, $[11]$.
- Когда после этого машина встречает 0, она начинает двигать головку влево (по две ячейки за раз) с помощью состояний 2 и 3. Машина останавливается, когда в очередной раз не встретит единицу.

Вычислимые функции

Частичная функция

Частичная функция — это функция, которая определена только на части элементов данного множества.

- Далее рассматриваем частичные функции $f(\bar{x}): \mathbb{N}_0 \rightarrow \mathbb{N}_0$.

Примеры функций

- $\text{sg } x = \begin{cases} 0, & x = 0, \\ 1, & x > 0. \end{cases} \quad \overline{\text{sg}} x = \begin{cases} 1, & x = 0, \\ 0, & x > 0. \end{cases}$
- Функция $x - y$ не определена при $y > x$.
- $x \dot{-} y = \begin{cases} x - y, & x \geq y, \\ 0, & x < y. \end{cases}$
- Функция $x/2$ не определена при нечётных x .
- Функция $\lfloor x/2 \rfloor$ всюду определена.

Вычислимые функции

Основной код

- Кодлируем число $a \in \mathbb{N}_0$ в виде 1^{a+1} .
- Кодлируем набор $(a_1, \dots, a_k)$ из $\mathbb{N}_0^k$ в виде $1^{a_1+1}01^{a_2+1}0 \dots 01^{a_k+1}$.

Определение

Машина Тьюринга **вычисляет** функцию $f(\bar{x})$, если, начиная работу на коде набора $\bar{a}$ (остальные символы ленты — нули, головка на первом символе кода), машина работает так:

1. Если $f(\bar{a})$ определено, то через некоторое число тактов машина завершает вычисление, и конфигурация на ленте соответствует значению функции.
2. Если $f(\bar{a})$ не определено, то машина не завершает вычисление, либо завершает его, но получающаяся конфигурация не соответствует никакому одному числу.

Вычислимые функции

Определение

Машина Тьюринга **правильно вычисляет** функцию $f(\bar{x})$, если, начиная работу на коде набора $\bar{a}$ (остальные символы ленты — нули, головка на первом символе кода), машина работает так:

1. Если $f(\bar{a})$ определено, то через некоторое число тактов машина завершает вычисление, и конфигурация на ленте соответствует значению функции, а головка машины находится на первом непустом символе.
2. Если $f(\bar{a})$ не определено, то машина не завершает вычисление.

Вычислимые функции

Определение

Функция **вычислима**, если существует машина Тьюринга, которая её вычисляет.

- Любую вычислимую функцию можно правильно вычислить.

Тезис Чёрча-Тьюринга

Любая функция, значение которой можно получить путём выполнения строго определённой последовательности действий (алгоритма), является вычислимой.

Вычислимые функции

[1], глава V, 1.14 (1), с. 167.

Построить машину Тьюринга, вычисляющую функцию f .

1. $f(x) = \text{sg } x$.

Вычислимые функции

Решение [1], глава V, 1.14 (1), с. 167.

Если на ленте одна единица, ничего не делаем. Иначе стираем все единицы, кроме двух.

	1*	2	3
0		$0Lq_0$	$0Lq_0$
1	$1R2$	$1R3$	$0R3$

- В состоянии 1 пропускаем первую единицу.
- В состоянии 2 проверяем условие $x = 0$.
- В состоянии 3 стираем все единицы, кроме двух.

Вычислимые функции

[1], глава V, 1.14 (2), с. 167.

Построить машину Тьюринга, вычисляющую функцию f .

2. $f(x) = \overline{\text{sg}} x.$

Вычислимые функции

Решение [1], глава V, 1.14 (2), с. 167.

Если на ленте одна единица, добавляем ещё одну. Иначе стираем все единицы, кроме одной.

	1*	2	3
0		$1Lq_0$	$0Lq_0$
1	$1R2$	$0R3$	$0R3$

Вычислимые функции

[1], глава V, 1.14 (3), с. 167.

Построить машину Тьюринга, вычисляющую функцию f .

3. $f(x) = x \div 1.$

Вычислимые функции

Решение [1], глава V, 1.14 (3), с. 167.

Удаляем первую единицу. Если она была единственная, возвращаем её.

	1^*	2
0		$1Sq_0$
1	$0R2$	$1Sq_0$

Вычислимые функции

[1], глава V, 1.14 (4), с. 167.

Построить машину Тьюринга, вычисляющую функцию f .

$$4. f(x) = \lfloor \frac{1}{x} \rfloor = \begin{cases} 1, & x = 1, \\ 0, & x \geq 2, \\ \text{не определено,} & x = 0. \end{cases}$$

Вычислимые функции

Решение [1], глава V, 1.14 (4), с. 167.

Проверяем перечисленные случаи, стирая первые единицы.

Зацикливаемся при $x = 0$. Стираем всё и ставим единицу при $x > 2$.

Ставим две единицы при $x = 1$.

	1*	2	3	4	5
0		0S2	<u>1L5</u>	1Sq ₀	1Sq ₀
1	0R2	0R3	0R4	0R4	

Вычислимые функции

[1], глава V, 1.14 (9), с. 167.

Построить машину Тьюринга, вычисляющую функцию f .

9. $f(x, y) = x \div y$.

Вычислимые функции

Решение [1], глава V, 1.14 (9), с. 167.

Стираем x сначала, а y с конца. Проверяем, закончилось ли слово, перед каждым стиранием символа. Если первым закончился y , то добавляем к остатку x две единицы и завершаем работу. Иначе стираем до конца y и записываем единицу.

	1*	2	3	4	5	6	7	8
0	<u>0R7</u>	0R3	0L4	<u>1L8</u>	0L6	0R1	1Sq ₀	1Sq ₀
1	0R2	1R2	1R3	0L5	1L5	1L6	0R7	1L8

- В состояниях от 1 до 6 происходит цикл: проверка окончания x , стирание первого символа x , движение вправо, проверка окончания y , стирание последнего символа y , движение влево.
- В состоянии 7 стирается слово y и выписывается 1. В состоянии 8 головка возвращается в начало x и дописывается единица (другая единица дописана в состоянии 4).

Вычислимые функции

[1], глава V, 1.14 (10), с. 167.

Построить машину Тьюринга, вычисляющую функцию f .

10. $f(x, y) = x - y$.

Вычислимые функции

Решение [1], глава V, 1.14 (10), с. 167.

Стираем x сначала, а y с конца. Проверяем, закончилось ли слово, перед каждым стиранием символа. Если первым закончился y , то добавляем к остатку x две единицы и завершаем работу. Иначе: если от y ничего не осталось, пишем 1, иначе закидываемся.

	1*	2	3	4	5	6	7	8
0	<u>0R7</u>	0R3	0L4	<u>1L8</u>	0L6	0R1	1S q_0	1S q_0
1	0R2	1R2	1R3	0L5	1L5	1L6	1S7	1L8

- В состояниях от 1 до 6 происходит цикл: проверка окончания x , стирание первого символа x , движение вправо, проверка окончания y , стирание последнего символа y , движение влево.
- В состоянии 7 происходит закидывание, если y не пусто, и запись 1 иначе. В состоянии 8 головка возвращается в начало x и дописывается единица (другая единица дописана в состоянии 4).

Вычислимые функции

[1], глава V, 1.15 (2), с. 168.

Построить машину Тьюринга, правильно вычисляющую функцию f .

2. $f(x) = \left\lfloor \frac{1}{x-3} \right\rfloor$.

Вычислимые функции

Решение [1], глава V, 1.15 (2), с. 168.

- $f(x) = \begin{cases} 1, & x = 4, \\ 0, & x \geq 5, \\ \text{не определено,} & x \leq 3. \end{cases}$
- Двигаемся по слову влево, стирая его. Каждое состояние i проверяет условие $x = i$. В нужные моменты либо зацикливаемся, либо стираем остаток и пишем единицу, либо пишем две единицы.

	s^*	0	1	2	3	4	5	6
0		0S0	0S1	0S2	0S3	<u>1L6</u>	1S q_0	1S q_0
1	0R0	0R1	0R2	0R3	0S4	0R5	0R5	

Вычислимые функции

[1], глава V, 1.15 (7), с. 168.

Построить машину Тьюринга, правильно вычисляющую функцию f .

7. $f(x, y) = \left\lfloor \frac{2+x}{2-y} \right\rfloor.$

Вычислимые функции

Решение [1], глава V, 1.15 (7), с. 168.

- $$f(x, y) = \begin{cases} \lfloor \frac{2+x}{2} \rfloor, & y = 0, \\ 2 + x, & y = 1, \\ \text{не определено,} & y \geq 2. \end{cases}$$
- Проверяем значение y , одновременно добавляя две единицы к x . Если $y \geq 1$, то возвращаемся в начало x и останавливаемся, либо закливаемся.
- Если $y = 0$, то делим выписанное на ленте число на 2. Для этого формируем справа от него результат: стираем две единицы слева у $x + 2$, записываем одну единицу справа у результата. Когда $x + 2$ кончится, возвращаемся к началу вычисления.
- В зависимости от чётности x может потребоваться добавить единицу.

Вычислимые функции

Решение [1], глава V, 1.15 (7), с. 168 (продолжение).

		1	2	3	4	5	6
0		1R2		<u>0L7</u>	0L5	0L6	0Rq ₀
1		1R1	1R3	0R4	1S4		1L6

		7	8	9	10	11	12	13
0		<u>0L13</u>	<u>1L13</u>	0L10	1R11	0R12	0L7	0Rq ₀
1		0L8	0L9	1L9	1L10	1R11	1R12	1L13

- В состояниях от 7 до 12 происходит цикл: стирание двух единиц, движение влево, записывание одной единицы, движение вправо.
- Если единицы кончились в состоянии 7, то число $x + 3$ было чётно. Это значит, что выписано $\lfloor (x + 3)/2 \rfloor = \lfloor (x + 2)/2 \rfloor + 1$ единиц, то есть нужно только вернуться в начало.
- Если единицы кончились в состоянии 8, то число $x + 3$ было нечётно. Это значит, что выписано $\lfloor (x + 3)/2 \rfloor = \lfloor (x + 2)/2 \rfloor$ единиц, то есть нужно добавить ещё одну.

Домашнее задание

[1], глава V, 1.8 (2, 6), с. 185.

Построить машину Тьюринга, которая:

2. Начав двигаться вправо от какой-то «начальной» ячейки, головка «находит» первую при таком перемещении ячейку с единицей (если такая ячейка «встретится на пути») и, сделав «один шаг» вправо, останавливается на соседней ячейке. Если в «начальной» ячейке записана единица, то головка останавливается на соседней справа ячейке. Содержимое ленты не меняется.
6. При заданном $l \geq 1$ головка машины, двигаясь вправо от какой-либо пустой ячейки, находит первый при таком перемещении массив, содержащий не менее l единиц, стирает в нем первые l единиц и останавливается на самой правой из ячеек, в которых были стёртые единицы. Остальное содержимое ленты не меняется.

Домашнее задание

[1], глава V, 1.4 (4), с. 184.

Построить машину Тьюринга, переводящую конфигурацию $1^n q_1 0 1^m$ в конфигурацию $1^m q_0 0 1^n$, $m, n \geq 1$.

Домашнее задание

[1], глава V, 1.14 (5, 6, 7, 12), с. 167.

Построить машину Тьюринга, вычисляющую функцию f .

5. $f(x) = 3x$.

6. $f(x) = 2^{1-x}$.

7. $f(x) = \lfloor x/2 \rfloor$.

12. $f(x, y) = \frac{4-2x}{y^2}$.

[1], глава V, 1.15 (4, 6), с. 168.

Построить машину Тьюринга, правильно вычисляющую функцию f .

4. $f(x) = x - 5$.

6. $f(x, y) = \frac{x}{2} + y$.

Список литературы

1. Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по дискретной математике. — М.: Физматлит, 2005. — 416 с.

Дополнительные главы дискретной математики и кибернетики

Семинар 7

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Примитивно-рекурсивные функции

Простейшие функции

- $0(x) = 0$.
- $s(x) = x + 1$.
- $I = \{I_i^n(x_1, \dots, x_n) = x_i, i = \overline{1, n}, n \in \mathbb{N}\}$.

Операция суперпозиции

- Функция $f(\bar{x})$ получается из функций $g_0(y_1, \dots, y_m)$, $g_1(\bar{x}), \dots, g_m(\bar{x})$ с помощью **операции суперпозиции**, если

$$f(\bar{x}) = g_0(g_1(\bar{x}), \dots, g_m(\bar{x})).$$

- Если какое-то из значений $g_1(\bar{x}), \dots, g_m(\bar{x})$ не определено, то и $f(\bar{x})$ не определено.
- Например функция $(x - y) + 1$ не определена при $x = 0, y = 1$, а функция $(x + 1) - y$ равна 0 при $x = 0, y = 1$.

Примитивно-рекурсивные функции

Операция примитивной рекурсии

- $f(\bar{x}, y)$ получается из функций $g(\bar{x})$, $h(\bar{x}, y, z)$ с помощью операции примитивной рекурсии, если для любых $\bar{x}, y \in \mathbb{N}_0$

$$\begin{cases} f(\bar{x}, 0) = g(\bar{x}), \\ f(\bar{x}, y + 1) = h(\bar{x}, y, f(\bar{x}, y)). \end{cases}$$

- Если $f(\bar{x}, y)$ не определено, то и $f(\bar{x}, y + 1)$ не определено.

Определение

Класс примитивно-рекурсивных функций — это класс функций, получающихся из простейших функций 0 , $x + 1$, I при помощи применения операций суперпозиции и примитивной рекурсии.

- Все примитивно-рекурсивные функции всюду определены.

Примитивно-рекурсивные функции

[1], глава V, 2.1 (9), с. 198.

Используя константы, простейшие функции и операции суперпозиции и примитивной рекурсии, построить следующие функции.

9. $x^2 + 2y^2$.

Примитивно-рекурсивные функции

Решение [1], глава V, 2.1 (9), с. 198.

- Константы: $n = 0 + \underbrace{1 + \dots + 1}_n$ при $n \in \mathbb{N}$.

- Функции $x + y$ и xy :

$$\begin{cases} x + 0 = x = I_1^1(x), \\ x + (y + 1) = (x + y) + 1. \end{cases}$$

$$(h(x, y, z) = z + 1 = I_3^3(x, y, z) + 1)$$

$$\begin{cases} x \cdot 0 = 0, \\ x \cdot (y + 1) = (x \cdot y) + x. \end{cases}$$

$$(h(x, y, z) = z + x = I_3^3(x, y, z) + I_1^3(x, y, z))$$

- Далее применяем операцию суперпозиции:

$$x^2 + 2y^2 = x \cdot x + 2 \cdot y \cdot y.$$

Примитивно-рекурсивные функции

[1], глава V, 2.1 (10), с. 198.

Используя константы, простейшие функции и операции суперпозиции и примитивной рекурсии, построить следующие функции.

10. $x!$

Примитивно-рекурсивные функции

Решение [1], глава V, 2.1 (10), с. 198.

- Получаем $x!$ с помощью примитивной рекурсии:

$$\begin{cases} 0! = 1, \\ (x+1)! = (x!) \cdot x. \end{cases}$$
$$(h(x, y, z) = xz = I_1^3(x, y, z) \cdot I_3^3(x, y, z))$$

Примитивно-рекурсивные функции

[1], глава V, 2.1 (12), с. 198.

Используя константы, простейшие функции и операции суперпозиции и примитивной рекурсии, построить следующие функции.

12. $x \oplus y$.

Замечание

$x \oplus y$ — сложение по модулю 2. Эта функция выдаёт остаток от деления $x + y$ на 2.

Примитивно-рекурсивные функции

Решение [1], глава V, 2.1 (12), с. 198.

- Получаем функцию $\overline{\text{sg}} x$:

$$\begin{cases} \overline{\text{sg}} 0 = 1, \\ \overline{\text{sg}}(x + 1) = 0. \end{cases}$$

- Получаем функцию чётности:

$$\begin{cases} \sigma(0) = 0, \\ \sigma(x + 1) = \overline{\text{sg}} \sigma(x). \end{cases}$$

- $x \oplus y = \sigma(x + y)$.

Примитивно-рекурсивные функции

[1], глава V, 2.2 (1), с. 198.

Выяснить, какие функции получаются при применении операции примитивной рекурсии к $g(x)$, $h(x, y, z)$.

1. $g(x) = x$, $h(x, y, z) = x + z$.

Примитивно-рекурсивные функции

Решение [1], глава V, 2.2 (1), с. 198.

- Схема рекурсии:

$$\begin{cases} f(x, 0) = x, \\ f(x, y + 1) = x + f(x, y). \end{cases}$$

- $f(x, y) = x \cdot (y + 1).$

Примитивно-рекурсивные функции

[1], глава V, 2.2 (3), с. 198.

Выяснить, какие функции получаются при применении операции примитивной рекурсии к $g(x)$, $h(x, y, z)$.

3. $g(x) = 2^x$, $h(x, y, z) = 2^x \cdot z$.

Примитивно-рекурсивные функции

Решение [1], глава V, 2.2 (3), с. 198.

- Схема рекурсии:

$$\begin{cases} f(x, 0) = 2^x, \\ f(x, y + 1) = 2^x \cdot z. \end{cases}$$

- $f(x, y) = 2^{x \cdot (y+1)}.$

Примитивно-рекурсивные функции

[2], занятие 7.

Применить операцию примитивной рекурсии к частичным функциям $g(x) = 2x$ и $h(x, y, z) = z - 2$

Примитивно-рекурсивные функции

Решение [2], занятие 7.

- Схема рекурсии:

$$\begin{cases} f(x, 0) = 2x, \\ f(x, y + 1) = f(x, y) - 2. \end{cases}$$

- $f(x, y) = 2x - 2y$ (функция не определена при $y > x$).

Примитивно-рекурсивные функции

Построение некоторых функций

- Ранее получены константы, $x + y$, xy , $\overline{\text{sg}} x$, $x \oplus y$.
- $\text{sg } x = \overline{\text{sg}} \overline{\text{sg}} x$.
- Получение $x \dot{\div} 1$:

$$\begin{cases} 0 \dot{\div} 1 = 0, \\ (x + 1) \dot{\div} 1 = x. \end{cases}$$

- Получение $x \dot{\div} y$:

$$\begin{cases} x \dot{\div} 0 = x, \\ x \dot{\div} (y + 1) = (x \dot{\div} y) \dot{\div} 1. \end{cases}$$

- $|x - y| = (x \dot{\div} y) + (y \dot{\div} x)$ (эта функция всюду определена и не является суперпозицией $x - y$ и $|x|$).

Примитивно-рекурсивные функции

[1], глава V, 2.3 (9), с. 198.

Доказать примитивную рекурсивность функций, используя простейшие функции и функции $\text{sg } x$, $\overline{\text{sg}} x$, $x + y$, $x \dot{-} y$, $x \cdot y$, $x \oplus y$.

$$9. f(x) = \begin{cases} 0, & x \text{ чётно,} \\ x, & x \text{ нечётно.} \end{cases}$$

Примитивно-рекурсивные функции

Решение [1], глава V, 2.3 (9), с. 198.

- $f(x) = x \cdot (x \oplus 0).$

Примитивно-рекурсивные функции

[1], глава V, 2.3 (10), с. 198.

Доказать примитивную рекурсивность функций, используя простейшие функции и функции $\text{sg } x$, $\overline{\text{sg}} x$, $x + y$, $x \dot{-} y$, $x \cdot y$, $x \oplus y$.

10. Проверка делимости на m :

$$f_m(x) = \begin{cases} 1, & x = lm, \text{ где } l \in \mathbb{N}_0, \\ 0, & \text{иначе,} \end{cases} \quad m \geq 2.$$

Замечание

Считаем, что m — параметр, а не аргумент функции.

Примитивно-рекурсивные функции

Решение [1], глава V, 2.3 (10), с. 198.

- $s_m(x) = (x + 1) \cdot \overline{\text{sg}}(x \dot{\div} (m \dot{\div} 2))$ — функция, вычисляющая $x + 1$ по модулю m для $x < m$.
- Получаем функцию остатка от деления на m $\text{rm}(x, m)$:
$$\begin{cases} \text{rm}(0, m) = 0, \\ \text{rm}(x + 1, m) = s_m(\text{rm}(x, m)) \end{cases}$$
- $f_m(x) = \overline{\text{sg}} \text{rm}(x, m)$.

Примитивно-рекурсивные функции

[1], глава V, 2.3 (5), с. 198.

Доказать примитивную рекурсивность функций, используя простейшие функции и функции $\text{sg } x$, $\overline{\text{sg}} x$, $x + y$, $x \dot{-} y$, $x \cdot y$, $x \oplus y$.

5. $f(x, y) = \lfloor x/y \rfloor$ (0 при $y = 0$).

Примитивно-рекурсивные функции

Решение [1], глава V, 2.3 (5), с. 198.

- $\lfloor x/y \rfloor = \sum_{z=0}^x z \cdot \text{sg}((x + 1 \div zy) \cdot ((z + 1)y \div x)).$
- В указанной сумме слагаемое будет ненулевым только если $x + 1 > zy$ и $(z + 1)y > x$, то есть если $zy \leq x$, а $(z + 1)y > x$.
- Заметим, что $(z = \lfloor x/y \rfloor) \equiv (zy \leq x) \ \& \ ((z + 1)y > x).$
- Поэтому в сумме только одно ненулевое слагаемое: $\lfloor x/y \rfloor$.
- Верхний предел суммы можно выбрать любым (только не меньшим $\lfloor x/y \rfloor$).
- То же самое через рекурсию:
$$\begin{cases} u(x, y, 0) = 0, \\ u(x, y, z + 1) = u(x, y, z) + z \cdot \text{sg}((x + 1 \div zy) \cdot ((z + 1)y \div x)). \end{cases}$$
$$\lfloor x/y \rfloor = u(x, y, x).$$

Примитивно-рекурсивные функции

- Предикат натурального аргумента — это отображение: $\mathbb{N}_0 \rightarrow \{\text{false}, \text{true}\}$. Мы будем рассматривать предикат как функцию: $\mathbb{N}_0 \rightarrow \{0, 1\}$.
- Предикаты $\rho_1(\bar{x})$ и $\rho_2(\bar{x})$ называются несовместными, если $\rho_1(\bar{x}) \cdot \rho_2(\bar{x}) \equiv 0$.

Разбор случаев по предикатам

Функция $f(\bar{x})$ получается из функций $g_1(\bar{x}), \dots, g_{m+1}(\bar{x})$ и попарно несовместных предикатов $\rho_1(\bar{x}), \dots, \rho_m(\bar{x})$ с помощью операции разбора случаев по предикатам, если

$$f(\bar{x}) = \begin{cases} g_1(\bar{x}), & \text{если } \rho_1(\bar{x}) = 1, \\ \dots & \\ g_m(\bar{x}), & \text{если } \rho_m(\bar{x}) = 1, \\ g_{m+1}(\bar{x}) & \text{иначе.} \end{cases}$$

Примитивно-рекурсивные функции

Разбор случаев по предикатам

- Пусть $g_1(\bar{x}), \dots, g_{m+1}(\bar{x})$ — примитивно-рекурсивные функции, а $\rho_1(\bar{x}), \dots, \rho_m(\bar{x})$ — попарно несовместные примитивно-рекурсивные предикаты и

$$f(\bar{x}) = \begin{cases} g_1(\bar{x}), & \text{если } \rho_1(\bar{x}) = 1, \\ \dots \\ g_m(\bar{x}), & \text{если } \rho_m(\bar{x}) = 1, \\ g_{m+1}(\bar{x}) & \text{иначе.} \end{cases}$$

- Тогда $f(\bar{x})$ примитивно-рекурсивна:

$$f(\bar{x}) = g_1(\bar{x})\rho_1(\bar{x}) + \dots + g_m(\bar{x})\rho_m(\bar{x}) + g_{m+1}(\bar{x})\overline{\text{sg}}(\rho_1(\bar{x}) + \dots + \rho_m(\bar{x})).$$

Примитивно-рекурсивные функции

[1], глава V, 2.4 (1), с. 199.

1. Доказать, что если $g(x)$ примитивно рекурсивна, то $f(x)$, отличающаяся от $g(x)$ только в конечном числе точек, тоже примитивно рекурсивна.

Примитивно-рекурсивные функции

Решение [1], глава V, 2.4 (1), с. 199.

- Пусть $f(\bar{x})$ отличается от $g(\bar{x})$ в точках $\bar{a}_1, \dots, \bar{a}_k$, при этом $f(\bar{a}_i) = b_i$, $i = \overline{1, k}$.
- Тогда
$$f(\bar{x}) = \begin{cases} b_1, & \text{если } \bar{x} = \bar{a}_1, \\ \dots & \\ b_k, & \text{если } \bar{x} = \bar{a}_k, \\ g(\bar{x}) & \text{иначе.} \end{cases}$$
- Пусть $\bar{x} = (x_1, \dots, x_n)$ и $\bar{a} = (a_1, \dots, a_n)$. Тогда предикат $(\bar{x} = \bar{a})$ выражается функцией $\overline{\text{sg}}(|x_1 - a_1| + \dots + |x_n - a_n|)$.
- Таким образом, функция $f(\bar{x})$ примитивно-рекурсивна.

Примитивно-рекурсивные функции

[1], глава V, 2.4 (2), с. 199.

2. Пусть $g_1(x), g_2(x)$ — примитивно рекурсивны. Доказать, что

$$f(x) = \begin{cases} g_1(x), & a \leq x \leq b, \\ g_2(x), & \text{иначе} \end{cases} \quad (0 \leq a \leq b) \text{ примитивно рекурсивна.}$$

Примитивно-рекурсивные функции

Решение [1], глава V, 2.4 (2), с. 199.

- $f(x) = \begin{cases} g_1(x), & a \leq x \leq b, \\ g_2(x), & \text{иначе.} \end{cases}$
- Предикат $a \leq x \leq b$ выражается функцией

$$\text{sg}(((x+1) \div a) + ((b+1) \div x)).$$

- Таким образом, функция $f(\bar{x})$ примитивно-рекурсивна.

Примитивно-рекурсивные функции

[1], глава V, 2.4 (5), с. 199.

5. Пусть $g(\bar{x}, z)$ примитивно рекурсивна. Доказать примитивную рекурсивность функций:

$$f_1(\bar{x}, y) = \sum_{z=0}^y g(\bar{x}, z), \quad f_2(\bar{x}, y) = \prod_{z=0}^y g(\bar{x}, z)$$

Примитивно-рекурсивные функции

Решение [1], глава V, 2.4 (5), с. 199.

- $f_1(\bar{x}, y) = \sum_{z=0}^y g(\bar{x}, z)$ выражается схемой:

$$\begin{cases} f_1(\bar{x}, 0) = g(\bar{x}, 0), \\ f_1(\bar{x}, y + 1) = f_1(\bar{x}, y) + g(\bar{x}, y + 1). \end{cases}$$

- $f_2(\bar{x}, y) = \prod_{z=0}^y g(\bar{x}, z)$ выражается схемой:

$$\begin{cases} f_2(\bar{x}, 0) = g(\bar{x}, 0), \\ f_2(\bar{x}, y + 1) = f_2(\bar{x}, y) \cdot g(\bar{x}, y + 1). \end{cases}$$

Примитивно-рекурсивные функции

[1], глава V, 2.4 (76), с. 199.

76. Доказать примитивную рекурсивность $f_2(x) = \lfloor \log_a(x + 1) \rfloor$,
 $a \geq 2$.

Замечание

Считаем a параметром, а не аргументом функции.

Примитивно-рекурсивные функции

Решение [1], глава V, 2.4 (76), с. 199.

- Функция a^z :

$$\begin{cases} a^0 = 1, \\ a^{z+1} = a^z \cdot a. \end{cases}$$

- $(z = \lfloor \log_a(x+1) \rfloor) \equiv (a^z \leq x+1 < a^{z+1})$.
- $\lfloor \log_a(x+1) \rfloor = \sum_{z=0}^x z \cdot \text{sg}((x+2 \div a^z) \cdot (a^{z+1} \div (x+1)))$.
- Верхний предел суммы можно выбрать любой не меньший $\lfloor \log_a(x+1) \rfloor$.

Домашнее задание

[1], глава V, 2.1 (2, 4), с. 198.

Используя константы и простейшие функции, и операции суперпозиции и примитивной рекурсии, построить следующие функции.

2. $\overline{sg} \, x$.

4. nx .

Домашнее задание

[1], глава V, 2.4 (3, 7a), с. 198.

3. Доказать, что если $g(y), \varphi_1(y), \varphi_2(y), \varphi_3(y)$ примитивно рекурсивны, то примитивно рекурсивна и

$$f(x, y) = \begin{cases} \varphi_1(x), & g(y) \leq a, \\ \varphi_2(x), & a < g(y) \leq b, \\ \varphi_3(x), & g(y) > b. \end{cases}$$

- 7a. Доказать примитивную рекурсивность

$$f_1(x) = \begin{cases} 1, & x = a^l, \text{ где } l \in \mathbb{N}_0, \\ 0 & \text{иначе.} \end{cases}$$

Домашнее задание

[1], глава V, 2.3 (7, 8), с. 198.

Доказать примитивную рекурсивность функций, используя простейшие функции и функции $\text{sg } x$, $\overline{\text{sg}} x$, $x + y$, $x \dot{-} y$, $x \cdot y$, $x \oplus y$.

$$7. f_{a,i}(x) = \begin{cases} a, & x = i, \\ x, & \text{иначе,} \end{cases} \quad a, i \in \mathbb{N}_0.$$

$$8. f(x) = \begin{cases} a_x, & x \leq m, \\ c, & \text{иначе,} \end{cases} \quad a_0, \dots, a_m, c \in \mathbb{N}_0.$$

Список литературы

1. Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по дискретной математике. — М.: Физматлит, 2005. — 416 с.
2. Марченков С. С. План семинарских занятий по курсу «Дополнительные главы дискретной математики», часть 2.

Дополнительные главы дискретной математики и кибернетики

Семинар 8

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Примитивно-рекурсивные функции

Операции суммирования и произведения

- $f(\bar{x}, y)$ получается из $g(\bar{x}, z)$ с помощью операции ограниченного суммирования, если

$$f(\bar{x}, y) = \sum_{z=0}^y g(\bar{x}, z).$$

- $f(\bar{x}, y)$ получается из $g(\bar{x}, z)$ с помощью операции ограниченного умножения, если

$$f(\bar{x}, y) = \prod_{z=0}^y g(\bar{x}, z)$$

- Операции ограниченного суммирования и умножения не выводят из класса примитивно-рекурсивных функций.

Примитивно-рекурсивные функции

[2], занятие 8.

Используя операции $\prod_{i \leq x}$ и $\sum_{i \leq x}$ доказать примитивную рекурсивность функций:

- $\lfloor x/y \rfloor$ ($\lfloor x/0 \rfloor = 0$).

Примитивно-рекурсивные функции

Решение [2], занятие 8.

- Заметим, что $(z = \lfloor x/y \rfloor) \equiv (zy \leq x) \ \& \ ((z+1)y > x)$.
- Произведём перебор с помощью суммирования:

$$\lfloor x/y \rfloor = \sum_{z=0}^x z \cdot \text{sg}((x+1 \dot{-} zy) \cdot ((z+1)y \dot{-} x)).$$

- В сумме только одно из слагаемых является ненулевым. Оно выделяется с помощью предиката $(z = \lfloor x/y \rfloor)$.

Примитивно-рекурсивные функции

[2], занятие 8.

Используя операции $\prod_{i \leq x}$ и $\sum_{i \leq x}$ доказать примитивную рекурсивность функций:

- $\lfloor \sqrt{x} \rfloor$.

Примитивно-рекурсивные функции

Решение [2], занятие 8.

- $x^2 = x \cdot x$.
- $(z = \lfloor \sqrt{x} \rfloor) \equiv (z^2 \leq x) \ \& \ ((z+1)^2 > x)$.
- Произведём перебор с помощью суммирования:

$$\lfloor \sqrt{x} \rfloor = \sum_{z=0}^x z \cdot \text{sg}((x+1 \dot{-} z^2) \cdot ((z+1)^2 \dot{-} x)).$$

Примитивно-рекурсивные функции

[2], занятие 8.

Используя операции $\prod_{i \leq x}$ и $\sum_{i \leq x}$ доказать примитивную рекурсивность функций:

- $\lfloor \log_a x \rfloor$ ($\lfloor \log_a 0 \rfloor = 0$).

Примитивно-рекурсивные функции

Решение [2], занятие 8.

- Функция a^z :

$$\begin{cases} a^0 = 1, \\ a^{z+1} = a^z \cdot a. \end{cases}$$

- $(z = \lfloor \log_a x \rfloor) \equiv (a^z \leq x < a^{z+1})$.
- Произведём перебор с помощью суммирования:

$$\lfloor \log_a x \rfloor = \sum_{z=0}^x z \cdot \text{sg}((x + 1 \div a^z) \cdot (a^{z+1} \div x)).$$

Примитивно-рекурсивные функции

[2], занятие 8.

Используя операции $\prod_{i \leq x}$ и $\sum_{i \leq x}$ доказать примитивную рекурсивность функций:

- Число делителей x .

Примитивно-рекурсивные функции

Решение [2], занятие 8.

- $\text{rm}(x, y)$ — остаток от деления x на y ($\text{rm}(x, 0) = 0$).
- $\text{rm}(x, y) = (\text{sg } y) \cdot (x \dot{-} y \lfloor x/y \rfloor)$.
- Искомая функция равна

$$\left(\sum_{z=0}^x \overline{\text{sg}} \text{rm}(x, z) \right) \dot{-} 1.$$

Примитивно-рекурсивные функции

[2], занятие 8.

Используя операции $\prod_{i \leq x}$ и $\sum_{i \leq x}$ доказать примитивную рекурсивность функций:

- Число решений полиномиального уравнения в заданном промежутке.

Замечание

Имеем в виду число целочисленных решений на отрезке $[a, b]$, $b \geq a \geq 0$.

Примитивно-рекурсивные функции

Решение [2], занятие 8.

- Пусть имеется уравнение $P(x) = 0$, где P — полином с целыми коэффициентами.
- Перенесём все отрицательные коэффициенты в левую часть, получим уравнение $Q_1(x) = Q_2(x)$, где Q_1, Q_2 — полиномы с натуральными коэффициентами.
- Полиномы Q_1, Q_2 можно получить суперпозицией констант и функций $x + y$, xy .
- Подсчитываем число решений на отрезке $[a, b]$:

$$\left(\sum_{x=0}^b \overline{\text{sg}} |Q_1(x) - Q_2(x)| \right) - (\text{sg } a) \left(\sum_{x=0}^{a-1} \overline{\text{sg}} |Q_1(x) - Q_2(x)| \right).$$

Операция минимизации

Операция минимизации

- $f(\bar{x}, y)$ получается из $g(\bar{x}, z)$ путём применения **операции минимизации** по переменной z , если $f(\bar{x}, y)$ — минимальное значение z , при котором $g(\bar{x}, z) = y$.
- Если такого z нет, то $f(\bar{x}, y)$ не определено. Если для некоторого z значение $g(\bar{x}, z)$ не определено и $g(\bar{x}, v) \neq x$ при $v < z$, то $f(\bar{x}, y)$ не определено.
- Обозначаем $f(\bar{x}, y) = (\mu z)(g(\bar{x}, z) = y)$.
- Иными словами, значение $f(\bar{x}, y)$ определяется так:
 - ▶ Перебираем $z = 0, 1, 2, \dots$
 - ▶ На каждом шаге, если значение $g(\bar{x}, z)$ не определено, то завершаем процесс: результат не определён.
 - ▶ Если значение $g(\bar{x}, z)$ определено и $g(\bar{x}, z) = y$, то завершаем процесс и выдаём z в качестве результата.
 - ▶ В противном случае продолжаем перебор.

Операция минимизации

[1], глава V, 2.5 (1), с. 200.

Применить операцию минимизации к f по переменной x_i , представив результат в «аналитической» форме.

1. $f(x_1) = 3, i = 1.$

Операция минимизации

Решение [1], глава V, 2.5 (1), с. 200.

- $$h(y) = (\mu z)(3 = y) = \begin{cases} 0, & y = 3, \\ \text{не определено}, & y \neq 3. \end{cases}$$
- $$h(y) = \overline{\text{sg}} |y - 3| - 1.$$

Операция минимизации

[1], глава V, 2.5 (2), с. 200.

Применить операцию минимизации к f по переменной x_i , представив результат в «аналитической» форме.

2. $f(x_1) = x_1 + 2, i = 1.$

Операция минимизации

Решение [1], глава V, 2.5 (2), с. 200.

- $$h(y) = (\mu z)(z + 2 = y) = \begin{cases} y - 2, & y \geq 2, \\ \text{не определено}, & y < 2. \end{cases}$$
- $h(y) = y - 2.$

Операция минимизации

[1], глава V, 2.5 (3), с. 200.

Применить операцию минимизации к f по переменной x_i , представив результат в «аналитической» форме.

3. $f(x_1) = x_1 \div 2, i = 1.$

Операция минимизации

Решение [1], глава V, 2.5 (3), с. 200.

- $h(y) = (\mu z)(z \dot{\div} 2 = y) = \begin{cases} 0, & y = 0, \\ y + 2, & y > 0. \end{cases}$
- $h(y) = (y + 2) \cdot \text{sg } y$

Операция минимизации

[1], глава V, 2.5 (7), с. 200.

Применить операцию минимизации к f по переменной x_i , представив результат в «аналитической» форме.

7. $f(x_1) = \lfloor x_2/2 \rfloor, i = 1.$

Операция минимизации

Решение [1], глава V, 2.5 (7), с. 200.

- $h(y) = (\mu z)(\lfloor z/2 \rfloor = y) = 2y.$

Операция минимизации

[1], глава V, 2.5 (11), с. 200.

Применить операцию минимизации к f по переменной x_i , представив результат в «аналитической» форме.

11. $f(x_1, x_2) = x_1 \div x_2, i = 1, 2.$

Операция минимизации

Решение [1], глава V, 2.5 (11), с. 200.

$$\bullet i = 1: h(x_2, y) = (\mu z)(z \dot{-} x_2 = y) = \begin{cases} 0, & y = 0, \\ y + x_2, & y > 0. \end{cases}$$

$$h(x_2, y) = (y + x_2) \cdot \text{sg } y.$$

$$\bullet i = 2: h(x_1, y) = (\mu z)(x_1 \dot{-} z = y) = \begin{cases} x_1 - y, & y \leq x_1, \\ \text{не определено}, & y > x_1. \end{cases}$$

$$h(x_1, y) = y - x_1.$$

Операция минимизации

[1], глава V, 2.7 (2), с. 200.

Доказать примитивную рекурсивность функций.

2. Результат минимизации по x функции $\lfloor x/2 \rfloor \div 1$.

Операция минимизации

Решение [1], глава V, 2.7 (2), с. 200.

- $f(y) = (\mu z)(\lfloor z/2 \rfloor \div 1 = y) = \begin{cases} 0, & y = 0, \\ 2y + 2, & y > 0. \end{cases}$
- $f(y) = (2y + 2) \cdot \text{sg } y$. Это суперпозиция примитивно-рекурсивных функций.

Операция минимизации

[1], глава V, 2.7 (6), с. 200.

Доказать примитивную рекурсивность функций.

6. Результат $f(y)$ минимизации по x функции $g(x) = x \dot{\div} \lfloor \sqrt{x} \rfloor^2$.

Операция минимизации

Решение [1], глава V, 2.7 (6), с. 200.

- $f(y) = (\mu x)(x \dot{-} \lfloor \sqrt{x} \rfloor^2 = y)$.
- Функция $x \dot{-} \lfloor \sqrt{x} \rfloor^2$ примитивно-рекурсивна как суперпозиция примитивно-рекурсивных функций.
- Будем вычислять результат минимизации с помощью перебора. Для этого требуется определить верхнюю границу перебора.
- Для нахождения x , при котором $x \dot{-} \lfloor \sqrt{x} \rfloor^2 = y$, достаточно проверить все x вплоть до $y^2 + y$ (это число удовлетворяет указанному условию, хотя, возможно, и не минимально).
- Получаем (слагаемое при $x = 0$ всегда равно 0):

$$f(y) = \sum_{x=0}^{y^2+y} x \cdot \overline{\text{sg}} |(x \dot{-} \lfloor \sqrt{x} \rfloor^2) - y| \cdot \prod_{z=0}^{x \dot{-} 1} \text{sg} |(z \dot{-} \lfloor \sqrt{z} \rfloor^2) - y|.$$

Домашнее задание

[2], занятие 8.

Используя операции $\prod_{i \leq x}$ и $\sum_{i \leq x}$ доказать примитивную рекурсивность функций:

- $p(x) = \begin{cases} 1, & x \text{ — простое число,} \\ 0, & \text{иначе.} \end{cases}$
- $f_1(x)$ — количество чисел вида y^y на отрезке $[0, x]$.
- $f_2(x)$ — количество чисел вида $2^y \cdot 3^z$ на отрезке $[0, x]$.

Домашнее задание

[1], глава V, 2.5 (4, 10, 13), с. 200.

Применить операцию минимизации к f по переменной x_i , представив результат в «аналитической» форме.

4. $f(x_1) = x_1 - 2, i = 1.$

10. $f(x_1, x_2) = I_1^2(x_1, x_2), i = 2.$

13. $f(x_1, x_2) = x_1 \div 1/x_2, i = 1, 2.$

[1], глава V, 2.7 (3, 5), с. 200.

Доказать примитивную рекурсивность функций.

3. Результат минимизации по x функции $x \div \lfloor x/2 \rfloor$.

5. Результат $f(y)$ минимизации по x функции $g(x) = \lfloor \sqrt[3]{x^2} \rfloor$.

Список литературы

1. Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по дискретной математике. — М.: Физматлит, 2005. — 416 с.
2. Марченков С. С. План семинарских занятий по курсу «Дополнительные главы дискретной математики», часть 2.

Дополнительные главы дискретной математики и кибернетики

Семинар 9

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Операция минимизации

[1], глава V, 2.8 (1), с. 200.

Доказать, что однократное применение операции минимизации к всюду определённой функции приводит к функции, определённой хотя бы в одной точке.

Операция минимизации

Решение [1], глава V, 2.8 (1), с. 200.

- Пусть $g(\bar{x}, y) = (\mu z)(f(\bar{x}, z) = y)$, f всюду определена.
- Фиксируем любой набор $\bar{x}$.
- Поскольку $f(\bar{x}, 0)$ определена, будет верно $g(\bar{x}, f(\bar{x}, 0)) = 0$.
- Таким образом, g определена в указанной точке.

Операция минимизации

[1], глава V, 2.8 (2), с. 200.

Привести пример одноместной примитивно рекурсивной функции, из которой двукратным применением операции минимизации можно получить нигде не определённую функцию.

Операция минимизации

Решение [1], глава V, 2.8 (2), с. 200.

- Рассмотрим $f(x) \equiv 1$.
- $g_1(y) = (\mu x)(1 = y)$, $g_1(y)$ определена только в единице.
- Тогда $g_2(x) = (\mu y)(g_1(y) = x)$ нигде не определена (так как $g_1(0)$ не определено).

Операция минимизации

[1], глава V, 2.8 (5), с. 200.

Доказать, что если функция $g(x, y)$ не является всюду определённой, то таким же свойством обладает каждая из функций $f_x(x, y) = (\mu z)(g(z, y) = x)$, $f_y(x, y) = (\mu z)(g(x, z) = y)$.

Операция минимизации

Решение [1], глава V, 2.8 (5), с. 200.

- Поскольку порядок переменных в условии не имеет значения, достаточно рассмотреть функцию $f_x(x, y) = (\mu z)(g(z, y) = x)$.
- Пусть значение $g(x_0, y_0)$ не определено.
- Выберем такое число n , которое не совпадает с $g(0, y_0), g(1, y_0), \dots, g(x_0 - 1, y_0)$ (это возможно, так как последовательность конечна).
- Тогда $f_x(n, y_0) = (\mu z)(g(z, y_0) = n)$ не определено.

Частично рекурсивные функции

- Селекторные функции $I = \{I_i^n(x_1, \dots, x_n) = x_i, i = \overline{1, n}, n \in \mathbb{N}\}$.
- Суперпозиция $f(\bar{x}) = g_0(g_1(\bar{x}), \dots, g_m(\bar{x}))$.
- Прimitивная рекурсия

$$\begin{cases} f(\bar{x}, 0) = g(\bar{x}), \\ f(\bar{x}, y + 1) = h(\bar{x}, y, f(\bar{x}, y)), y \in \mathbb{N}_0. \end{cases}$$

- Минимизация $f(\bar{x}, y) = (\mu z)(g(\bar{x}, z) = y)$.

Определение

Класс частично рекурсивных функций — это класс функций, получающихся из простейших функций 0 , $x + 1$, I при помощи применения операций суперпозиции, примитивной рекурсии и минимизации.

Частично рекурсивные функции

Теорема

Класс частично рекурсивных функций совпадает с классом вычислимых функций.

Определение

Класс общерекурсивных функций — это класс всюду определённых частично рекурсивных функций.

- Существуют общерекурсивные, но не примитивно рекурсивные функции.

Частично рекурсивные функции

Аналог функции Аккермана

$$A_1(x, i) = \begin{cases} 2 + x, & i = 0, \\ 2x, & i = 1, \\ 2^x, & i = 2, \\ 2^{\underbrace{2 \dots 2^2}_x}, & i = 3, \\ \dots \end{cases}$$

- При $c \geq 1$ функция $A_1(x, c + 1)$ получается в результате x -кратного применения $A_1(x, c)$ к 1. Например, $2^x = \underbrace{2 \cdot \dots \cdot 2}_x \cdot 1$.
- Поэтому для вычисления значения $A_1(x, i)$ достаточно вычислить несколько значений
 $z_1 = A_1(1, i - 1), z_2 = A_1(z_1, i - 1), \dots, z_x = A_1(z_{x-1}, i - 1)$.

Частично рекурсивные функции

Аналог функции Аккермана

- Продолжая рассуждения аналогично, можно свести вычисление $A_1(x, i)$ к вычислению $A_1(z_1, 1), \dots, A_1(z_k, 1)$ для некоторых чисел $z_1, \dots, z_k$.
- Таким образом, функция $A_1(x, i)$ вычислима; а значит, частично-рекурсивна. Поскольку она всюду определена, она общерекурсивна.
- С другой стороны, для получения функции $A_1(x, c)$ необходимо применить не менее c операций примитивной рекурсии. Например:
 - ▶ Без рекурсии можно получить лишь функции, ограниченные функциями $x + c$.
 - ▶ С помощью одной рекурсии можно получить функции, ограниченные линейными функциями.
 - ▶ С помощью двух рекурсий можно получить функции, ограниченные степенными башнями не зависящей от x высоты.

Частично рекурсивные функции

Аналог функции Аккермана

- Но если бы $A_1(x, i)$ была примитивно-рекурсивной, её можно было бы получить с использованием конечного числа рекурсий — c .
- Тогда бы и $A_1(x, c + 1)$ можно было бы получить с помощью c рекурсий, что невозможно.
- Таким образом, функция $A_1(x, i)$ не примитивно-рекурсивна. Эта функция растёт быстрее, чем любая примитивно-рекурсивная функция.

Частично рекурсивные функции

[2], занятие 9 (пункт 2).

Доказать частичную рекурсивность функций.

2. $f^{-1}(x)$, где f — общерекурсивная перестановка (биективная функция) на $\mathbb{N}_0$.

Частично рекурсивные функции

Решение [2], занятие 9 (пункт 2).

- $f^{-1}(x) = (\mu z)(f(z) = x).$

Частично рекурсивные функции

[2], занятие 9 (пункт 3).

Доказать частичную рекурсивность функций.

$$3. f(x) = \begin{cases} x, & x \in \{a_1, \dots, a_m\}, \\ \text{не определено} & \text{иначе,} \end{cases} \quad a_1, \dots, a_m \in \mathbb{N}_0.$$

Частично рекурсивные функции

Решение [2], занятие 9 (пункт 3).

- $\rho(x) = \overline{\text{sg}} |x - a_1| + \dots + \overline{\text{sg}} |x - a_m|$ — предикат $(x \in \{a_1, \dots, a_m\})$.
- $f(x) = x + 0(\rho(x) - 1)$.

Частично рекурсивные функции

[2], занятие 9 (пункт 5).

Доказать частичную рекурсивность функций.

5.
$$f(x) = \begin{cases} x, & f_1(x) \geq f_2(x), \\ \text{не определено} & \text{иначе,} \end{cases}$$

 $f_1(x), f_2(x)$ — примитивно-рекурсивные функции.

Частично рекурсивные функции

Решение [2], занятие 9 (пункт 5).

- $f(x) = x + 0(f_1(x) - f_2(x)).$

Частично рекурсивные функции

[2], занятие 9 (пункт 7).

Доказать частичную рекурсивность функций.

7. x/y .

Частично рекурсивные функции

Решение [2], занятие 9 (пункт 7).

- Функция $g(x, y) = (\mu z)(y \cdot z = x)$ совпадает с x/y во всех точках, кроме $(0, 0)$.
- $x/y = (\mu z)(y \cdot z = x) + 0(\overline{\text{sg}} y - 1)$.

Частично рекурсивные функции

[2], занятие 9 (пункт 8).

Доказать частичную рекурсивность функций.

8. $\sqrt{x}$.

Частично рекурсивные функции

Решение [2], занятие 9 (пункт 8).

- $\sqrt{x} = (\mu z)(z^2 = x).$

Частично рекурсивные функции

[2], занятие 9 (пункт 10).

Доказать частичную рекурсивность функций.

$$10. f(x) = \begin{cases} x, & \text{в последовательности } g \text{ есть две единицы} \\ & \text{на расстоянии } x + 1, \\ \text{не опр.} & \text{иначе,} \end{cases}$$

$g(z)$ — примитивно-рекурсивная функция, принимающая значения 0, 1 (двоичная последовательность).

Частично рекурсивные функции

Решение [2], занятие 9 (пункт 10).

- Рассмотрим функцию $g(x) = (\mu z)((g(z) \cdot g(z + x + 1)) = 1)$. Эта функция ищет две единицы на расстоянии $x + 1$ и выдаёт номер первой из этих единиц в первой найденной паре. Если таких единиц нет, выдаётся неопределённое значение.
- Минимизация выше имеет нестандартный вид. Её можно реализовать обычной минимизацией так:
$$h(x, y) = (\mu z)((g(z) \cdot g(z + x + 1)) = y), \quad g(x) = h(x, 1).$$
- $f(x) = x + 0(g(x))$.

Домашнее задание

[1], глава V, 2.8 (3), с. 200.

Сформулировать условие, необходимое и достаточное для того, чтобы функция

$$f(y) = (\mu x)(g(x) = y)$$

была нигде не определённой.

Домашнее задание

[2], занятие 9 (пункты 1, 4, 6, 9, 11).

Доказать частичную рекурсивность функций.

$$1. \quad f(x) = \begin{cases} x, & x \text{ входит в пересечение областей значений} \\ & \text{частично-рекурсивных функций } g_1(z), g_2(z), \\ \text{не опр.} & \text{иначе.} \end{cases}$$

$$4. \quad f(x) = \begin{cases} 1, & x \text{ входит в область значений функции } g, \\ \text{не опр.} & \text{иначе,} \end{cases}$$

$g(z)$ — примитивно-рекурсивная функция, например $z^2, 2^z$.

$$6. \quad x - y.$$

$$9. \quad \log_2 x.$$

$$11. \quad f(x) = \begin{cases} x, & \text{в посл-ти } g \text{ есть } x + 1 \text{ идущих подряд единиц,} \\ \text{не опр.} & \text{иначе,} \end{cases}$$

$g(z)$ — прим.-рек. функция, принимающая 0, 1 (двоичная посл-ть).

Список литературы

1. Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по дискретной математике. — М.: Физматлит, 2005. — 416 с.
2. Марченков С. С. План семинарских занятий по курсу «Дополнительные главы дискретной математики», часть 2.

Дополнительные главы дискретной математики и кибернетики

Семинар 10

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Полиномиально решаемые задачи

Время вычисления машины Тьюринга

- Пусть A — алфавит, $\Lambda \notin A$ — пустой символ. Рассматриваем словарные функции $f: A^* \rightarrow A^*$.
- Вычисление функции на машине Тьюринга определяется так же, как и в случае числовых функций, только вместо кодов чисел на ленте записываются слова.
- Если функция имеет несколько аргументов, они разделяются символом Λ .
- Длина входа — это число ячеек в отрезке от самой левой непустой ячейки до самой правой непустой ячейки.
- Машина Тьюринга **вычисляет** функцию $f(w)$ **за время** $t(N)$, если она вычисляет эту функцию и при любом $N \geq 0$ машина останавливается на любом входе длины N не позже, чем через $t(N)$ тактов.

Полиномиально решаемые задачи

Полиномиально вычислимые функции

- Пусть $\mathcal{T}$ — семейство одноместных функций натурального аргумента. Машина Тьюринга $\mathcal{M}$ вычисляет функцию $f(w)$ за время $\mathcal{T}$, если существует $t(N) \in \mathcal{T}$ такое, что $\mathcal{M}$ вычисляет f за время $t(N)$.
- Функция **полиномиально вычислима**, если существует машина Тьюринга, которая вычисляет её за полиномиальное время.

$\mathcal{O}$ -символика для неотрицательных функций, $\bar{x} \rightarrow \infty$

- $\underline{\mathcal{O}}(f(\bar{x})) = \{g(\bar{x}) : (\exists c)(\exists \bar{x}_0)(\forall \bar{x} \geq \bar{x}_0)(g(\bar{x}) < c \cdot f(\bar{x}))\}$.
- $f(\bar{x}) = \underline{\mathcal{O}}(g(\bar{x}))$ значит $f(\bar{x}) \in \underline{\mathcal{O}}(g(\bar{x}))$.
- $\varphi(\underline{\mathcal{O}}(g_1(\bar{x})), \dots, \underline{\mathcal{O}}(g_m(\bar{x})))$ есть $\{\varphi(f_1(\bar{x}), \dots, f_m(\bar{x})) \mid f_i(\bar{x}) \in \underline{\mathcal{O}}(g_i(\bar{x})), i = \overline{1, m}\}$.

Полиномиально решаемые задачи

Многоленточные машины Тьюринга

- Многоленточная машина имеет фиксированное число лент с независимо движущимися головками.
- В каждой команде программы указано, как движется каждая головка.
- Вход и выход записываются на первой ленте; остальные ленты в начале работы пусты.

Полиномиально решаемые задачи

Многоленточные машины Тьюринга

- Если многоленточная машина вычисляет функцию за время $t(N) \geq N$, то одноленточная машина может вычислить эту функцию за время $t_1(N) = c_1 \cdot t^2(N)$.
 - ▶ Для этого нужно расширить алфавит машины и использовать механизм дорожек.
 - ▶ В каждой клетке хранится информация о соответствующих клетках всех лент и о наличии или отсутствии головок машины каждой ленты на этой клетке в текущий момент.
 - ▶ Для моделирования такта многоленточной машины нужно сделать один проход по всем данным на ленте, в процессе которого изменяются обозреваемые значения и положения головок, а также считываются новые значения.

Полиномиально решаемые задачи

Многоленточные машины Тьюринга

- Если одноленточная машина некотором алфавите вычисляет функцию за время $t_1(N)$, то одноленточная машина в другом алфавите вычислит функцию за время, не большее $t_2(N) = c_2 \cdot (t_1(N) + N^2)$.
 - Таким образом, можно промоделировать вычисления многоленточной машины с любым рабочим алфавитом одноленточной машиной, рабочий алфавит которой совпадает с входным, за время $t_2(N) = c \cdot (t^2(N) + N^2)$.
 - Если $t(N)$ — полином, то $t_2(N)$ — тоже полином.
-
- Мы будем проводить вычисления на многоленточных машинах, пользуясь любыми дополнительными символами. Оценка времени для одноленточной машины будет следовать автоматически.
 - В отличие от предыдущего раздела, для чисел мы всегда будем использовать двоичную запись: $x = (x_k \dots x_0)_2$.

Полиномиально решаемые задачи

[1], 2.3 (2), с. 12 (с изменениями).

Оценить сверху время работы машины Тьюринга с рабочим алфавитом $A \supseteq \{0, 1, \Lambda\}$ (Λ — пустой символ) и доказать, что указанные преобразования вычислимы за полиномиальное время.

2. Умножение двух целых чисел в двоичной записи.

Замечание

Длина входа N — сумма длин двоичной записи чисел плюс 1.

Полиномиально решаемые задачи

Решение [1], 2.3 (2), с. 12 (с изменениями).

- Сложение чисел в двоичной записи, записанных на разных лентах, требует линейного времени от длины максимального из этих чисел (сложение в столбик, в состояниях запоминаем перенос).
- Умножение в столбик: $xy = \sum_{i=0}^k x \cdot y_i \cdot 2^i$, где $y = (y_k \dots y_0)_2$.
- Имеем ленту с текущей суммой. Если $y_i = 0$, переходим к следующему разряду. Иначе за $\underline{O}(N)$ тактов переписываем очередное слагаемое на ещё одну ленту. Это слагаемое получается дописыванием нужного числа нулей в конец x .
- Производим прибавление нового слагаемого к текущей сумме. Это требует времени не больше, чем максимальная длины суммы, то есть длина слова xy . То есть времени $\underline{O}(N)$.
- Всего сложений может быть не более N . Итого время $\underline{O}(N^2)$.
- После перехода к одноленточной машине — $\underline{O}(N^4)$.

Полиномиально решаемые задачи

[1], 2.3 (4), с. 12 (с изменениями).

Оценить сверху время работы машины Тьюринга с рабочим алфавитом $A \supseteq \{0, 1, \Lambda\}$ (Λ — пустой символ) и доказать, что указанные преобразования вычислимы за полиномиальное время.

4. Нахождение определителя квадратной матрицы из нулей и единиц в поле $E_2 = \{0, 1\}$ с операциями сложения и умножения по модулю 2.

Замечание

Строки матрицы записаны друг за другом и разделены символами $|$.
Размер матрицы — n , длина записи $N = \underline{O}(n^2)$.

Полиномиально решаемые задачи

Решение [1], 2.3 (4), с. 12 (с изменениями).

- Применяем метод Гаусса. Если первый столбец нулевой (проверка линейна от N), выдаём 0. Иначе меняем местами строку с единицей в первом столбце с первой строкой матрицы (за счёт доп. ленты это делается за линейное от N время).
- Далее через доп. ленту вычитаем первую строку из других строк с единицей в первом столбце за линейное от N время.
- Повторяем то же самое для второго столбца (но проверяем и обрабатываем только строки ниже номера текущего столбца) и далее. Храним текущий номер столбца в унарном коде на отдельной ленте. Каждый раз операция линейна от N .
- Всего шагов будет столько, сколько столбцов, то есть $\underline{O}(\sqrt{N})$. Если в итоге в последней строке и последнем столбце есть единица, то выдаём 1, иначе 0.
- Итого $\underline{O}(N^{3/2}) = \underline{O}(n^3)$. Для одноленточной машины $\underline{O}(N^3)$.

Полиномиально решаемые задачи

[1], 2.3 (5), с. 12 (с изменениями).

Оценить сверху время работы машины Тьюринга с рабочим алфавитом $A \supseteq \{0, 1, \Lambda\}$ (Λ — пустой символ) и доказать, что указанные преобразования вычислимы за полиномиальное время.

5. Нахождение произведения двух квадратных матриц из нулей и единиц в поле $E_2 = \{0, 1\}$ с операциями сложения и умножения по модулю 2.

Замечание

Строки матрицы записаны друг за другом и разделены символами $|$. Размер матрицы — n , длина записи $N = \underline{O}(n^2)$. Матрицы $A = (a_{ij})$ и $B = (b_{ij})$ разделены пустым символом Λ .

Полиномиально решаемые задачи

Решение [1], 2.3 (5), с. 12 (с изменениями).

- Перепишем B на отдельную ленту в транспонированном виде.
- Для этого проходим по всей матрице B , выписывая на эту ленту подряд первые символы строк B , затирая их на исходной ленте специальным символом $*$.
- Возвращаемся в начало, и снова выписываем на вторую ленту первые символы строк (не считая затёртые). Продолжаем, пока не затрём все символы.
- Выписывание каждого столбца требует $\underline{O}(N)$ операций, а всего столбцов n . Таким образом, этот этап требует $\underline{O}(N^{3/2})$ операций.
- Вычисляем произведение поэлементно:
$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}.$$
- Индексы текущего элемента определяются положениями головок на лентах с матрицами A и B .

Полиномиально решаемые задачи

Решение [1], 2.3 (5), с. 12 (с изменениями; продолжение).

- Для расчёта одного элемента c_{ij} вычисляем скалярное произведение текущей строки A и столбца B (время $\underline{O}(n)$ — за счёт того, что матрица B выписана в транспонированном виде, то есть по столбцам).
- После этого либо нужно вернуться к началу строки и перейти к следующему столбцу ($\underline{O}(n)$ тактов), либо перейти к следующей строке и вернуться к первому столбцу ($\underline{O}(N)$ тактов).
- Всего элементов матрицы n^2 , то есть без учёта возвратов к первому столбцу требуется $\underline{O}(n^3) = \underline{O}(N^{3/2})$ тактов.
- Всего возвратов к первому столбцу не более n , поэтому на все возвраты требуется $\underline{O}(nN) = \underline{O}(N^{3/2})$ тактов.
- Итого $\underline{O}(N^{3/2})$. При переходе к одноленточной машине получим $\underline{O}(N^3)$.

Полиномиально решаемые задачи

Перебор значений двоичного числа

- Рассмотрим процесс перебора всех чисел от a до 0: вычитание по одной единице из a , пока не достигнем нуля.
- Этот процесс требуется, если мы хотим сдвинуть какую-то головку ровно на a ячеек и имеем число a в двоичном виде. Могут быть и другие способы применения этого процесса.
- Считаем, что число a выписано на отдельной ленте машины Тьюринга в двоичном виде. В начале каждого вычитания единицы головка машины находится на последнем разряде числа.
- Вычитание единицы требует не более $\underline{O}(\log_2 a)$ тактов, всего вычитаний a . Итого $\underline{O}(a \log_2 a)$ тактов.
- Оценка выше — это грубая оценка сверху. Далеко не все вычитания требуют $\underline{O}(\log_2 a)$ тактов. Покажем, что на самом деле процесс длится $\underline{O}(a)$ тактов.

Полиномиально решаемые задачи

Перебор значений двоичного числа

- Вычитание единицы из нечётного числа требует 1 такт. Таких вычитаний не более $\lceil a/2 \rceil$.
- Вычитание единицы из чётного числа, не делящегося на 4, требует 2 такта (с учётом возвращения головки). Таких вычитаний не более $\lceil a/4 \rceil$.
- Вычитание единицы из числа, делящегося на 2^i и не делящегося на 2^{i+1} требует $2i$ тактов, таких вычитаний не более $\lceil a/2^{i+1} \rceil$.
- Итого тактов

$$t(a) \leq \lceil a/2 \rceil + \sum_{i=1}^{\lceil \log_2(a+1) \rceil} 2i \lceil a/2^{i+1} \rceil \leq \underline{O}(a) + a \sum_{i=1}^{\infty} \frac{i}{2^i}.$$

- $\overline{\lim}_{i \rightarrow \infty} \sqrt[i]{\frac{i}{2^i}} = \frac{1}{2}$. По признаку Коши ряд сходится и $t(a) = \underline{O}(a)$.

Полиномиально решаемые задачи

[1], 2.3 (7), с. 12 (с изменениями).

Оценить сверху время работы машины Тьюринга с рабочим алфавитом $A \supseteq \{0, 1, \Lambda\}$ (Λ — пустой символ) и доказать, что указанные преобразования вычислимы за полиномиальное время.

7. Нахождение значений булевой функции по формуле над базисом $\{\&, \vee, \neg\}$ на данном наборе $(\alpha_1, \dots, \alpha_n)$ значений переменных.

Замечание

Формула записывается на ленте с использованием символов $(,), \&, \vee, \neg, x, 0, 1$. Индексы переменных записываются в двоичном виде. После формулы через разделитель Λ пишется набор значений. Длина набора n , длина входа N равна длине формулы плюс $n + 1$.

Полиномиально решаемые задачи

Решение [1], 2.3 (7), с. 12 (с изменениями).

- За время $\underline{O}(N)$ выписываем вектор значений переменных на отдельную ленту.
- Далее в формуле последовательно заменяем определённые подформулы на их значения на указанном наборе.
- Общий ход замены такой: проходим по текущей формуле и переносим её на другую ленту. Заменяемый блок не переносим; вместо него записываем 0 или 1. При этом формула переходит на другую ленту.
- В следующий раз считываем формулу со второй ленты и записываем на первую; таким образом роль лент на каждом шаге меняется местами.

Полиномиально решаемые задачи

Решение [1], 2.3 (7), с. 12 (с изменениями; продолжение).

- Сначала заменяем переменные на их значения. При нахождении переменной в формуле выписываем на отдельную ленту индекс переменной i , последовательно уменьшая его на 1, ищем значение переменной на ленте с набором значений.
- Индекс не может быть более n , поэтому весь процесс вычитания единиц требует $\underline{O}(n)$ операций. Всего этот этап требует $\underline{O}(Nn) = \underline{O}(N^2)$ операций.
- Далее преобразовываем формулу: ищем операцию, в которую подставляются только константы, заменяем её на результат. Каждый шаг такого преобразования производится за $\underline{O}(N)$ и удаляет хотя бы одну операцию, всего шагов не более $\underline{O}(N)$.
- Итого $\underline{O}(N^2) + \underline{O}(N^2) = \underline{O}(N^2)$ операций. При переходе к одноленточной машине получим $\underline{O}(N^4)$.

Полиномиально решаемые задачи

[2], занятие 10.

Доказать, что задача определения существенных переменных у функции, заданной таблицей значений, может быть выполнена на машине Тьюринга за полиномиальное время.

Замечание

У функции n переменных, длина записи $N = 2^n$.

Полиномиально решаемые задачи

Решение [2], занятие 10.

- Проверяем существенность каждой переменной, начиная с последней, результаты записываем на отдельную ленту.
- Для проверки существенности переменной x_{n-i+1} сравниваем соседние блоки из i наборов по всей таблице (если блоки каждой пары совпадают, то переменная фиктивна). Храним номер i в унарном коде на отдельной ленте.
- Чтобы произвести одно сравнение, идём по таблице, выписывая блок нужной длины на отдельную ленту. Потом идём далее и сравниваем его со следующим блоком (одновременно стирая).
- Если нашли различие, переходим к следующей переменной. Если нет, то проверяем таблицу до конца.
- Проверка одной переменной требует $\underline{O}(N)$ операций. Весь процесс требует $\underline{O}(N \log_2 N)$ операций. При переходе к одноленточной машине получим $\underline{O}(N^2 \log_2^2 N)$.

Полиномиально решаемые задачи

[1], 2.19 (7), с. 16 (с изменениями).

Доказать полиномиальность задач.

7. Получение правильной раскраски графа в 2 цвета (2 смежные вершины всегда имеют разные цвета) или ответа, что её не существует.

Замечание

Граф записан в виде матрицы смежности, строки матрицы записаны друг за другом и разделены символами $|$. n вершин, размер матрицы n , длина записи $N = \underline{O}(n^2)$. На выход подаём набор из n чисел 1, 2 — цвета вершин, либо «нет», если раскраски не существует.

Полиномиально решаемые задачи

Решение [1], 2.19 (7), с. 16 (с изменениями).

- Раскраска имеется, если граф имеет циклы только чётной длины.
- Производим обход графа в ширину. Помечаем первую вершину цветом 1, смежные с ней цветом 2, следующие в обходе за каждым цветом i — цветом $3 - i$.
- Если попали в уже пройденную вершину, проверяем, должны ли мы поставить тот же цвет, что там уже поставлен. Если другой — останавливаемся и выдаём «нет». Иначе не продолжаем эту ветвь обхода, но продолжаем обход в целом.
- По итогу получим правильную раскраску в 2 цвета.
- Опишем этот процесс детально и оценим его сложность.

Полиномиально решаемые задачи

Решение [1], 2.19 (7), с. 16 (с изменениями; продолжение).

- Для организации обхода в ширину храним очередь номеров вершин (в двоичной записи) на отдельной ленте. Также на отдельной ленте храним набор цветов вершин (цвета 1, 2; 0 — ещё не раскрасили). Вначале это набор нулей.
- Сначала записываем в очередь номер 1 ($\underline{O}(1)$ операций).
- Обработывая каждую вершину, берём её из очереди и ставим на отдельную ленту (число операций не более длины очереди $\underline{O}(n \log_2 n)$, после выбора возвращаем головку в конец очереди).
- Далее за $\underline{O}(n^2 + n)$ операций ищем строку текущей вершины в матрице смежности (перебираем строки, вычитая по единице из выбранного номера; все вычитания вместе требуют $\underline{O}(n)$ операций; сам обход матрицы — не более $\underline{O}(n^2)$).

Полиномиально решаемые задачи

Решение [1], 2.19 (7), с. 16 (с изменениями; продолжение).

- За $\underline{O}(n)$ операций (с помощью вычитания единиц) ищем позицию вершины в наборе цветов вершин, запоминая цвет вершины.
- За $\underline{O}(n)$ тактов проходим по строке вершины с увеличением текущего двоичного индекса строки на отдельной ленте; номера смежных с данной вершин записываем на отдельную ленту (это требует не более $\underline{O}(n \log_2 n)$ операций).
- Проходим по номерам смежных с данной вершин. За $\underline{O}(n)$ операций ищем позицию вершины в наборе цветов вершин. Если там указан цвет 1 или 2, а цвет последней взятой из очереди вершины другой, завершаем процесс и выдаём «нет». Если цвет тот же, то пропускаем эту вершину.
- Если у текущей смежной вершины цвет 0, то записываем вместо него цвет, противоположный вершине из очереди; а номер самой смежной вершины записываем в конец очереди ($\underline{O}(\log_2 n)$).

Полиномиально решаемые задачи

Решение [1], 2.19 (7), с. 16 (с изменениями; продолжение).

- Всего обход не более n смежных вершин требует $\underline{O}(n^2)$ тактов. Столько же требует и вся обработка вершины.
- Если очередь опустела, завершаем работу. Всего будет обработано не более n вершин, итоговая сложность $\underline{O}(n^3) = \underline{O}(N^{3/2})$. При переходе к одноленточной машине получим $\underline{O}(N^3)$.

Полиномиально решаемые задачи

[1], 2.20 (1a), с. 17 (с изменениями).

Доказать полиномиальность задач.

- 1a) Распознавание нелинейности булевой функции по её таблице значений.

Замечание

У функции n переменных, длина записи $N = 2^n$.

Полиномиально решаемые задачи

Решение [1], 2.20 (1a), с. 17 (с изменениями).

- Функция линейна, если она линейна по всем переменным.
- Для проверки линейности x_{n-i+1} сравниваем соседние блоки из i наборов по всей таблице (если блоки каждой пары совпадают, либо блоки каждой пары противоположны, то функция линейна по x_{n-i+1}). Храним номер i в унарном коде на отдельной ленте.
- Чтобы произвести одно сравнение, идём по таблице, выписывая блок нужной длины на отдельную ленту. Потом идём далее и сравниваем его со следующим блоком (одновременно стирая). Для первой пары запоминаем, равны они или противоположны.
- Если нашли нарушение линейности, переходим к следующей переменной. Если нет, то проверяем таблицу до конца.
- Проверка линейности по одной переменной требует $\underline{O}(N)$ операций. Весь процесс требует $\underline{O}(N \log_2 N)$ операций. При переходе к одноленточной машине получим $\underline{O}(N^2 \log_2^2 N)$.

Полиномиально решаемые задачи

[1], 2.20 (2а), с. 17 (с изменениями).

Доказать полиномиальность задач.

2а) Распознавание немонотонности булевой функции по её таблице значений.

Замечание

У функции n переменных, длина записи $N = 2^n$.

Полиномиально решаемые задачи

Решение [1], 2.20 (2a), с. 17 (с изменениями).

- Функция монотонна тогда и только тогда, когда она монотонна по каждой переменной.
- Для каждой переменной перебираем 2^{n-1} наборов значений остальных переменных и сравниваем значение получающейся функции одной переменной в нуле и в единице.
- Получим $\underline{O}(N \log_2 N)$ шагов, каждый из которых полиномиален.
- Поэтому вся задача полиномиальна.

Полиномиально решаемые задачи

[1], 2.20 (2д), с. 17 (с изменениями).

Доказать полиномиальность задач.

2д) Распознавание немонотонности булевой функции по её полиному Жегалкина.

Замечание

Полином записывается на ленте с использованием символов $\oplus, \cdot, x, 0, 1$. Индексы переменных записываются в двоичном виде. Длина записи N .

Полиномиально решаемые задачи

Решение [1], 2.20 (2д), с. 17 (с изменениями).

- Нижняя единица функции — это набор на котором функция выдаёт 1, при этом на всех меньших наборах выдавая 0.
- Сначала нужно найти среди всех ЭК полинома Жегалкина множество минимальных по включению переменных ЭК. Это множество соответствует множеству нижних единиц функции.
- Для этого сравниваем каждое слагаемое с каждым, и находим набор минимальных за полиномиальное время.
- Теперь нужно проверить, что для каждой минимальной ЭК при подстановке вместо её переменных единиц в полином получится тождественная единица.
- Если это верно, то полином задаёт функцию, у которой выше нижних единиц находятся только единицы, то есть монотонную функцию.
- Вся проверка полиномиальна.

Полиномиально решаемые задачи

[1], 2.20 (3а), с. 17 (с изменениями).

Доказать полиномиальность задач.

3а) Распознавание несамодвойственности булевой функции по её таблице значений.

Замечание

У функции n переменных, длина записи $N = 2^n$.

Полиномиально решаемые задачи

Решение [1], 2.20 (3а), с. 17 (с изменениями).

- Вектор самодвойственной функции антисимметричен.
- Копируем вектор на другую ленту.
- Проходим по вектору на одной ленте сначала, а на другой с конца, проверяя антисимметричность.
- Требуется время $\underline{O}(N)$.

Полиномиально решаемые задачи

[1], 2.20 (4а), с. 17 (с изменениями).

Доказать полиномиальность задач.

- 4а) Определение существования в графе k -клики (полного графа на k вершинах, являющегося подграфом данного), если число k — фиксированная константа.

Замечание

Граф записан в виде матрицы смежности, строки матрицы записаны друг за другом и разделены символами $|$. n вершин, размер матрицы n , длина записи $N = \underline{O}(n^2)$.

Полиномиально решаемые задачи

Решение [1], 2.20 (4а), с. 17 (с изменениями).

- Перебираем все n^k наборов из k вершин.
- Для каждого набора проверяем, что вершины различны и что между всеми вершинами есть рёбра.
- Проверка требует полиномиального времени, число проверок $\underline{O}(n^k)$.

Домашнее задание

[1], 2.3 (6), с. 12 (с изменениями).

Оценить сверху время работы машины Тьюринга с рабочим алфавитом $A \supseteq \{0, 1, \Lambda\}$ (Λ — пустой символ) и доказать, что указанные преобразования вычислимы за полиномиальное время.

6. Решение системы линейных алгебраических уравнений в поле $E_2 = \{0, 1\}$ с операциями сложения и умножения по модулю 2.

Замечание

Считаем, что матрица коэффициентов СЛАУ квадратная и невырожденная (то есть, у неё существует единственное решение). СЛАУ записывается в виде матрицы коэффициентов, после которой через разделитель $\#$ записан вектор правой части. Строки матрицы записаны друг за другом и разделены символами $|$. Размер матрицы — n , длина записи $N = \underline{O}(n^2)$.

Домашнее задание

[1], 2.3 (8), с. 12 (с изменениями).

Оценить сверху время работы машины Тьюринга с рабочим алфавитом $A \supseteq \{0, 1, \Lambda\}$ (Λ — пустой символ) и доказать, что указанные преобразования вычислимы за полиномиальное время.

8. Нахождение набора, на котором полином Жегалкина обращается в ноль.

Замечание

Полином записывается на ленте с использованием символов $\oplus, \cdot, x, 0, 1$. Индексы переменных записываются в двоичном виде. Длина записи N . Длина искомого набора совпадает с числом различных переменных, встречающихся в полиноме, значения упорядочены в порядке возрастания индексов переменных.

Домашнее задание

[1], 2.3 (9), с. 12 (с изменениями).

Оценить сверху время работы машины Тьюринга с рабочим алфавитом $A \supseteq \{0, 1, \Lambda\}$ (Λ — пустой символ) и доказать, что указанные преобразования вычислимы за полиномиальное время.

9. Нахождение кратчайшего расстояния между данной парой вершин в графе.

Замечание

Граф записан в виде матрицы смежности, строки матрицы записаны друг за другом и разделены символами $|$. n вершин, размер матрицы n . Номера вершин подаются в двоичной записи отдельными аргументами через разделители Λ . Длина записи $N = \underline{O}(n^2)$.

Домашнее задание

[1], 2.3 (10), с. 12 (с изменениями).

Оценить сверху время работы машины Тьюринга с рабочим алфавитом $A \supseteq \{0, 1, \Lambda\}$ (Λ — пустой символ) и доказать, что указанные преобразования вычислимы за полиномиальное время.

10. Построение остовного дерева графа.

Замечание

Граф записан в виде матрицы смежности, строки матрицы записаны друг за другом и разделены символами $|$. n вершин, размер матрицы n , длина записи $N = \underline{O}(n^2)$. Результат выдаём в том же формате.

Домашнее задание

[1], 2.19 (1), с. 16 (с изменениями).

Доказать полиномиальность задач.

10. Определение существования двух противоположных наборов, на которых ДНФ обращается в 1.

Замечание

ДНФ записывается на ленте с использованием символов $\&$, $\vee$, $\neg$, x , 0 , 1 . Индексы переменных записываются в двоичном виде. Длина записи N .

Домашнее задание

[1], 2.20 (1г), с. 17 (с изменениями).

Доказать полиномиальность задач.

1г) Распознавание нелинейности булевой функции по её сокращённой ДНФ.

Замечание

ДНФ записывается на ленте с использованием символов $\&$, $\vee$, $\neg$, x , 0 , 1 . Индексы переменных записываются в двоичном виде. Длина записи N .

Домашнее задание

[1], 2.20 (2г), с. 17 (с изменениями).

Доказать полиномиальность задач.

2г) Распознавание немонотонности булевой функции по её сокращённой ДНФ.

Замечание

ДНФ записывается на ленте с использованием символов $\&$, $\vee$, $\neg$, x , 0 , 1 . Индексы переменных записываются в двоичном виде. Длина записи N .

Домашнее задание

[1], 2.20 (3г), с. 17 (с изменениями).

Доказать полиномиальность задач.

3г) Распознавание несамодвойственности булевой функции по её совершенной ДНФ.

Замечание

ДНФ записывается на ленте с использованием символов $\&, \vee, \neg, x, 0, 1$. Индексы переменных записываются в двоичном виде. У функции n переменных, длина записи $N = \underline{O}(n \cdot 2^n)$.

Список литературы

1. Алексеев В. Б., Вороненко А. А., Ложкин С. А., Романов Д. С., Сапоженко А. А., Селезнёва С. Н. Задачи по курсу «Основы кибернетики» (2-е изд.) — М.: Издательский отдел факультета ВМиК МГУ, 2011. — 71 с.
2. Марченков С. С. План семинарских занятий по курсу «Дополнительные главы дискретной математики», часть 2.

Дополнительные главы дискретной математики и кибернетики

Семинар 11

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Клики

Определение

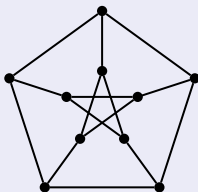
- Пусть $G = (V, E)$ — граф.
- **Клика** графа G — это подграф G , являющийся полным графом.
- **Размер клики** — это число её вершин.

Клики

[1], 2.5 (рис. 1), с. 13.

Найти размер максимальной клики графа.

1) Граф Петерсена:



Клики

Решение [1], 2.5 (рис. 1), с. 13.

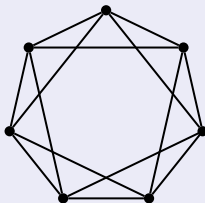
- Легко видеть, что в графе нет циклов длины 3. Это значит, что в нём нет клик размера 3 и более.
- Максимальная клика имеет размер 2 (любое ребро графа).

Клики

[1], 2.5 (рис. 3), с. 13.

Найти размер максимальной клики графа.

3) Граф



Клики

Решение [1], 2.5 (рис. 3), с. 13.

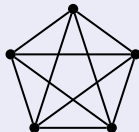
- В графе есть клики размера 3.
- Легко видеть, что ни один цикл длины 4 в графе не может не содержать граничных рёбер семиугольника
- Цикл длины 4 может содержать 2 ребра семиугольника только если эти рёбра идут через одно. Таких циклов 7, и они не образуют клик размера 4.
- Циклов длины 4, содержащих ровно одно граничное ребро семиугольника, 7, и ни один из них не образует клику размера 4.
- Максимальная клика имеет размер 3.

Клики

[1], 2.5 (рис. 6), с. 13.

Найти размер максимальной клики графа.

6) Граф K_5 :



Клики

Решение [1], 2.5 (рис. 6), с. 13.

- Максимальная клика имеет размер 5 (весь граф).

Класс P

Задачи и предикаты

- Задача распознавания — это предикат: $A^* \rightarrow \{\text{true}, \text{false}\}$.
- Например, задача проверки линейности функции по её таблице значений — это предикат, который выдаёт «да» тогда и только тогда, когда ему на вход подана таблица значений линейной функции.
- Решение задачи — это вычисление предиката на данном входе.

Класс P

- Рассмотрим машины Тьюринга (с входным алфавитом A), которые всегда останавливаются и могут выдавать только результаты вида «да» (1) или «нет» (0).
- Эти машины вычисляют некоторые предикаты на A^* .
- **Класс P** — это класс всех предикатов, распознаваемых на машинах Тьюринга за полиномиальное время.

Недетерминированная машина Тьюринга

- Недетерминированная машина Тьюринга (с входным алфавитом A) может иметь несколько команд вида $(a, q_i) \rightarrow (b, D, q_j)$ с одинаковой левой частью.
- Каждый раз, когда машина обозревает символ a и находится в состоянии q_i , она может выполнить любую из своих команд с левой частью (a, q_i) .
- Таким образом, на одном и том же входе машина может работать, вообще говоря, бесконечным числом различных способов, используя разные варианты выбора команд на каждом шаге вычисления.
- Недетерминированная машина Тьюринга, подобно автомату-распознавателю, может выдавать только результаты вида «да» или «нет».

Недетерминированная машина Тьюринга

- Недетерминированная машина Тьюринга **допускает слово** $\bar{a} \in A^*$ (результат «да»), если, начиная работу на входе $\bar{a}$, она останавливается хотя бы в одном из своих способов вычисления.
- В противном случае машина **отвергает слово** $\bar{a} \in A^*$ (результат «нет»).
- Недетерминированная машина Тьюринга **допускает слово** $\bar{a} \in A^*$ **за время** t , если, начиная работу на входе $\bar{a}$, она хотя бы в одном из своих способов вычисления останавливается не позже, чем через t тактов.
- В противном случае машина **отвергает слово** $\bar{a} \in A^*$ **за время** t .
- Недетерминированная машина Тьюринга **вычисляет предикат** $\rho(x)$ на A^* **за время** $T(N)$, если для любого $\bar{a} \in A^*$ машина допускает слово $\bar{a}$ за время $T(|\bar{a}|)$ тогда и только тогда, когда $\rho(\bar{a})$ истинно.

Класс NP

Определение

Класс NP — это класс всех предикатов, вычислимых на недетерминированных машинах Тьюринга за полиномиальное время.

Определение (равносильное)

Предикат $\rho(x)$ принадлежит **NP**, если существует полином $q(N)$ и полиномиально вычислимый предикат $Q(x, y)$ такой, что

$$\rho(x) = (\exists y)_{|y| \leq q(|x|)} Q(x, y).$$

- Функция $Q(x, y)$ называется функцией проверки сертификата.
- Слово $y = y(x)$ такое, что $Q(x, y)$ истинно, называется сертификатом для входа x .
- $\rho(x)$ истинно, если существует сертификат для x (имеющий длину, полиномиальную от длины x).

Класс NP

Содержательно о классе NP

- NP — это класс задач распознавания, для проверки положительного ответа на которые существуют сертификаты полиномиальной длины, проверяемые за полиномиальное время.
- NP — это класс задач, которые можно сформулировать как задачи проверки существования объекта с некоторыми свойствами (причём проверить, обладает ли этим свойством один конкретный объект, можно полиномиально).
- Задачи из NP можно решать перебором, но это требует экспоненциального времени (из-за экспоненциального количества перебираемых объектов). Неизвестно, можно ли для этих задач придумать алгоритм, избегающий перебора.
- На практике для решения задач из NP применяют SAT-солверы, использующие оптимизированный перебор. Обычно это работает, но нет гарантии быстрой работы во всех случаях.

Класс NP

- $P \subseteq NP$. Справедливость обратного включения неизвестна.

Определение

- Предикат $\rho(x)$ **полиномиально сводится** к $\sigma(y)$, если существует полиномиально вычисляемое преобразование $y = f(x)$, такое, что $\rho(x) = \sigma(f(x))$.
- Задача распознавания **NP-трудна**, если любая задача из класса NP полиномиально сводится к данной.
- Задача распознавания **NP-полна**, если она принадлежит классу NP и является NP-трудной.

Класс NP

- NP-полные задачи — это самые «труднорешаемые» задачи из класса NP. Каждая такая задача «содержит в себе» всю сложность задач из NP.
- Если хотя бы одна NP-полная задача принадлежит P, то $P = NP$.

Свойства сводимости

- Если ρ полиномиально сводится к σ , а σ полиномиально сводится к τ , то ρ полиномиально сводится к τ .
- Все задачи из P полиномиально сводятся друг к другу.
- Если $\rho \in P$ и σ полиномиально сводится к ρ , то $\sigma \in P$.
- Если $\rho \in NP$ и σ полиномиально сводится к ρ , то $\sigma \in NP$.
- Если σ NP-трудна и σ полиномиально сводится к ρ , то ρ NP-трудна.

Примеры NP-полных задач

- Задача ВЫП. Дана КНФ. Существует ли набор значений переменных, на которых КНФ обращается в 1? (Выполнима ли КНФ?)
- Задача 3-ВЫП. Дана КНФ, в каждой элементарной дизъюнкции которой не более трёх букв (3-КНФ). Выполнима ли она?
- Чтобы доказать, что некоторая задача NP-полна, нужно:
 - ▶ Доказать, что задача из NP: предъявить легко проверяемый сертификат.
 - ▶ Свести к данной задаче одну из NP-полных задач (например, задачу ВЫП).

NP-полные задачи

[1], 2.17 (задача 2), с. 16.

Доказать NP-полноту задачи:

- 2) КЛИКА. Дан граф $G = (V, E)$ в виде матрицы смежности и число $k \leq n$. Определить, существует ли в G клика размера k .

NP-полные задачи

Решение [1], 2.17 (задача 2), с. 16.

- Сертификат для графа и числа k — это набор из k номеров его вершин, образующих клику:
 - ▶ Если в графе существует клика из k вершин, то сертификат существует и длина его записи полиномиальна (не превосходит длины записи графа).
 - ▶ Если в графе нет клики, то сертификата не существует.
 - ▶ Функция проверки сертификата полиномиальна: она проверяет, является ли вход набором из k различных чисел, не превосходящих числа вершин графа, и соединены ли все указанные вершины рёбрами между собой ($\frac{k(k-1)}{2}$ проверок).
- Таким образом, КЛИКА принадлежит NP.
- Полиномиально сведём задачу ВЫП к задаче КЛИКА. Имеем КНФ $K = D_1 \& \dots \& D_s$, где $D_i = t_{i1} \vee \dots \vee t_{im_i}$, $i = \overline{1, s}$.

NP-полные задачи

Решение [1], 2.17 (задача 2), с. 16 (продолжение).

- В качестве числа k выберем число ЭД s . Строим граф $G = (V, E)$. В качестве множества вершин V выберем множество всех букв КНФ K (с повторениями).
- Во множество рёбер включим всевозможные пары букв, которые принадлежат разным ЭД и не противоположны друг другу.
- Если КНФ выполнима, то на некотором наборе все её ЭД обращаются в 1. В каждой ЭД хотя бы одна буква обратилась в 1. Выберем s таких букв из разных ЭД — среди этих букв не может быть противоположных, и они образуют клику размера s в графе.
- Если в графе есть клика размера s , то среди её букв нет противоположных и существует набор, который обратит все их в 1. Поскольку клика содержит по одной букве каждой ЭД, этот набор обращает в 1 КНФ и она выполнима.
- Сведение полиномиально. Задача КЛИКА NP-полна.

NP-полные задачи

[1], 2.18 (1), с. 16.

Доказать, что задачи являются NP-полными:

1. Определение существования набора, обращающего данную ДНФ в 0.

NP-полные задачи

Решение [1], 2.18 (1), с. 16.

- Сертификат — искомый набор, проверка полиномиальна. Задача принадлежит NP.
- Сведём ВЫП к нашей задаче. Пусть дана КНФ K .
- Заменяем все конъюнкции на дизъюнкции и наоборот, инвертируем отрицания над буквами. Получили ДНФ D , равную отрицанию K .
- Тогда набор, обращающий D в 0, существует тогда и только тогда, когда существует набор, обращающий K в 1.
- Сведение полиномиально. Задача NP-полна.

NP-полные задачи

[1], 2.18 (4), с. 16 (с изменениями).

Доказать NP-полноту задачи.

4. Определение существования набора, на котором заданные полиномы Жегалкина (их количество заранее неизвестно) одновременно обращаются в 0.

NP-полные задачи

Решение [1], 2.19 (4), с. 16 (с изменениями).

- Сертификат — искомый набор, проверка полиномиальна. Задача принадлежит NP.
- Сведём задачу о существовании набора, обращающего ДНФ в 0, к нашей задаче. Пусть дана ДНФ D .
- По каждой ЭК строим эквивалентный ей полином Жегалкина (он состоит только из одного слагаемого). Получаем набор полиномов.
- Все полиномы обращаются в 0 на наборе тогда и только тогда, когда на нём обращается в 0 исходная ДНФ.
- Сведение полиномиально. Задача NP-полна.

NP-полные задачи

[1], 2.20 (16), с. 17 (с изменениями).

Доказать NP-полноту задачи.

- 16) Распознавание нелинейности булевой функции, если функция задана формулой.

Замечание

Считаем, что имеется в виду формула с функциональными символами $\{\vee, \&, \neg\}$.

NP-полные задачи

Решение [1], 2.20 (16), с. 17 (с изменениями).

- Функция $f(x_1, \dots, x_n)$ линейна тогда и только тогда для любых $\alpha, \beta \in E_2^n$ верно $f(\alpha \oplus \beta) = f(\alpha) \oplus f(\beta) \oplus f(0, \dots, 0)$ (для доказательства справа налево используем лемму о нелинейной функции: достаточно будет подобрать наборы для $\&$ и $\vee$).
- Сертификат — 2 набора, на которых нарушается указанное выше условие, проверка полиномиальна. Задача принадлежит NP.
- Сведём задачу ВВП, к нашей задаче. Пусть дана КНФ K .
- Домножим её на $x_i \cdot x_j$, где индексы i, j не встречались у переменных КНФ.
- Если исходная КНФ была выполнима, то получившаяся функция будет нелинейна (полином ненулевой, а каждое слагаемое содержит буквы x_i, x_j). Если исходная КНФ была невыполнима, то получившаяся функция будет равна 0 (линейна).
- Сведение полиномиально. Задача NP-полна.

NP-полные задачи

[1], 2.20 (1в), с. 17 (с изменениями).

Доказать NP-полноту задачи.

- 1в) Распознавание нелинейности булевой функции, если функция задана в виде ДНФ.

NP-полные задачи

Решение [1], 2.20 (1в), с. 17 (с изменениями).

- Сертификат — 2 набора, на которых нарушается критерий (см. 2.20 (1б)), проверка полиномиальна. Задача принадлежит NP.
- Сведём задачу о существовании набора, обращающего ДНФ в 0, к нашей задаче. Пусть дана ДНФ D .
- Если $D(0, \dots, 0) = 0$, выдаём ДНФ x_1x_2 . Иначе выдаём ДНФ $x_i \vee D$, где x_i не встречается среди переменных D .
- Если существует набор, обнуляющий D , то либо будет выдана нелинейная функция x_1x_2 , либо D — не константа и $D(0, \dots, 0) = 1$. Тогда $x_i \vee D$ нелинейна: при подстановке вместо всех переменных, кроме x_i , нулей получим тождественную 1 (но x_i существенна), что невозможно для линейной функции.
- Если исходная ДНФ — константа 1, то получим линейную функцию 1.
- Сведение полиномиально. Задача NP-полна.

NP-полные задачи

[1], 2.20 (26), с. 17 (с изменениями).

Доказать NP-полноту задачи.

26) Распознавание немонотонности булевой функции, если функция задана формулой.

Замечание

Считаем, что имеется в виду формула с функциональными символами $\{\vee, \&, \neg\}$.

NP-полные задачи

Решение [1], 2.20 (26), с. 17 (с изменениями).

- Сертификат — 2 набора, на которых нарушается монотонность, проверка полиномиальна. Задача принадлежит NP.
- Сведём задачу ВВП к нашей задаче. Пусть дана КНФ K .
- Домножим K на $\bar{x}_i$, где индекс i не встречался у переменных K . Получим желаемую формулу.
- Если исходная КНФ была выполнима, то получившаяся функция будет немонотонна (по переменной $\bar{x}_i$). Если исходная КНФ была невыполнима, то получившаяся функция будет равна 0 (монотонна).
- Сведение полиномиально. Задача NP-полна.

NP-полные задачи

[1], 2.20 (3в), с. 17 (с изменениями).

Доказать NP-полноту задачи.

3в) Распознавание несамодвойственности булевой функции, если функция задана в виде ДНФ.

NP-полные задачи

Решение [1], 2.20 (Зв), с. 17 (с изменениями).

- Сертификат — набор, нарушающий антисимметричность вектора значений, проверка полиномиальна. Задача принадлежит NP.
- Сведём задачу о существовании набора, обращающего ДНФ в 0, к нашей задаче. Пусть дана ДНФ D .
- Добавим в каждую ЭД D множитель x_i , где индекс i не встречался у переменных D , и перенумеруем переменные так, чтобы новая переменная стала первой. Получим требуемую ДНФ.
- Если D имеет набор, обращающий её в 0, то новая ДНФ имеет нули в первой половине своего вектора (при $x_i = 0$) и хотя бы один ноль во второй половине своего вектора (при $x_i = 1$).
- Если D является константой 1, то новая ДНФ задаёт самодвойственную функцию x .
- Сведение полиномиально. Задача NP-полна.

Список литературы

1. Алексеев В. Б., Вороненко А. А., Ложкин С. А., Романов Д. С., Сапоженко А. А., Селезнёва С. Н. Задачи по курсу «Основы кибернетики» (2-е изд.) — М.: Издательский отдел факультета ВМиК МГУ, 2011. — 71 с.
2. Марченков С. С. План семинарских занятий по курсу «Дополнительные главы дискретной математики», часть 2.

Дополнительные главы дискретной математики и кибернетики

Семинар 12

Савицкий Игорь Владимирович

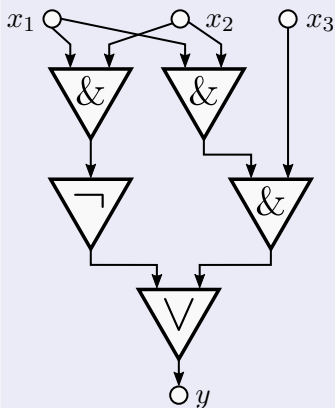
факультет ВМК МГУ

осень 2020

Множества функций и операторов

- P_2 — множество функций алгебры логики (ФАЛ)
 $f: \{0, 1\}^n \rightarrow \{0, 1\}, n \in \mathbb{N}$.
- P_2^m — множество булевых операторов с m выходами
 $f: \{0, 1\}^n \rightarrow \{0, 1\}^m, n \in \mathbb{N}$.
- Пусть Q — класс ФАЛ или булевых операторов. Тогда $Q(n)$ — класс всех функций/операторов из Q , зависящих ровно от n переменных.

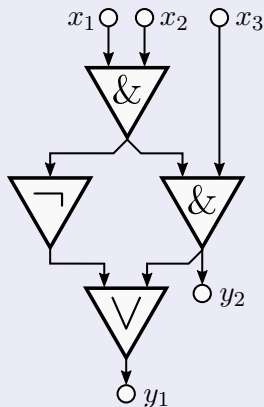
Формула



$$y = \overline{x_1 x_2} \vee (x_1 x_2) x_3$$

- Число входных полюсов равно числу различных символов переменных.
- Во входные полюса не входят дуги, но из них может исходить любое число дуг.
- Выходной полюс один, в него входит одна дуга, и из него не исходит дуга.
- В функциональный элемент входит по одной дуге на каждый аргумент его функции, а исходит ровно одна дуга.
- Формула без входных полюсов представляет собой дерево.
- Формула реализует ФАЛ.

Схема из функциональных элементов (СФЭ)



$$y_1 = \overline{x_1 x_2} \vee (x_1 x_2) x_3$$

$$y_2 = x_1 x_2 x_3$$

Отличия от формулы:

- Из каждого функционального элемента может исходить любое число дуг.
- СФЭ может не являться деревом, но в ней нет контуров.
- Выходных полюсов может быть несколько.
- СФЭ реализует булев оператор

$$F: \{0, 1\}^n \rightarrow \{0, 1\}^m.$$

- В примере $F = (\overline{x_1 x_2} \vee x_1 x_2 x_3, x_1 x_2 x_3)$.

Сложность реализации класса

- Пусть $Q(n) \subseteq P_2^{m(n)}(n)$. Если $m(n) \equiv 1$, то это класс функций; иначе это класс операторов.
- Сложность СФЭ — это число функциональных элементов в ней.
- $L^C(f)$ — минимальная сложность реализации функции/оператора с помощью СФЭ в стандартном базисе $\{\&, \vee, \neg\}$.
- $L^C(Q(n)) = \max_{f \in Q(n)} L^C(f)$ — **функция Шеннона** для класса Q .
- Функция Шеннона для класса Q зависит от n (числа переменных) и выражает минимальную сложность реализации самой сложной функции из Q от n переменных.
- $L^C(P_2(n)) \sim \frac{2^n}{n}$.

Асимптотические соотношения

- $\bar{o}(f(\bar{x})) = \{g(\bar{x}) : (\exists \alpha(\bar{x})) (\lim_{\bar{x} \rightarrow \infty} \alpha(\bar{x}) = 0 \ \& \ g(\bar{x}) = \alpha(\bar{x})f(\bar{x}))\}.$
- $f(\bar{x}) \gtrsim g(\bar{x})$, если $f(\bar{x}) \geq g(\bar{x}) - \alpha(\bar{x})$ для некоторого $\alpha(\bar{x}) = \bar{o}(g(\bar{x}))$.
- $f(\bar{x}) \sim g(\bar{x})$, если $f(\bar{x}) \gtrsim g(\bar{x})$ и $f(\bar{x}) \lesssim g(\bar{x})$.

Невырожденные классы

- $\mathcal{J}(N) = \frac{\log_2 N}{\log_2 \log_2 N}.$
- Класс $Q(n) \subseteq P_2^{m(n)}(n)$ **невырожденный**, если $n + m(n) = \bar{o}(\mathcal{J}(|Q(n)|))$.
- Для невырожденного класса Q верна нижняя оценка на функцию Шеннона:

$$L^C(Q(n)) \gtrsim \mathcal{J}(|Q(n)|).$$

Инвариантные классы

Разложение Шеннона

- $x^\sigma = \begin{cases} x, & \sigma = 1, \\ \bar{x}, & \sigma = 0. \end{cases}$
- Для любой ФАЛ $f(\bar{x})$ верно

$$f(\bar{x}) = \bigvee_{\sigma_1, \dots, \sigma_k \in \{0,1\}} x_1^{\sigma_1} \dots x_k^{\sigma_k} f_{\sigma_1 \dots \sigma_k}(x_{k+1}, \dots, x_n),$$

где $f_{\sigma_1 \dots \sigma_k}(x_{k+1}, \dots, x_n) = f(\sigma_1, \dots, \sigma_k, x_{k+1}, \dots, x_n)$.

- Аналогичное разложение можно провести по любым переменным.

Конгруэнтные функции

- Функции f и g называются **конгруэнтными**, если одну из них можно получить из другой перестановкой переменных и добавлением/удалением фиктивных переменных.

Инвариантные классы

Квазиинвариантные классы

- Класс ФАЛ $Q(n)$ **квазиинвариантный**, если для достаточно больших n и любого $\sigma \in \{0, 1\}$, верно

$$f(x_1, \dots, x_n) \in Q(n) \rightarrow f(x_1, \dots, x_{n-1}, \sigma) \in Q(n-1).$$

- Пусть Q — класс ФАЛ. $\sigma_Q(n) = \frac{\log_2 |Q(n)|}{2^n}$.
- Если Q — квазиинвариантный класс, то $\exists \lim_{n \rightarrow \infty} \sigma_Q(n) = \sigma_Q \in [0, 1]$.
- Если класс Q квазиинвариантный, то все функции от достаточно большого числа переменных, используемые в разложении Шеннона по последним переменным любой функции $f \in Q$, тоже принадлежат Q .

Инвариантные классы

Инвариантные классы

- Класс ФАЛ Q **инвариантный**, если Q замкнут относительно
 - 1) добавления и изъятия фиктивных переменных;
 - 2) переименования переменных без отождествления;
 - 3) подстановки констант вместо переменных.
- Инвариантный класс является квазиинвариантным.
- Если функция f принадлежит инвариантному классу Q , то и все функции, конгруэнтные f , тоже принадлежат Q .
- Если класс Q инвариантный, то все функции используемые в любом разложении Шеннона любой функции $f \in Q$, тоже принадлежат Q .

Невырожденные и инвариантные классы

[1], семинар 12: в классе, 1 и 2 (1).

1. Выяснить, какие из следующих классов ФАЛ (операторов) являются невырожденными, и получить нижние мощностные оценки (НМО) функций Шеннона для сложности реализации ФАЛ из данных классов СФЭ в стандартном базисе.
2. Выделить среди классов ФАЛ из п. 1 квазиинвариантные и инвариантные классы, для которых найти пределы соответствующих им мощностных последовательностей.
 - Q — класс ФАЛ, равных 1 при $x_1 = 0$.

Невырожденные и инвариантные классы

Решение [1], семинар 12: в классе, 1 и 2 (1).

1. Первая половина вектора функции из $Q(n)$ содержит только единицы. Каждое из 2^{n-1} значений второй половины вектора функции задаётся произвольно (0 или 1). $|Q(n)| = 2^{2^{n-1}}$.
 - $\mathcal{J}(|Q(n)|) = \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|} = \frac{2^{n-1}}{n-1}$, класс невырожденный.
 - НМО: $L^C(Q(n)) \gtrsim \frac{1}{2} \cdot \frac{2^n}{n}$.
2. Класс квазиинвариантный, но не инвариантный (перестановка переменных выводит из класса).
 - $\sigma_Q = \lim_{n \rightarrow \infty} \frac{\log_2 |Q(n)|}{2^n} = \frac{1}{2}$.

Невырожденные и инвариантные классы

[1], семинар 12: в классе, 1 и 2 (2).

1. Выяснить, какие из следующих классов ФАЛ (операторов) являются невырожденными, и получить нижние мощностные оценки (НМО) функций Шеннона для сложности реализации ФАЛ из данных классов СФЭ в стандартном базисе.
2. Выделить среди классов ФАЛ из п. 1 квазиинвариантные и инвариантные классы, для которых найти пределы соответствующих им мощностных последовательностей.
 - Q — класс ФАЛ, симметричных по БП x_1, x_2, x_3 .

Невырожденные и инвариантные классы

Решение [1], семинар 12: в классе, 1 и 2 (2).

1. Если $f \in Q(n)$, то $f(\bar{x}) = \bigvee_{\sigma_4, \dots, \sigma_n \in \{0,1\}} x_4^{\sigma_4} \dots x_n^{\sigma_n} f_{\sigma_4 \dots \sigma_n}(x_1, x_2, x_3)$.
 - Для задания f нужно задать все 2^{n-3} функций $f_{\sigma_4 \dots \sigma_n}$. Все эти функции должны быть симметричными. Для задания симметричной функции от 3 переменных нужно задать её значения на наборах с 0, 1, 2, 3 единицами: всего 2^4 вариантов.
 - Для задания любой функции из $Q(n)$ достаточно выбрать 2^{n-3} функций, каждую 2^4 способами. $|Q(n)| = 2^{4 \cdot 2^{n-3}} = 2^{2^{n-1}}$.
 - $\mathcal{J}(|Q(n)|) = \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|} = \frac{2^{n-1}}{n-1}$, класс невырожденный.
 - НМО: $L^C(Q(n)) \gtrsim \frac{1}{2} \cdot \frac{2^n}{n}$.
2. Класс квазиинвариантный, но не инвариантный (перестановка переменных выводит из класса).
 - $\sigma_Q = \lim_{n \rightarrow \infty} \frac{\log_2 |Q(n)|}{2^n} = \frac{1}{2}$.

Невырожденные и инвариантные классы

[1], семинар 12: в классе, 1 и 2 (3).

1. Выяснить, какие из следующих классов ФАЛ (операторов) являются невырожденными, и получить нижние мощностные оценки (НМО) функций Шеннона для сложности реализации ФАЛ из данных классов СФЭ в стандартном базисе.
2. Выделить среди классов ФАЛ из п. 1 квазиинвариантные и инвариантные классы, для которых найти пределы соответствующих им мощностных последовательностей.
 - Q — класс ФАЛ, монотонных по БП x_1, x_2 .

Невырожденные и инвариантные классы

Решение [1], семинар 12: в классе, 1 и 2 (3).

1. Если $f \in Q(n)$, то $f(\bar{x}) = \bigvee_{\sigma_3, \dots, \sigma_n \in \{0,1\}} x_3^{\sigma_3} \dots x_n^{\sigma_n} f_{\sigma_3 \dots \sigma_n}(x_1, x_2)$.
 - Для задания f нужно задать все 2^{n-2} функций $f_{\sigma_3 \dots \sigma_n}$. Все эти функции должны быть монотонными. Монотонных функций от 2 переменных всего 6: 0, 1, x_1 , x_2 , $x_1 \cdot x_2$, $x_1 \vee x_2$.
 - Для задания любой функции из $Q(n)$ достаточно выбрать 2^{n-2} функций, каждую 6 способами. $|Q(n)| = 6^{2^{n-2}} = 2^{2^{n-2} \log_2 6}$.
 - $\mathcal{J}(|Q(n)|) = \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|} = \frac{2^{n-2} \log_2 6}{n-2 + \log_2 \log_2 6}$, класс невырожденный.
 - НМО: $L^C(Q(n)) \gtrsim \frac{\log_2 6}{4} \cdot \frac{2^n}{n}$.
2. Класс квазиинвариантный, но не инвариантный (перестановка переменных выводит из класса).
 - $\sigma_Q = \lim_{n \rightarrow \infty} \frac{\log_2 |Q(n)|}{2^n} = \frac{\log_2 6}{4}$.

Невырожденные и инвариантные классы

[1], семинар 12: в классе, 1 и 2 (4).

1. Выяснить, какие из следующих классов ФАЛ (операторов) являются невырожденными, и получить нижние мощностные оценки (НМО) функций Шеннона для сложности реализации ФАЛ из данных классов СФЭ в стандартном базисе.
2. Выделить среди классов ФАЛ из п. 1 квазиинвариантные и инвариантные классы, для которых найти пределы соответствующих им мощностных последовательностей.
 - Q — класс ФАЛ, равных 0 на наборах с чётным числом единиц.

Невырожденные и инвариантные классы

Решение [1], семинар 12: в классе, 1 и 2 (4).

1. Наборов с чётным числом единиц половина от всех наборов. Достаточно задать функцию f из $Q(n)$ на остальных 2^{n-1} наборах. $|Q(n)| = 2^{2^{n-1}}$.
 - $\mathcal{J}(|Q(n)|) = \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|} = \frac{2^{n-1}}{n-1}$, класс невырожденный.
 - НМО: $L^C(Q(n)) \gtrsim \frac{1}{2} \cdot \frac{2^n}{n}$.
2. Класс не квазиинвариантный: пусть $f \in Q(n)$. Выберем f так, чтобы было верно $f(0, \dots, 0, 1) = 1$ (это возможно в рамках $Q(n)$). Пусть $g(x_1, \dots, x_{n-1}) = f(x_1, \dots, x_{n-1}, 1)$. Тогда $g(0, \dots, 0) = f(0, \dots, 0, 1) = 1$, то есть $g \notin Q(n-1)$.

Невырожденные и инвариантные классы

[1], семинар 12: в классе, 1 и 2 (5).

1. Выяснить, какие из следующих классов ФАЛ (операторов) являются невырожденными, и получить нижние мощностные оценки (НМО) функций Шеннона для сложности реализации ФАЛ из данных классов СФЭ в стандартном базисе.
2. Выделить среди классов ФАЛ из п. 1 квазиинвариантные и инвариантные классы, для которых найти пределы соответствующих им мощностных последовательностей.
 - Q — класс линейных ФАЛ.

Невырожденные и инвариантные классы

Решение [1], семинар 12: в классе, 1 и 2 (5).

1. Общий вид линейной функции от n переменных:

$f(x_1, \dots, x_n) = a_0 \oplus a_1 x_1 \oplus \dots \oplus a_n x_n$. Каждый коэффициент может быть 0 или 1. $|Q(n)| = 2^{n+1}$.

- $\mathcal{J}(|Q(n)|) = \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|} = \frac{n+1}{\log_2(n+1)}$, класс вырожденный.
2. Класс инвариантный (так как замкнут и содержит 0 и 1).
- $\sigma_Q = \lim_{n \rightarrow \infty} \frac{\log_2 |Q(n)|}{2^n} = 0$.

Невырожденные и инвариантные классы

[1], семинар 12: в классе, 1 и 2 (6).

1. Выяснить, какие из следующих классов ФАЛ (операторов) являются невырожденными, и получить нижние мощностные оценки (НМО) функций Шеннона для сложности реализации ФАЛ из данных классов СФЭ в стандартном базисе.
2. Выделить среди классов ФАЛ из п. 1 квазиинвариантные и инвариантные классы, для которых найти пределы соответствующих им мощностных последовательностей.
 - Q — класс самодвойственных ФАЛ.

Невырожденные и инвариантные классы

Решение [1], семинар 12: в классе, 1 и 2 (6).

1. Вектор самодвойственной функции антисимметричен. Это значит, что достаточно задать первую половину вектора (2^{n-1} значений), а вторая будет определена автоматически. $|Q(n)| = 2^{2^{n-1}}$.
 - $\mathcal{J}(|Q(n)|) = \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|} = \frac{2^{n-1}}{n-1}$, класс невырожденный.
 - НМО: $L^C(Q(n)) \gtrsim \frac{1}{2} \cdot \frac{2^n}{n}$.
2. Класс не квазиинвариантный: рассмотрим любую функцию $x_1 \oplus \dots \oplus x_{2n+1}$ (это самодвойственная линейная функция: нечётное число переменных). При подстановке 0 вместо x_{2n+1} получим функцию $x_1 \oplus \dots \oplus x_{2n}$ (чётное число переменных, функция линейна и несамодвойственна).

Невырожденные и инвариантные классы

[1], семинар 12: в классе, 1 и 2 (7).

1. Выяснить, какие из следующих классов ФАЛ (операторов) являются невырожденными, и получить нижние мощностные оценки (НМО) функций Шеннона для сложности реализации ФАЛ из данных классов СФЭ в стандартном базисе.
2. Выделить среди классов ФАЛ из п. 1 квазиинвариантные и инвариантные классы, для которых найти пределы соответствующих им мощностных последовательностей.
 - Q — класс симметрических ФАЛ.

Невырожденные и инвариантные классы

Решение [1], семинар 12: в классе, 1 и 2 (7).

1. Чтобы задать функцию $f \in Q(n)$, необходимо и достаточно задать её значение для наборов с различным количеством единиц (т. е. на каждом слое булева куба). Всего разных количеств единиц $n + 1$. $|Q(n)| = 2^{n+1}$.
 - $\mathcal{J}(|Q(n)|) = \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|} = \frac{n+1}{\log_2(n+1)}$, класс вырожденный.
2. Класс квазиинвариантный, но не инвариантный (при добавлении фиктивных переменных симметричность нарушается).
 - $\sigma_Q = \lim_{n \rightarrow \infty} \frac{\log_2 |Q(n)|}{2^n} = 0$.

Невырожденные и инвариантные классы

[1], семинар 12: в классе, 1 (8).

Выяснить, какие из следующих классов ФАЛ (операторов) являются невырожденными, и получить нижние мощностные оценки (НМО) функций Шеннона для сложности реализации ФАЛ из данных классов СФЭ в стандартном базисе.

- Q — класс операторов вида $F = (f_1, f_2, f_3)$ таких, что $f_i \cdot f_j \equiv 0$ при $i \neq j$ и $f_1 \vee f_2 \vee f_3 \equiv 1$.

Невырожденные и инвариантные классы

Решение [1], семинар 12: в классе, 1 (8).

- На каждом наборе выбираем одну из 3 функций, которая примет значение 1. Получаем $|Q(n)| = 3^{2^n} = 2^{2^n \log_2 3}$.
- $\mathcal{J}(|Q(n)|) = \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|} = \frac{2^n \log_2 3}{n + \log_2 \log_2 3}$, класс невырожденный.
- НМО: $L^C(Q(n)) \gtrsim (\log_2 3) \cdot \frac{2^n}{n}$.

Инвариантные классы

Порождающий элемент и порождающее множество

- Все функции, которые можно получить из f подстановкой нулей и единиц вместо переменных, называются **подфункциями** f .
- Подфункция f , не совпадающая с f , называется собственной подфункцией f .
- **Порождающий элемент** инвариантного класса Q — это функция, не принадлежащая Q , все собственные подфункции которой принадлежат Q .
- **Порождающее множество** класса Q — это максимальное по включению множество попарно не конгруэнтных порождающих элементов Q .

Инвариантные классы

[1], семинар 12: в классе, 3.

Найти порождающее множество класса линейных ФАЛ.

Инвариантные классы

Решение [1], семинар 12: в классе, 3.

- Докажем, что $x^{\sigma_1}y^{\sigma_2}$, $x^{\sigma_1} \vee y^{\sigma_2}$, $\sigma_1, \sigma_2 \in \{0, 1\}$ — это все порождающие элементы L .
- Указанные функции — это все нелинейные функции от 2 переменных.
- Среди них 6 попарно не конгруэнтных функций. Порождающее множество: $\{xy, \bar{x}y, \bar{x}\bar{y}, x \vee y, \bar{x} \vee y, \bar{x} \vee \bar{y}\}$.
- Легко видеть, что все указанные функции нелинейны, но при подстановке нулей и единиц в них получаются линейные функции.
- Осталось показать, что других порождающих элементов у L нет. Будем действовать по аналогии с леммой о нелинейной функции.
- Пусть $f(x_1, \dots, x_n)$, $n \geq 3$ — нелинейная функция. Выберем в её полиноме Жегалкина любое нелинейное слагаемое с минимальным числом переменных.

Инвариантные классы

Решение [1], семинар 12: в классе, 3 (продолжение).

- Подставим вместо остальных переменных нули: все другие нелинейные слагаемые обнулятся.
- Подставляем единицы вместо оставшихся переменных, кроме двух. Получим нелинейную функцию от 2 переменных.
- Таким образом, у f есть собственная подфункция не из L , так что f — не порождающий элемент Q .

Домашнее задание

[1], семинар 12: на дом, 1 и 2 (1, 2, 3).

1. Исследовать на невырожденность и, в случае невырожденности, установить асимптотику НМО функции Шеннона $L^C(Q(n))$ для сложности реализации ФАЛ (операторов) от БП $x_1, \dots, x_n$ из класса Q .
2. Выделить среди классов ФАЛ из п. 1 квазиинвариантные и инвариантные классы, для которых найти пределы соответствующих им мощностных последовательностей.
 - 1) Q — класс ФАЛ, равных 1 при $x_1 = x_2 = 1$.
 - 2) Q — класс ФАЛ, монотонных по x_1 и антимонотонных по x_2 .
 - 3) Q — класс ФАЛ, у которых любая подфункция от БП x_1, x_2 принадлежит множеству $\{\bar{x}_1, \bar{x}_2, x_1, x_2, x_1 \vee x_2, x_1 \sim x_2\}$.

Домашнее задание

[1], семинар 12: на дом, 1 и 2 (4).

1. Исследовать на невырожденность и, в случае невырожденности, установить асимптотику НМО функции Шеннона $L^C(Q(n))$ для сложности реализации ФАЛ (операторов) от БП $x_1, \dots, x_n$ из класса Q .
2. Выделить среди классов ФАЛ из п. 1 квазиинвариантные и инвариантные классы, для которых найти пределы соответствующих им мощностных последовательностей.
 - Q — класс ФАЛ, симметричных по своим n , $n = 1, 2, \dots$ существенным БП с рабочими числами вида $a, a + 4, a + 8, \dots, a + 4k$, где $a \in \{0, 1, 2, 3\}$ и $k = \lfloor (n - a)/4 \rfloor$.

Замечание

Рабочие числа — количества единиц в наборах, на которых симметрическая функция выдаёт 1.

Домашнее задание

[1], семинар 12: на дом, 1 (5).

Исследовать на невырожденность и, в случае невырожденности, установить асимптотику НМО функции Шеннона $L^C(Q(n))$ для сложности реализации ФАЛ (операторов) от БП $x_1, \dots, x_n$ из класса Q .

- Q — класс операторов $F = (f_1, f_2)$ таких, что $f_2(\alpha_1, \dots, \alpha_n) = \bar{f}_1(\beta_1, \dots, \beta_n)$, где набор $\tilde{\beta}$ имеет номер на единицу больше, чем набор $\tilde{\alpha}$, если $\tilde{\alpha} \neq (1, \dots, 1)$, и равен нулевому набору в противном случае.

Домашнее задание

[1], семинар 12: на дом, 3.

Найти порождающее множество инвариантного класса ФАЛ, состоящего из констант и всех монотонных элементарных конъюнкций.

Список литературы

1. Ложкин С. А. Планы семинарских занятий и методические указания к ним на осенний семестр 2020–2021 уч. года. // Курс «Дополнительные главы дискретной математики и кибернетики» (часть III) — М., 2020.

Дополнительные главы дискретной математики и кибернетики

Семинар 13

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Асимптотически наилучший метод синтеза СФЭ

Мультиплексор

- **Мультиплексор** — это функция
 $\mu_k(x_1, \dots, x_k; y_0, \dots, y_{2^k-1}) = y_{i(x_1, \dots, x_k)}$, где i — это число, двоичным представлением которого является набор $(x_1, \dots, x_k)$.
- Переменные $x_1, \dots, x_k$ кодируют адрес ячейки памяти, а переменные $y_0, \dots, y_{2^k-1}$ представляют собой саму память. Мультиплексор берёт бит из памяти по указанному адресу.
- $\mu_k(x_1, \dots, x_k; y_0, \dots, y_{2^k-1}) = \bigvee_{\sigma_1, \dots, \sigma_k \in \{0,1\}} x_1^{\sigma_1} \dots x_k^{\sigma_k} y_{i(x_1, \dots, x_k)}.$
- Разложение Шеннона можно представить в виде

$$f(x_1, \dots, x_n) = \mu_k(x_1, \dots, x_k; (f_\sigma(x_{k+1}, \dots, x_n) \mid \sigma \in \{0,1\}^k)),$$

$$\text{где } f_\sigma(x_{k+1}, \dots, x_n) = f(\sigma, x_{k+1}, \dots, x_n).$$

Асимптотически наилучший метод синтеза СФЭ

Утверждение

$$L^C(\mu_n) \leq 4 \cdot 2^n - 4.$$

Доказательство

- Индукция по n . База: $\mu_1(x_1; y_0, y_1) = \bar{x}_1 y_0 \vee x_1 y_1$,
 $L^C(\mu_1) \leq 4 = 4 \cdot 2^1 - 4.$
- Пусть утверждение верно для n . Тогда

$$\begin{aligned} \mu_{n+1}(x_1, \dots, x_{n+1}; y_0, \dots, y_{2^n-1}, y_{2^n}, \dots, y_{2^{n+1}-1}) &= \\ &= \mu_1(x_1; \mu_n(x_2, \dots, x_{n+1}; y_0, \dots, y_{2^n-1}), \\ &\quad \mu_n(x_2, \dots, x_{n+1}; y_{2^n}, \dots, y_{2^{n+1}-1})). \end{aligned}$$

- $L^C(\mu_{n+1}) \leq 2L^C(\mu_n) + L^C(\mu_1) = 4 \cdot 2^{n+1} - 4.$



Асимптотически наилучший метод синтеза СФЭ

Теорема

$$L^C(P_2(n)) \lesssim \frac{2^n}{n}.$$

Доказательство

- Пусть $f(x_1, \dots, x_n) \in P_2(n)$. Произвольным образом разобьём набор переменных на 2 поднабора: набор x' размера q и набор x'' размера $n - q$. Параметр q будет выбран позже.
- Применяем разложение Шеннона (по x'')

$$f(x', x'') = \mu_{n-q}(x''; (f_{\sigma''}(x') \mid \sigma'' \in \{0, 1\}^{n-q})).$$

- Мультиплексор μ_{n-q} реализуется со сложностью $(4 \cdot 2^{n-q} - 4) \sim 2^{n-q+2}$ при $n - q \rightarrow \infty$.
- Для реализации всех 2^{n-q} функций $f_{\sigma''}(x')$ от q переменных, построим дизъюнктивно универсальное множество (ДУМ).

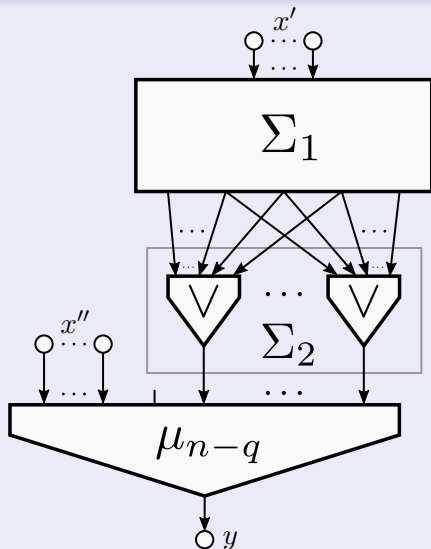
Асимптотически наилучший метод синтеза СФЭ

Доказательство (продолжение)

- ДУМ порядка q и ранга p — это такое множество функций из $P_2(q)$, что любую функцию из $P_2(q)$ можно получить в виде $g_1 \vee \dots \vee g_p$, где все g_i принадлежат данному ДУМ.
- Вводим произвольный параметр s (ширину).
- Разбиваем все 2^q входов для функций из $P_2(q)$ на полосы (подмножества) размера s (последняя может иметь меньший размер). Всего полос получится $p = \lceil \frac{2^q}{s} \rceil$.
- Определяем $G^{(i)}$ как множество всех функций, равных нулю на входах вне i -й полосы, $i = \overline{1, p}$.
- Очевидно, что любую функцию f из $P_2(q)$ можно получить в виде $f = g_1 \vee \dots \vee g_p$, где $g_i \in G^{(i)}$, $i = \overline{1, p}$.
- Тогда $G = G^{(1)} \cup \dots \cup G^{(p)}$ — ДУМ порядка q и ранга $p = \lceil \frac{2^q}{s} \rceil$.
- $|G^{(i)}| \leq 2^s$, $i = \overline{1, n}$, $|G| \leq p \cdot 2^s = \lceil \frac{2^q}{s} \rceil 2^s \sim \frac{2^{q+s}}{s}$ при $\frac{2^q}{s} \rightarrow \infty$.

Асимптотически наилучший метод синтеза СФЭ

Доказательство (продолжение)



- Реализуем функцию f с помощью трёх схем.
- Схема Σ_1 реализует все функции из G .
- Схема Σ_2 состоит из 2^{n-q} p -местных дизъюнкций и реализует все функции $f_{\sigma''}(x')$ с помощью функций из G .
- Схема μ_{n-q} реализует разложение Шеннона по переменным x'' .

Асимптотически наилучший метод синтеза СФЭ

Доказательство (продолжение)

- Функции из G реализуем при помощи совершенных ДНФ.
Поскольку каждая функция зависит от q переменных, сложность её СДНФ не превосходит $q \cdot 2^{q+1}$.
- Тогда $L^C(\Sigma_1) \leq q|G| \cdot 2^{q+1} \sim \frac{q}{s} \cdot 2^{2q+s+1}$ при $\frac{2^q}{s} \rightarrow \infty$.
- $L^C(\Sigma_2) = (p-1) \cdot 2^{n-q} = (\lceil \frac{2^q}{s} \rceil - 1) \cdot 2^{n-q} \sim \frac{2^n}{s}$ при $\frac{2^q}{s} \rightarrow \infty$.
- $L^C(\mu_{n-q}) \lesssim 2^{n-q+2}$ при $n-q \rightarrow \infty$.
- Итак, при $n-q, \frac{2^q}{s} \rightarrow \infty$ имеем
 - 1) $L^C(\Sigma_1) \lesssim \frac{q}{s} \cdot 2^{2q+s+1}$;
 - 2) $L^C(\Sigma_2) \sim \frac{2^n}{s}$;
 - 3) $L^C(\mu_{n-q}) \lesssim 2^{n-q+2}$.

Асимптотически наилучший метод синтеза СФЭ

Доказательство (продолжение)

- Подбираем параметры q, s . Условия:

1) $q \leq n, s \leq 2^q, n - q, \frac{2^q}{s} \rightarrow \infty$ при $n \rightarrow \infty$;

2) $\Sigma_1: \frac{q}{s} \cdot 2^{2q+s+1} \lesssim \frac{2^n}{n}$;

3) $\Sigma_2: \frac{2^n}{s} \lesssim \frac{2^n}{n}$.

4) $\mu_{n-q}: 2^{n-q+2} \lesssim \frac{2^n}{n}$;

5) $L^C(\Sigma_1) + L^C(\Sigma_2) + L^C(\mu_{n-q}) \lesssim \frac{2^n}{n}$

- Условия $q \leq n, s \leq 2^q$ при достаточно больших n выполняются автоматически за счёт $n - q, \frac{2^q}{s} \rightarrow \infty$ при $n \rightarrow \infty$.
- В силу (2) $s \lesssim n$. В силу (3) $s \gtrsim n$. Получаем $s \sim n$.
- Тогда $\frac{2^n}{s} \sim \frac{2^n}{n}$, (3) выполнено. Также $\frac{2^q}{s} \sim \frac{2^q}{n}, \frac{q}{s} \sim \frac{q}{n}$.
- Теперь для выполнения (5) достаточно $\frac{q}{s} \cdot 2^{2q+s+1} \sim \frac{q}{n} \cdot 2^{2q+s+1} = \bar{o}\left(\frac{2^n}{n}\right), 2^{n-q+2} = \bar{o}\left(\frac{2^n}{n}\right)$.

Асимптотически наилучший метод синтеза СФЭ

Доказательство (продолжение)

- Теперь условия таковы:

1) $n - q, \frac{2^q}{n} \rightarrow \infty$ при $n \rightarrow \infty$;

2) $\Sigma_1: \frac{q}{n} \cdot 2^{2q+s+1} = \bar{o}\left(\frac{2^n}{n}\right)$;

3) $L^C(\Sigma_2) \sim \frac{2^n}{n}$, $s \sim n$; отметим также $(p-1)2^{n-q} \sim \frac{2^n}{n}$;

4) $\mu_{n-q}: 2^{n-q+2} = \bar{o}\left(\frac{2^n}{n}\right)$;

- Из (2) ясно, что $q = \bar{o}(n)$. Для (4) выберем $q = \lceil 2 \log_2 n \rceil$.
- Тогда $2^{n-q+2} \leq 4 \cdot \frac{2^n}{n^2} = \bar{o}\left(\frac{2^n}{n}\right)$. Автоматически выполняется и (1).
- (2): $\frac{q}{n} \cdot 2^{2q+s+1} \lesssim \frac{2 \log_2 n}{n} \cdot 2^{4 \log_2 n + s + 3}$. Пусть $s = \lfloor n - 5 \log_2 n \rfloor \sim n$. Тогда $\frac{2 \log_2 n}{n} \cdot 2^{4 \log_2 n + s + 3} \lesssim \frac{2 \log_2 n}{n} \cdot 2^{n - \log_2 n + 3} = 16 \cdot \frac{2^n \cdot \log_2 n}{n^2}$.
- Получаем $L^C(\Sigma_1), L^C(\mu_{n-q}) = \bar{o}\left(\frac{2^n}{n}\right)$, $L^C(\Sigma_2) \sim \frac{2^n}{n}$. Тогда сложность всей схемы асимптотически равна $\frac{2^n}{n}$.



Модификация асимптотически наилучшего метода

[1], семинар 13: в классе, 1 (1).

Найти асимптотику функции Шеннона для класса ФАЛ или булевых операторов Q .

- Q — класс ФАЛ, равных 1 при $x_1 = 0$.

Модификация асимптотически наилучшего метода

Решение [1], семинар 13: в классе, 1 (1).

- Модифицируем асимптотически оптимальный метод синтеза.
- Выбираем x', x'' так, чтобы x_1 входило в x'' .
- Пусть $\sigma'' = (\sigma_1, \dots, \sigma_{n-q})$. Тогда при $\sigma_1 = 0$ верно $f_{\sigma''}(x') \equiv 1$.
- Схема Σ_2 должна реализовывать только $\frac{1}{2}2^{n-q}$ функций $f_{\sigma''}$, не равных 1. Плюс к этому требуется 2 элемента для реализации 1.
- Тогда $L(\Sigma_2) = \frac{1}{2}(p-1)2^{n-q} + 2 \sim \frac{1}{2} \cdot \frac{2^n}{n}$ при том же выборе параметров.
- Сложности схем Σ_1 и μ_{n-q} не изменяется.
- Получаем $L^C(\Sigma_1), L^C(\mu_{n-q}) = \bar{o}(\frac{2^n}{n})$. $L^C(\Sigma_2) \sim \frac{1}{2} \cdot \frac{2^n}{n}$.
- Нижняя оценка доказана на прошлом семинаре. Получаем $L^C(Q(n)) \sim \frac{1}{2} \cdot \frac{2^n}{n}$.

Квазиинвариантные классы

Квазиинвариантные классы

- Класс ФАЛ $Q(n)$ **квазиинвариантный**, если для достаточно больших n и любого $\sigma \in \{0, 1\}$, верно

$$f(x_1, \dots, x_n) \in Q(n) \rightarrow f(x_1, \dots, x_{n-1}, \sigma) \in Q(n-1).$$

- Пусть Q — класс ФАЛ. $\sigma_Q(n) = \frac{\log_2 |Q(n)|}{2^n}$.
- Если Q — квазиинвариантный класс, то $\exists \lim_{n \rightarrow \infty} \sigma_Q(n) = \sigma_Q \in [0, 1]$.

Утверждение

Пусть Q — квазиинвариантный класс.

- Если $\sigma_Q > 0$, то $L^C(Q(n)) \lesssim \sigma_Q \frac{2^n}{n}$.
- Если $\sigma_Q = 0$, то $L^C(Q(n)) = \overline{o}\left(\frac{2^n}{n}\right)$.

Квазиинвариантные классы

[1], семинар 13: в классе, 1 (2).

Найти асимптотику функции Шеннона для класса ФАЛ или булевых операторов Q .

- Q — класс ФАЛ, симметричных по БП x_1, x_2, x_3 .

Квазиинвариантные классы

Решение [1], семинар 13: в классе, 1 (2).

- На прошлом семинаре доказано, что Q — квазиинвариантный класс, $\sigma_Q = \frac{1}{2}$. Тогда $L^C(Q(n)) \lesssim \frac{1}{2} \cdot \frac{2^n}{n}$.
- Нижняя оценка доказана на прошлом семинаре.
- $L^C(Q(n)) \sim \frac{1}{2} \cdot \frac{2^n}{n}$.

Квазиинвариантные классы

[1], семинар 13: в классе, 1 (3).

Найти асимптотику функции Шеннона для класса ФАЛ или булевых операторов Q .

- Q — класс ФАЛ, монотонных по БП x_1, x_2 .

Квазиинвариантные классы

Решение [1], семинар 13: в классе, 1 (3).

- На прошлом семинаре доказано, что Q — квазиинвариантный класс, $\sigma_Q = \frac{\log_2 6}{4}$. Тогда $L^C(Q(n)) \lesssim \frac{\log_2 6}{4} \cdot \frac{2^n}{n}$.
- Нижняя оценка доказана на прошлом семинаре.
- $L^C(Q(n)) \sim \frac{\log_2 6}{4} \cdot \frac{2^n}{n}$.

Погружение в квазиинвариантный класс

[1], семинар 13: в классе, 1 (4).

Найти асимптотику функции Шеннона для класса ФАЛ или булевых операторов Q .

- Q — класс ФАЛ, равных 0 на наборах с чётным числом единиц.

Погружение в квазиинвариантный класс

Решение [1], семинар 13: в классе, 1 (4).

- Нижняя оценка доказана в прошлый раз: $L^C(Q(n)) \gtrsim \frac{1}{2} \cdot \frac{2^n}{n}$.
- Погрузим Q в квазиинвариантный класс. $|Q(n)| = 2^{2^{n-1}}$.
- Пусть $f \in Q$. При подстановке нуля вместо последней переменной мы получим функцию из Q . При подстановке единицы — ФАЛ, равную 0 на наборах с нечётным числом единиц.
- Пусть Q' — класс ФАЛ, равных 0 на наборах с нечётным числом единиц. $|Q'(n)| = 2^{2^{n-1}}$.
- Легко увидеть, что $Q'' = Q \cup Q'$ — квазиинвариантный класс.
- $2^{2^{n-1}} \leq |Q''(n)| \leq 2^{2^{n-1}+1}$. Тогда $\frac{1}{2} \leq \frac{\log_2 |Q''(n)|}{2^n} \leq \frac{2^{n-1}+1}{2^n}$. При $n \rightarrow \infty$ получим $\frac{1}{2} \leq \sigma_{Q''} \leq \frac{1}{2}$.
- Тогда $L^C(Q(n)) \leq L^C(Q''(n)) \lesssim \frac{1}{2} \cdot \frac{2^n}{n}$.
- $L^C(Q(n)) \sim \frac{1}{2} \cdot \frac{2^n}{n}$.

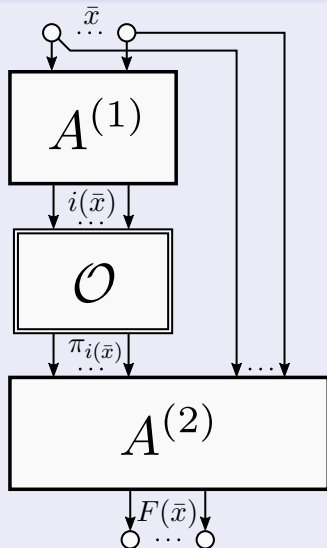
Принцип локального кодирования

Кодирование операторов

- Пусть Q — класс булевых операторов. Любой оператор $F \in Q(n)$ кодируется двоичным набором π_F длины $d(n)$.
- В π_F выделены куски кода π_i длины не более $\lambda(n)$, составленные из подряд идущих битов π_F . Куски кода могут пересекаться.
- Каждый кусок кода π_i задаётся своими координатами i : например, номером куска при равномерном разбиении кода.
- Свойство локальности кода: для вычисления $F(\alpha)$ достаточно знать лишь один кусок кода $\pi_{i(\alpha)}$.

Принцип локального кодирования

Доказательство (продолжение)



- Реализуем оператор F с помощью трёх схем.
- Схема $A^{(1)}$ — оператор кодирования. Он по входному набору $\bar{x}$ вычисляет координаты куска кода $i(\bar{x})$.
- Схема $\mathcal{O}$ — основной оператор. Он по координатам куска $i(\bar{x})$ вычисляет сам кусок $\pi_{i(\bar{x})}$.
- Схема $A^{(2)}$ — оператор декодирования. Он по куску кода $\pi_{i(\bar{x})}$ и части переменных из $\bar{x}$ вычисляет $F(\bar{x})$.

Принцип локального кодирования

Численные условия на код и куски

- Неизбыточность кодирования: $d(n) \sim \log_2 |Q(n)|$.
- Достаточно малые куски: $\log_2 \lambda(n) = \bar{o}(\log_2 d(n))$. Например, подойдёт $\lambda(n) \lesssim \log_2(d(n))$.
- Неизбыточность числа кусков и координат: число бит, кодирующих координаты, $c(n) = \log_2 \left(\frac{d(n)}{\lambda(n)} (1 + \bar{o}(1)) \right)$. Это условие выполняется, если области пересечения кусков малы, а координаты кодируются номерами кусков (или другим не менее экономным способом).
- Напомним, что $L^C(P_2(n)) \sim \frac{2^n}{n}$. Тогда $L^C(P_2^m(n)) \lesssim m \cdot \frac{2^n}{n}$.
- Тогда сложность схемы $\mathcal{O}$ асимптотически равна
$$\lambda(n) \cdot \frac{2^{c(n)}}{c(n)} = \frac{d(n)(1+\bar{o}(1))}{\log_2 d(n) - \log_2 \lambda(n) + \log_2(1+\bar{o}(1))} \sim \frac{d(n)}{\log_2 d(n)} \sim \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|}$$
 и совпадает с нижней оценкой $L^C(Q(n))$, если Q невырожденный.

Принцип локального кодирования

Итоговая сложность схемы

- Итак, при выполнении указанных выше условий
$$L^C(\mathcal{O}) \sim \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|}.$$
- При этом нужно построить $A^{(1)}$ и $A^{(2)}$ так, чтобы было верно
$$L^C(A^{(1)}), L^C(A^{(2)}) = \bar{O}\left(\frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|}\right).$$
- Тогда получим $L^C(Q(n)) \lesssim \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|}.$
- Если Q невырожденный, то $L^C(Q(n)) \sim \frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|}.$

Принцип локального кодирования

Общая идея принципа

- Пусть размер куска $\lambda(n) = 1$. Кодируем функцию из $Q(n)$ двоичным вектором. Бит вектора определяет значение функции на нескольких наборах, поэтому длина вектора меньше, чем 2^n .
- Фактическое вычисление значения функции происходит в операторе $\mathcal{O}$. Но на вход этому оператору подаются не входные переменные, а номер бита кода. Поскольку число этих бит меньше, чем 2^n , длина номера (число входов $\mathcal{O}$) будет меньше n .
- Это значит, что реализация $\mathcal{O}$ асимптотически оптимальным методом синтеза даст сложность, меньшую, чем $\frac{2^n}{n}$.
- Но результат оператора $\mathcal{O}$ содержит информацию о значении функции на нескольких наборах. Поэтому с помощью $A^{(2)}$ нужно извлечь значение именно на входном наборе.
- Кроме того, требуется оператор $A^{(1)}$, который определяет, в каком бите кода содержится информация о входном наборе.

Принцип локального кодирования

[1], семинар 13: в классе, 1 (6).

Найти асимптотику функции Шеннона для класса ФАЛ или булевых операторов Q .

- Q — класс самодвойственных ФАЛ.

Принцип локального кодирования

Решение [1], семинар 13: в классе, 1 (6).

- Используем принцип локального кодирования.
- Код функции $f \in Q(n)$ — это первая половина её вектора значений. $d(n) = 2^{n-1}$.
- Кусок кода — один бит. Этот бит равен значению $f(0, x_2, \dots, x_n)$ и противоположен $f(1, \bar{x}_2, \dots, \bar{x}_n)$. Таким образом, он содержит информацию о значении f на двух наборах.
- Координаты куска — его номер (представленный в виде $n - 1$ бит).
- Оператор $A^{(1)}$ выдаёт $(x_2 \oplus x_1, \dots, x_n \oplus x_1)$ — номер куска, хранящего значение функции на данном наборе с учётом антисимметричности вектора значений.
- $L^C(A^{(1)}) \leq 4n - 4$, так как $x \oplus y = \overline{x_1 x_2} \cdot (x_1 \vee x_2)$.

Принцип локального кодирования

Решение [1], семинар 13: в классе, 1 (6) (продолжение).

- Оператор $\mathcal{O}$ по $n - 1$ переменной выделяет бит вектора f с нужным номером. Реализуем асимптотически наилучшим методом. $L^C(\mathcal{O}) \sim \frac{1}{2} \cdot \frac{2^n}{n}$.
- Оператор $A^{(2)}$ прибавляет к результату $\mathcal{O}$ переменную x_1 .
- Если $x_1 = 0$, то кусок кода, выданный $\mathcal{O}$, уже является значением функции на нужном наборе.
- Если $x_1 = 1$, то выданный кусок кода содержит бит, противоположный значению функции.
- $L^C(A^{(2)}) = 4$.
- Итого $L^C(Q(n)) \lesssim \frac{1}{2} \cdot \frac{2^n}{n}$. Нижняя оценка доказана на прошлом занятии.

Принцип локального кодирования

[1], семинар 13: в классе, 1 (8).

Найти асимптотику функции Шеннона для класса ФАЛ или булевых операторов Q .

- Q — класс операторов вида $F = (f_1, f_2, f_3)$ таких, что $f_i \cdot f_j \equiv 0$ при $i \neq j$ и $f_1 \vee f_2 \vee f_3 \equiv 1$.

Принцип локального кодирования

Решение [1], семинар 13: в классе, 1 (8).

- Используем принцип локального кодирования.
- Каждый оператор из Q можно закодировать вектором длины 2^n с элементами из $\{1, 2, 3\}$ (каждый элемент относится к результату работы оператора на одном наборе).
- Выберем любое $q \leq n$. Пусть кусок кода кодирует 2^q элементов $\{1, 2, 3\}$, относящихся к 2^q наборов с одним и тем же значением первых $n - q$ переменных.
- Длина куска $\lambda(n) = \lceil \log_2 3 \cdot 2^q \rceil$ бит. Всего кусков 2^{n-q} .
- Длина кода $d(n) = 2^{n-q} \lceil \log_2 3 \cdot 2^q \rceil \sim \log_2 3 \cdot 2^n$ при $q \rightarrow \infty$.
- Координаты куска — его номер ($n - q$ бит). Его двоичная запись кодируется первыми $n - q$ переменными функции.

Принцип локального кодирования

Решение [1], семинар 13: в классе, 1 (8) (продолжение).

- Оператор $A^{(1)}$ — выделение первых $n - q$ переменных, которые и представляют номер куска. Сложность 0.
- Оператор $\mathcal{O}$ по $n - q$ битам выдаёт кусок длины $\lceil \log_2 3 \cdot 2^q \rceil$. Сложность реализации асимптотически наилучшим методом $\lceil \log_2 3 \cdot 2^q \rceil \frac{2^{n-q}}{n-q} \sim \log_2 3 \cdot \frac{2^n}{n-q}$ при $q \rightarrow \infty$.
- Оператор $A^{(2)}$ по куску кода длины $\lceil \log_2 3 \cdot 2^q \rceil$ и q последним входным переменным получает из куска кода результат работы оператора: 3 бита. Реализация асимптотически оптимальным методом даст сложность $3 \cdot \frac{2^{\lceil \log_2 3 \cdot 2^q \rceil + q}}{\lceil \log_2 3 \cdot 2^q \rceil + q}$.

Принцип локального кодирования

Решение [1], семинар 13: в классе, 1 (8) (продолжение).

- Чтобы сделать сложность $A^{(2)}$ малой, выберем любое достаточно малое, но растущее $q(n)$. Например, $q = \lfloor \log_2 \log_2 n \rfloor$.
- $L^C(A^{(1)}) = 0$.
- $L^C(\mathcal{O}) \sim \log_2 3 \cdot \frac{2^n}{n-q} \sim \log_2 3 \cdot \frac{2^n}{n}$.
- $L^C(A^{(2)}) \lesssim 6 \log_2 q \cdot \frac{(2^{2^q})^{\log_2 3}}{\log_2 3 \cdot 2^q} \leq \frac{12 \log_2 q}{\log_2 3} \cdot \frac{n^{\log_2 3}}{\log_2 n} = \bar{o}(\log_2 3 \cdot \frac{2^n}{n})$.
- Итог $L^C(Q(n)) \lesssim \log_2 3 \cdot \frac{2^n}{n}$. Нижняя оценка доказана на прошлом занятии.

Домашнее задание

[1], семинар 13: на дом, 1 (1, 2, 3, 5).

Найти асимптотику функции Шеннона для класса ФАЛ или булевых операторов Q .

1. Q — класс ФАЛ, равных 1 при $x_1 = x_2 = 1$.
2. Q — класс ФАЛ, монотонных по x_1 и антимонотонных по x_2 .
3. Q — класс ФАЛ, у которых любая подфункция от БП x_1, x_2 принадлежит множеству $\{\bar{x}_1, \bar{x}_2, x_1, x_2 x_1 \vee x_2, x_1 \sim x_2\}$.
5. Q — класс операторов $F = (f_1, f_2)$ таких, что $f_2(\alpha_1, \dots, \alpha_n) = \bar{f}_1(\beta_1, \dots, \beta_n)$, где набор $\tilde{\beta}$ имеет номер на единицу больше, чем набор $\tilde{\alpha}$, если $\tilde{\alpha} \neq (1, \dots, 1)$, и равен нулевому набору в противном случае.

Список литературы

1. Ложкин С. А. Планы семинарских занятий и методические указания к ним на осенний семестр 2020–2021 уч. года. // Курс «Дополнительные главы дискретной математики и кибернетики» (часть III) — М., 2020.

Дополнительные главы дискретной математики и кибернетики

Семинар 14

Савицкий Игорь Владимирович

факультет ВМК МГУ

осень 2020

Не всюду определённые функции

Не всюду определённые ФАЛ

- $f: \{0, 1\}^n \rightarrow \{0, 1, 2\}$ — **не всюду определённая ФАЛ**.
- Считаем 2 неопределённым значением. Если $f(\alpha) = 2$, считаем, что «не важно», что выдаёт функция на наборе α .
- $\delta(f) = f^{-1}(\{0, 1\})$ — **область определённости** f .
- $\hat{P}_2$ — множество всех не всюду определённых ФАЛ.
- $g \in P_2$ — **доопределение** $f \in \hat{P}_2$, если g совпадает с f на $\delta(f)$.
- СФЭ Σ реализует не всюду определённую ФАЛ f , если она реализует некоторое доопределение f .
- Для $f \in \hat{P}_2$ сложность $L^C(f)$ есть минимально возможная сложность доопределения f .

Не всюду определённые функции

[1], семинар 14: в классе, 1.

Найти сложность не всюду определённой ФАЛ f , $f \in \hat{P}_2(3)$, для которой $\tilde{\alpha}_f = (0001\ 1222)$.

Не всюду определённые функции

Решение [1], семинар 14: в классе, 1.

- $g(x_1, x_2, x_3) = x_2 \cdot x_3 \vee x_1$ имеет вектор (0001 1111). Это доопределение f сложности 2.
- Любое доопределение f существенно по всем трём переменным. Чтобы задействовать три переменные в схеме, необходимо хотя бы 2 элемента. Поэтому $L^C(f) \geq 2$.
- Получаем $L^C(f) = 2$.

Не всюду определённые функции

Не всюду определённые ФАЛ

- Пусть $Q \subseteq \hat{P}_2$. Функция Шеннона $L^C(Q(n)) = \max_{f \in Q(n)} L^C(f)$.
- $\hat{P}_2(n, t)$ — множество всех не всюду определённых ФАЛ от n переменных, определённых ровно на t наборах.

Теорема

Пусть $n \log_2^2 n = \bar{o}(t(n))$. Тогда

$$L^C(\hat{P}_2(n, t(n))) \sim \frac{t}{\log_2 t}.$$

Сложность реализации симметрической функции

Симметрические ФАЛ

- Функция $f \in P_2$ Называется симметрической, если она не меняется при любой перестановке переменных.
- Функция f симметрическая тогда и только тогда, когда её значение зависит только от числа единиц во входном наборе.
- $s_n^{\{i_1, \dots, i_k\}}$ — это симметрическая функция от n переменных, которая принимает значение 1 на наборах с $i_1, i_2, \dots$ или i_k единицами и значение 0 на всех остальных наборах.

Сложность реализации симметрической функции

Сумматор

- Сумматор S_n — это СФЭ, которая имеет $2n$ входов, $n + 1$ выходов и производит суммирование двух n -разрядных чисел.
- Сложение производится «в столбик».
- Полусумматор используется для сложения в младшем разряде: $x \oplus y = \overline{xy}(x \vee y)$, перенос реализуется xy . Сложность 4.
- Для сложения других разрядов используется сумматор трёх битов. $x \oplus y \oplus z$ реализуется с помощью двух полусумматоров как $(x \oplus y) \oplus z$. Получается 8 элементов.
- Перенос реализуется с помощью функции $m(x, y, z) = xy \vee (x \oplus y)z$. Достаточно добавить к схеме один элемент $\vee$ (xy и $(x \oplus y)z$ реализуются на элементах полусумматоров). Итого 9 элементов.
- Итого $L^C(S_n) \leq 9n - 5$.

Сложность реализации симметрической функции

Квазисумматор

- Квазисумматор S'_n — это СФЭ, которая работает так же, как и сумматор S_n , но только если каждое из входных чисел не превосходит 2^{n-1} . При нарушении этого условия на выходе может быть любой результат.
- При реализации квазисумматора при сложении старших разрядов на входе не может одновременно быть единицы в бите переноса и в бите какого-либо из разрядов.
- Поэтому достаточно использовать 5 элементов $\overline{xy}(x \vee y) \vee z$ для реализации суммы (xy реализуют перенос), вместо 9.
- Таким образом, сложность квазисумматора $L^C(S'_n) \leq 9n - 9$.

Сложность реализации симметрической функции

Счётчик

- Счётчик C_n — это СФЭ с n входами и $\lceil \log_2(n+1) \rceil$ выходами, которая вычисляет число единиц среди значений входных переменных

Утверждение

При $n = 2^k$ верно $L^C(C_n) \leq 11n$.

Доказательство

- Реализуем C_{2^k} с помощью двух счётчиков $C_{2^{k-1}}$, считающих число единиц в первой и во второй половине входов.
- Суммируем результаты двух счётчиков с помощью квазисумматора S'_k . Суммирование корректно, так как каждое из входных чисел квазисумматора не превосходит 2^{k-1} .

Сложность реализации симметрической функции

Доказательство (продолжение)

- Получаем $L^C(C_{2^k}) \leq 2L^C(C_{2^{k-1}}) + 9(k-1)$.
- Аналогичное неравенство можно записать для $L^C(C_{2^{k-1}})$ и т. д.
- C_2 реализуется полусумматором: $\overline{xy}(x \vee y)$ (старший разряд xy). Поэтому $L^C(C_2) \leq 4$.

$$\begin{aligned} L^C(C_{2^k}) &\leq 2L^C(C_{2^{k-1}}) + 9(k-1) \leq \\ &\leq 4L^C(C_{2^{k-2}}) + 2 \cdot 9(k-2) + 9(k-1) \leq \dots \\ &\dots \leq 2^{k-1}L^C(C_2) + 9 \sum_{i=1}^{k-1} 2^{i-1}(k-i) \leq \\ &\leq 2^{k+1} + 9 \sum_{i=1}^{k-1} 2^{i-1}(k-i) = 2^{k+1} + \frac{9}{2} \sum_{i=1}^{k-1} 2^i(k-i). \end{aligned}$$

Сложность реализации симметрической функции

Доказательство (продолжение)

$$\begin{aligned}\sum_{j=1}^{k-1} 2^j (k-j) &= \sum_{j=1}^{k-1} \sum_{t=1}^{k-j} 2^j = \sum_{t=1}^{k-1} \sum_{j=1}^{k-t} 2^j = \sum_{t=1}^{k-1} (2^{k-t+1} - 2) = \\ &= -2(k-1) + 2 \sum_{t=1}^{k-1} 2^{k-t} = -2(k-1) + 2 \sum_{t=1}^{k-1} 2^t = 2^{k+1} - 2k - 2.\end{aligned}$$

- Таким образом, $L^C(C_{2^k}) \leq 2^{k+1} + \frac{9}{2}(2^{k+1} - k - 2) \leq 11 \cdot 2^k$.

Сложность реализации симметрической функции

Утверждение

$$L^C(C_n) \leq 11(n + (\lceil \log_2(n+1) \rceil)^2).$$

Доказательство

- Разложим n по степеням двойки: $n = 2^{k_1} + \dots + 2^{k_l}$. Ясно, что можно выбрать $l \leq \lceil \log_2(n+1) \rceil$.
- Реализуем C_n с помощью l счётчиков $C_{2^{k_1}}, \dots, C_{2^{k_l}}$. Каждый счётчик считает число единиц в своей части бит.
- Сложность всех счётчиков
$$L^C(C_{2^{k_1}}) + \dots + L^C(C_{2^{k_l}}) \leq 11(2^{k_1} + \dots + 2^{k_l}) = 11n.$$
- Суммируем результаты всех счётчиков с помощью $l-1$ сумматоров, у каждого по $k = \lceil \log_2(n+1) \rceil$ входов.
- $$L^C(C_n) \leq 11n + (9k - 5)(l - 1) \leq 11(n + (\lceil \log_2(n+1) \rceil)^2).$$

Сложность реализации симметрической функции

Утверждение

Пусть $f \in P_2(n)$ — симметрическая функция. Тогда

$$L^C(f) \lesssim 11n.$$

Доказательство

- Симметрическая функция зависит только от числа единиц в своих входах.
- Это значит, что $f(x_1, \dots, x_n) = g(C_n(x_1, \dots, x_n))$, где $g \in P_2(\lceil \log_2(n+1) \rceil)$.
- Реализуем g асимптотически наилучшим методом синтеза.
$$L^C(g) \lesssim \frac{2^{\lceil \log_2(n+1) \rceil}}{\lceil \log_2(n+1) \rceil} \lesssim \frac{2^{\log_2(n+1)+1}}{\log_2 n} \sim \frac{2(n+1)}{\log_2 n} = \bar{o}(n).$$
- $L^C(C_n) \lesssim 11n$. Тогда $L^C(f) \lesssim 11n + \bar{o}(n) \sim 11n$.

Оценки для факториала и биномиальных коэффициентов

Формула Стирлинга

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n.$$

Оценка для биномиального коэффициента

- Рассмотрим наибольший биномиальный коэффициент $C_n^{\lfloor n/2 \rfloor}$.
- $C_n^{\lfloor n/2 \rfloor} = C_n^{\lceil n/2 \rceil} = \frac{n!}{\lfloor n/2 \rfloor! \lceil n/2 \rceil!}.$
- При чётных n имеем $C_n^{\lfloor n/2 \rfloor} = \frac{n!}{(n/2)!^2} \sim \sqrt{\frac{2}{\pi}} \frac{2^n}{\sqrt{n}}.$

Синтез схем с помощью не всюду определённых ФАЛ

[1], семинар 14: в классе, 2.

Найти асимптотику функции Шеннона $L^C(Q(n))$ для сложности ФАЛ от переменных $x_1, \dots, x_n$ из класса $Q(n)$, включающего в себя все те ФАЛ, которые обращаются в ноль на наборах куба B^n , имеющих не меньше $n/2$ единиц.

Синтез схем с помощью не всюду определённых ФАЛ

Решение [1], семинар 14: в классе, 2.

- Обозначим через $t(n)$ число наборов из $\{0, 1\}^n$, имеющих меньше $n/2$ единиц. Покажем, что $t(n) \sim 2^{n-1}$.
- Если n нечётно, то число слоёв в булевом кубе $\{0, 1\}^n$ чётно, и нижняя половина слоёв имеет меньше $n/2$ единиц. $t(n) = 2^{n-1}$.
- Если n чётно, то число слоёв $\{0, 1\}^n$ нечётно. Меньше $n/2$ единиц имеют только наборы, лежащие ниже среднего слоя. Размер среднего слоя есть $C_n^{n/2}$.
- Тогда $2^{n-1} - C_n^{n/2} \leq t(n) \leq 2^{n-1}$. Но $C_n^{n/2} \sim \sqrt{\frac{2}{\pi}} \frac{2^n}{\sqrt{n}} = o(2^{n-1})$.
- Получаем $t(n) \sim 2^{n-1}$.
- $|Q(n)| = 2^{t(n)}$. Тогда $\frac{\log_2 |Q(n)|}{\log_2 \log_2 |Q(n)|} = \frac{t(n)}{\log_2 t(n)} \sim \frac{2^{n-1}}{n}$.
- $n = o(\frac{2^{n-1}}{n})$, класс Q невырожденный. Тогда $L^C(Q(n)) \gtrsim \frac{2^{n-1}}{n}$.

Синтез схем с помощью не всюду определённых ФАЛ

Решение [1], семинар 14: в классе, 2 (продолжение).

- Пусть теперь $f \in Q(n)$. Представим f в виде

$$f(\bar{x}) = g(\bar{x}) \cdot s_n^{\{0, \dots, \lfloor (n-1)/2 \rfloor\}}(\bar{x}).$$

- Здесь $g(\bar{x}) \in \hat{P}_2(n, t(n))$ совпадает с f на наборах, имеющих меньше $n/2$ единиц, и не определена на остальных наборах.
- $s_n^{\{0, \dots, \lfloor (n-1)/2 \rfloor\}}$ равна 1 на наборах с менее, чем $n/2$ единицами.
- Конъюнкция неопределённого значения с нулём даёт ноль, так как на самом деле вместо неопределённого значения будет подставлено некое неизвестное определённое значение (0 или 1).
- Поскольку $n \log_2^2 n = \bar{o}(2^{n-1})$, по теореме о не всюду определённых функциях $L^C(g) \lesssim \frac{2^{n-1}}{n}$.
- $L^C(s_n^{\{0, \dots, \lfloor (n-1)/2 \rfloor\}}) \lesssim 11n = \bar{o}(\frac{2^{n-1}}{n})$. Итого $L^C(f) \sim \frac{2^{n-1}}{n}$.

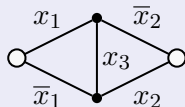
Контактные схемы

Двухполюсные контактные схемы

- Двухполюсная контактная схема — это неориентированный мультиграф, у которого две вершины выделены (полюса), а все рёбра (контакты) помечены символами переменных или их отрицаний.
- Пусть фиксировано значение каждой переменной. Тогда считаем, что контакты, пометка которых даёт 1, проводят сигнал, а остальные контакты — не проводят.
- Если сигнал может пройти от одного полюса до другого, то на данном наборе схема выдаёт значение 1; иначе — 0.
- Таким образом, контактная схема реализует булеву функцию.

Контактные схемы

Пример

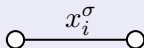


- Указанная схема реализует функцию

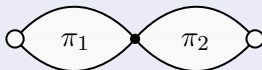
$$x_2\bar{x}_2 \vee x_1x_3x_2 \vee \bar{x}_1x_2 \vee \bar{x}_1x_3\bar{x}_2 = x_3 \vee (x_1 \oplus x_2).$$

Определение π -схемы

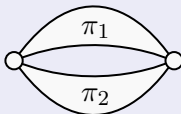
- Схема из одного контакта является π -схемой:



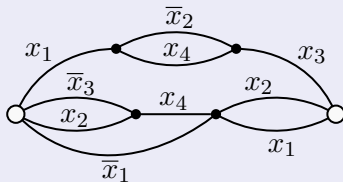
- Две последовательно соединённые π -схемы образуют π -схему:



- Две параллельно соединённые π -схемы образуют π -схему:



Пример



- Схема реализует функцию

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

Сложность π -схем

Функционал сложности для π -схем

- Сложность π -схемы — это число контактов в ней.
- Пусть $f \in P_2$. Сложность реализации f в классе π -схем $L^\pi(f)$ — это наименьшая возможная сложность π -схемы, реализующей функцию f .
- $l_n(x_1, \dots, x_n) = x_1 \oplus \dots \oplus x_n$; $\bar{l}_n(x_1, \dots, x_n) = x_1 \oplus \dots \oplus x_n \oplus 1$.

Утверждение

1. При $n = 2^k$, $k \in \mathbb{N}_0$ верно $L^\pi(l_n) = L^\pi(\bar{l}_n) = n^2$.
2. При любом n верно $n^2 \leq L^\pi(l_n), L^\pi(\bar{l}_n) \leq 4n^2$.

Сложность π -схем

Множества наборов

- Характеристическое множество N_f функции f — это множество наборов из $\{0, 1\}^n$, на которых f принимает значение 1.
- $\overline{N}_f = \{0, 1\}^n \setminus N_f$.
- Пусть $\alpha, \beta \in \{0, 1\}^n$. $\rho(\alpha, \beta)$ — это число отличающихся разрядов у α и β . Если $\rho(\alpha, \beta) = 1$, то эти наборы называются соседними.
- $|\alpha|$ — число единиц в α . $B_k^n = \{\alpha \in \{0, 1\}^n : |\alpha| = k\}$. $|B_k^n| = C_n^k$.
- Пусть $N_1, N_2 \subseteq \{0, 1\}^n$, $N_1 \cap N_2 = \emptyset$. Тогда

$$R(N_1, N_2) = \{(\alpha, \beta) \mid \alpha \in N_1, \beta \in N_2, \rho(\alpha, \beta) = 1\}.$$

Теорема (Храпченко)

Пусть $f \in P_2(n)$, $N_1 \subseteq N_f$, $N_2 \subseteq \overline{N}_f$. Тогда $L^\pi(f) \geq \frac{|R(N_1, N_2)|^2}{|N_1| \cdot |N_2|}$.

Сложность π -схем

[1], семинар 14: в классе, 3.

Доказать, что

$$12 \leq L^\pi(s_4^2) \leq 16,$$

где s_n^I — симметрическая ФАЛ от n БП, «рабочие» числа которой составляют множество I , $I \subseteq [0, n]$.

Решение [1], семинар 14: в классе, 3.

- Верхняя оценка. Функция s_4^2 принимает значение 1 тогда и только тогда, когда на входе 2 единицы. Это условие можно разложить следующим образом:

$$\begin{aligned}s_4^2(x_1, x_2, x_3, x_4) &= \\ &= s_2^0(x_1, x_2)s_2^2(x_3, x_4) \vee s_2^1(x_1, x_2)s_2^1(x_3, x_4) \vee s_2^2(x_1, x_2)s_2^0(x_3, x_4)\end{aligned}$$

- $s_2^0(x_1, x_2) = \bar{x}_1\bar{x}_2$, $s_2^1(x_1, x_2) = x_1 \oplus x_2$, $s_2^2(x_1, x_2) = x_1x_2$.
- $L^\pi(s_2^0) = L^\pi(s_2^2) = 2$. $x_1 \oplus x_2 = x_1\bar{x}_2 \vee \bar{x}_1x_2$, $L^\pi(s_2^1) = 4$.
- Получаем верхнюю оценку: $L^\pi(s_4^2) \leq 2(2 + 4 + 2) = 16$.

Решение [1], семинар 14: в классе, 3 (продолжение).

- Нижняя оценка. Нужно выбрать $N_1 \subseteq N_{s_4^2}$, $N_2 \subseteq \overline{N}_{s_4^2}$ так, чтобы между ними было как можно больше соседних наборов, но сами множества были как можно меньше.
- Пусть $N_1 = B_2^4$, $N_2 = B_1^4 \cup B_3^4$. Каждый набор из N_1 имеет 4 соседних набора из N_2 . Значит, $|R(N_1, N_2)| = 4|N_1|$.
- $|N_1| = C_4^2 = 6$. $|N_2| = 2C_4^1 = 8$.
- По теореме Храпченко

$$L^\pi(s_4^2) \geq \frac{|R(N_1, N_2)|^2}{|N_1| \cdot |N_2|} = \frac{16|N_1|^2}{|N_1| \cdot |N_2|} = \frac{16|N_1|}{|N_2|} = 12.$$

Сложность π -схем

[1], семинар 14: в классе, 4.

Доказать, что

$$L^{\pi}((x_1 \oplus \dots \oplus x_k)(x_{k+1} \oplus \dots \oplus x_n)) \geq n^2/2.$$

Сложность π -схем

Решение [1], семинар 14: в классе, 4.

- Обозначим

$$l'(\bar{x}) = (x_1 \oplus \dots \oplus x_k), \quad l''(\bar{x}) = (x_{k+1} \oplus \dots \oplus x_n), \quad f(\bar{x}) = l'(\bar{x})l''(\bar{x}).$$

- Заметим, что $\{0, 1\}^n$ разделяется на 4 непересекающихся подмножества

$$N_{ab} = \{\bar{\alpha} \in \{0, 1\}^n : l'(\bar{\alpha}) = a, l''(\bar{\alpha}) = b\}, \quad a, b \in \{0, 1\}.$$

- Неконстантная линейная функция принимает значение 1 ровно на половине всех своих наборов. Поэтому $|N_{l_k^\sigma}| = 2^{k-1}$.
- Выбор набора из N_{ab} : выбираем значения первых k переменных из $|N_{l_k^a}|$, а последних $n - k$ переменных — из $|N_{l_{n-k}^b}|$.
- Поэтому $|N_{ab}| = |N_{l_k^a}| \cdot |N_{l_{n-k}^b}| = 2^{k-1} \cdot 2^{n-k-1} = 2^{n-2}$.

Сложность π -схем

Решение [1], семинар 14: в классе, 4 (продолжение).

- Легко видеть, что $N_f = N_{11}$, а все наборы, соседние с наборами из N_f , лежат в N_{01} и N_{10} .
- Выбираем $N_1 = N_{11}$, $N_2 = N_{01} \cup N_{10}$.
- Каждый набор из N_1 имеет n соседних наборов из N_2 , поэтому $|R(N_1, N_2)| = n|N_1|$.
- $|N_1| = 2^{n-2}$, $|N_2| = 2^{n-1}$.
- По теореме Храпченко

$$L^\pi(f) \geq \frac{|R(N_1, N_2)|^2}{|N_1| \cdot |N_2|} = \frac{n^2|N_1|^2}{|N_1| \cdot |N_2|} = \frac{n^2|N_1|}{|N_2|} = \frac{n^2}{2}.$$

Домашнее задание

[1], семинар 13: на дом, 1.

Найти сложность не всюду определённой ФАЛ f , $f \in \hat{P}_2(3)$, для которой $\tilde{\alpha}_f = (0111\ 2221)$.

[1], семинар 13: на дом, 2.

Найти асимптотику функции Шеннона $L^C(Q(n))$ для сложности ФАЛ от переменных $x_1, \dots, x_n$ из класса $Q(n)$, включающего в себя все те ФАЛ, которые обращаются в ноль на наборах куба B^n , имеющих не равное i , $2 \leq i \leq n - 2$ число единиц.

Домашнее задание

[1], семинар 13: на дом, 3.

Доказать, что $63 \leq L^\pi(s_8^{\{2,4,6\}}) \leq 80$.

[1], семинар 13: на дом, 4.

Доказать, что

$$L^\pi((x_1 \oplus \dots \oplus x_k)(x_{k+1} \oplus \dots \oplus x_s)(x_{s+1} \oplus \dots \oplus x_n)) \geq n^2/3,$$

где $1 \leq k < s < n$.

Приложение

- Докажем утверждение из [3] (сноска на с. 10).

Утверждение

Пусть $t(n)$, $n \in \mathbb{N}$ — последовательность натуральных чисел, причём, начиная с некоторого n , верно $t(n) \geq 2$. Тогда при $n \rightarrow \infty$

$$n \log_2 n = \bar{o}(t(n)) \iff n = \bar{o}\left(\frac{t(n)}{\log_2 t(n)}\right).$$

Доказательство

- $\Leftarrow$. Пусть $n = \bar{o}\left(\frac{t(n)}{\log_2 t(n)}\right)$. Тогда $n \log_2 t(n) = \bar{o}(t(n))$.
- Ясно, что при достаточно больших n верно $n \leq t(n)$.
- Тогда $n \log_2 n \leq n \log_2 t(n) = \bar{o}(t(n))$.

Приложение

Доказательство (продолжение)

- $\Rightarrow$. Рассмотрим функцию $\varphi(x) = \frac{x}{\log_2 x}$ на области $x > 1$.
- $\varphi'(x) = \frac{\log_2 x - 1}{\log_2^2 x} \geq 0$ при $x \geq 2$. $\frac{x}{\log_2 x}$ неубывает при $x \geq 2$.
- Пусть $n \log_2 n = \bar{o}(t(n))$. Это значит, что $\frac{n \log_2 n}{t(n)} \xrightarrow{n \rightarrow \infty} 0$.
- $(\forall A > 0)(\exists n_0)(\forall n \geq n_0) \frac{n \log_2 n}{t(n)} \leq \frac{1}{A}$, то есть $An \log_2 n \leq t(n)$.
- Выберем $n_0 = n_0(c)$ настолько большим, чтобы было верно также $(\forall n \geq n_0)(An \log_2 n \geq 2 \ \& \ t(n) \geq 2)$.
- Подействуем функцией $\frac{x}{\log_2 x}$ на обе части неравенства $An \log_2 n \leq t(n)$. В силу монотонности $\frac{x}{\log_2 x}$ при $x \geq 2$ неравенство сохранится:

$$\frac{An \log_2 n}{\log_2 A + \log_2 n + \log_2 \log_2 n} \leq \frac{t(n)}{\log_2 t(n)} \quad \text{при } n \geq n_0.$$

Доказательство (продолжение)

$$\frac{An \log_2 n}{\log_2 A + \log_2 n + \log_2 \log_2 n} \leq \frac{t(n)}{\log_2 t(n)} \quad \text{при } n \geq n_0.$$

- Преобразуем:

$$\frac{\log_2 n}{\log_2 A + \log_2 n + \log_2 \log_2 n} \cdot \frac{n \log_2 t(n)}{t(n)} \leq \frac{1}{A} \quad \text{при } n \geq n_0.$$

- Легко доказать следующее свойство верхнего предела. Если

$$\exists \lim_{n \rightarrow \infty} a(n) = a, \text{ то } \overline{\lim}_{n \rightarrow \infty} (a(n)b(n)) = a \overline{\lim}_{n \rightarrow \infty} b(n).$$

- $\lim_{n \rightarrow \infty} \frac{\log_2 n}{\log_2 A + \log_2 n + \log_2 \log_2 n} = 1.$

- Тогда $\overline{\lim}_{n \rightarrow \infty} \left(\frac{\log_2 n}{\log_2 A + \log_2 n + \log_2 \log_2 n} \cdot \frac{n \log_2 t(n)}{t(n)} \right) = \overline{\lim}_{n \rightarrow \infty} \frac{n \log_2 t(n)}{t(n)}.$

Приложение

Доказательство (продолжение)

- Из

$$\frac{\log_2 n}{\log_2 A + \log_2 n + \log_2 \log_2 n} \cdot \frac{n \log_2 t(n)}{t(n)} \leq \frac{1}{A} \quad \text{при } n \geq n_0.$$

следует

$$\overline{\lim}_{n \rightarrow \infty} \left(\frac{\log_2 n}{\log_2 A + \log_2 n + \log_2 \log_2 n} \cdot \frac{n \log_2 t(n)}{t(n)} \right) = \overline{\lim}_{n \rightarrow \infty} \frac{n \log_2 t(n)}{t(n)} \leq \frac{1}{A}.$$

- Это верно для всех $A > 0$. При $A \rightarrow \infty$ имеем $\overline{\lim}_{n \rightarrow \infty} \frac{n \log_2 t(n)}{t(n)} = 0$.
- Но $0 \leq \underline{\lim}_{n \rightarrow \infty} \frac{n \log_2 t(n)}{t(n)} \leq \overline{\lim}_{n \rightarrow \infty} \frac{n \log_2 t(n)}{t(n)}$.
- Это значит $\exists \lim_{n \rightarrow \infty} \frac{n \log_2 t(n)}{t(n)} = \underline{\lim}_{n \rightarrow \infty} \frac{n \log_2 t(n)}{t(n)} = \overline{\lim}_{n \rightarrow \infty} \frac{n \log_2 t(n)}{t(n)} = 0$.
- То есть $n = \bar{o} \left(\frac{t(n)}{\log_2 t(n)} \right)$.

Список литературы

1. Ложкин С. А. Планы семинарских занятий и методические указания к ним на осенний семестр 2020–2021 уч. года. // Курс «Дополнительные главы дискретной математики и кибернетики» (часть III) — М., 2020.
2. Алексеев В. Б., Ложкин С. А. Элементы теории графов, схем и автоматов (учебное пособие для студентов). — М.: Издательский отдел факультета ВМиК МГУ, 2000. — 58 с.
3. Ложкин С. А. Дополнительные главы кибернетики. — М., 2019. — 18 с.