

# Элементы Теории Алгоритмов

## 1.1 Понятие алгоритма в интуитивном смысле слова

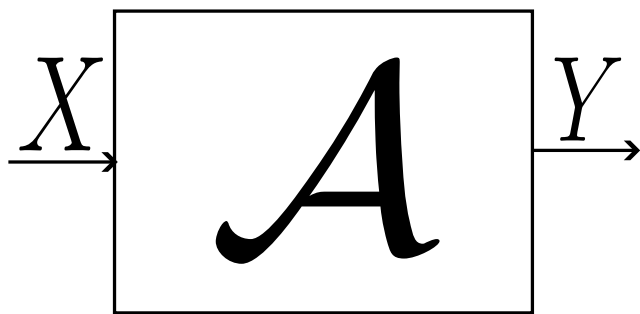


Рис. 1.1: Команда

$$\mathcal{A} : X \rightarrow Y$$

Признаки алгоритма:

- Признак детерминизированности (нет выбора в алгоритме)
- Признак массовости (работает для всех входных данных одного типа , например, квадратных уравнений)
- Признак результативности (ожидается какой-то результат)

**Определение 1.** алгоритм  $A$  применим к элементу  $x$ . (То есть останавливается за  $n$  шагов)

$$(x \in X)(!A(x))$$

**Определение 2.**  $\neg!A(x)$  - алгоритм  $A$  не применим к  $x$ .

**Определение 3.** Конструктивный объект - слово в конечном алфавите.

**Определение 4.** Вербальная, или словарная, функция - это

$$f : V^* \rightarrow W^*$$

Вербальная функция  $(V, W)$ .

**Определение 5.** Алгоритм можно записать так:

$$\mathcal{A} : V^* \rightarrow W^*$$

**Определение 6.** Функция  $f : V^* \rightarrow W^*$  называется вычислимой в интуитивном смысле слова, если существует алгоритм  $\mathcal{A}_f : V^* \rightarrow W^*$  такой, что

$$(\forall x \in V^*)((\neg!\mathcal{A}_f(x) \iff x \in D(f)) \& (\mathcal{A}_f(x) = f(x)))$$

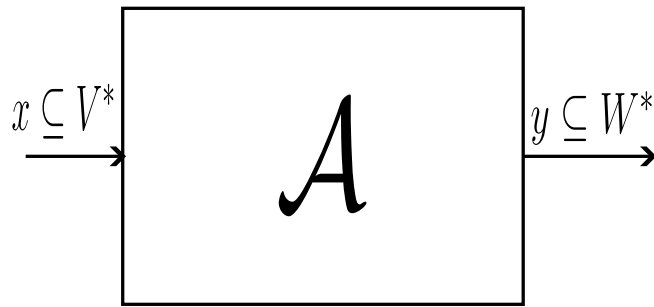


Рис. 1.2: Автомат

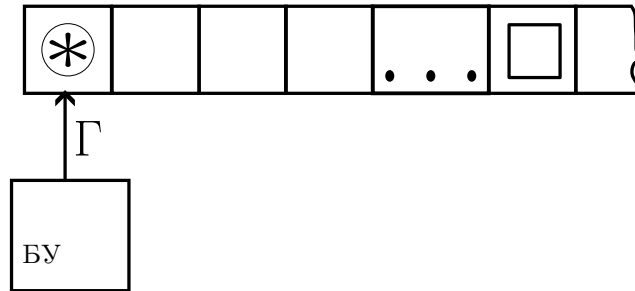


Рис. 1.3: Машина Тьюринга

## 1.2 Машина Тьюринга.

Команды следующего формата:

$$qa \rightarrow rb, \begin{Bmatrix} S \\ L \\ R \end{Bmatrix}; q, r \in Q; a, b \in V \cup \{*, \square\}$$

**Заметка.** Мы считаем, что у нас не может быть команд с одинаковыми левыми частями.

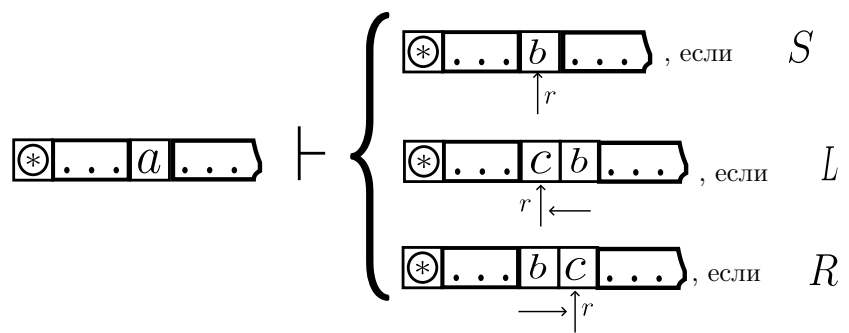
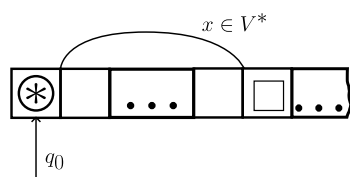
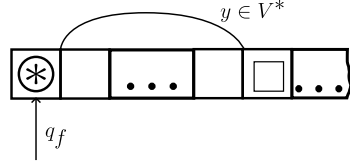


Рис. 1.4: Что к чему

Начальная конфигурация:



Заключительная конфигурация:



Пример программы:

$$\begin{aligned}
q_0 \otimes &\rightarrow q_0 \otimes, R \\
q_0 a &\rightarrow q_0 a, R \\
q_0 b &\rightarrow q_0 b, R \\
q_0 c &\rightarrow q_1 c, R \\
q_1 a &\rightarrow q_2 a, R \\
q_1 b &\rightarrow q_0 b, R \\
q_1 c &\rightarrow q_1 c, R \\
q_2 a &\rightarrow q_0 a, R \\
q_2 b &\rightarrow q_3 b, R \\
q_2 c &\rightarrow q_1 c, R \\
q_3 \alpha &\rightarrow q_3 \alpha, R // \alpha \in \{a, b, c\} \\
q_3 \square &\rightarrow q_4 \square, R \\
q_i \square &\rightarrow q_5 \square, L // i = 0, 1, 2 \\
q_4 \otimes &\rightarrow q_f 1, L \\
q_5 \alpha &\rightarrow q_5 \square, L \\
q_5 \otimes &\rightarrow q_5 *, R \\
q_5 \square &\rightarrow q_f 0, L
\end{aligned}$$

$$f(x) = \begin{cases} 1, & \text{если } cab \sqsubseteq x \in \{a, b, c\} \\ 0 & \text{иначе} \end{cases}$$

**Определение 7.** Машина Тьюринга (МТ):

$$\mathcal{J} = (V, Q, q_0, q_f, *, \square, S, L, R, \delta)$$

Конфигурация МТ:

$$C = (q, x, ay),$$

где  $q \in Q$ , а  $x, y \in (V \cup \{*, \square\})^*$ ,  $a \in V \cup \{*, \square\}$

Мы полагаем, что

$$(q, x, ay) \vdash_{\mathcal{J}} \begin{cases} (r, x, by), & \text{если } qa \rightarrow rb, S \in \delta \\ (r, x', cby), & \text{где } x'c = x, \text{ если } qa \rightarrow rb, L \in \delta \\ (r, xb, dy'), & \text{где } y = dy', \text{ если } qa \rightarrow rb, R \in \delta \end{cases}$$

**Определение 8.** Вывод на множестве конфигураций:

$K_0, K_1, \dots, K_n$ , где  $(\forall i \geq 0)(K_i \vdash K_{i+1}$ , если  $K_{i+1}$  определен в последовательности)

$$K \vdash_{\mathcal{J}}^* K', \text{ если существует вывод } K = K_0 \vdash K_1 \vdash \dots \vdash K_n = K'$$

Дано:

Начальная конфигурация  $C_0 = (q_0, \lambda, \otimes x \square)$ , где  $x \in V^*$

Конечная конфигурация  $C_f = (q_f, \lambda, \otimes y \square)$ , где  $y \in V^*$

**Определение 9.** Машина Тьюринга применима к слову  $x$ , то есть

$$\begin{aligned} & !\mathcal{T}(x) \Leftarrow \\ & \Leftarrow C_0 = (q_0, \lambda, \otimes x \square) \vdash^* C_f = (q_f, \lambda, \otimes y \square); \end{aligned}$$

при этом  $y \Leftarrow \mathcal{T}(x)$

При этом если не применимо к машине тьюринга данное слово, то

$$\neg !\mathcal{T}(x)$$

**Определение 10.** Конфигурация машины Тьюринга называется тупиковой, если она не является заключительной и при этом из нее не выводится ни одна конфигурация.

**Пример.**

$$f(x) = \begin{cases} \#, & \text{если } x = \lambda \\ \lambda, & \text{если } cab \sqsubseteq x \\ x, & \text{если } x \neq \lambda \text{ и } cab \not\sqsubseteq x \end{cases}$$

$\lambda$  - Пустое слово.

Тогда программа записывается так:

$$\begin{aligned} q_0 \otimes & \rightarrow q_0 \otimes, R \\ q_0 \square & \rightarrow q_f \#, L \\ q_0 a & \rightarrow q'_0 a, R \\ q_0 b & \rightarrow q'_0 b, R \\ q_0 c & \rightarrow q_1 c, R \\ q'_0 a & \rightarrow q'_0 a, R \\ q'_0 b & \rightarrow q'_0 b, R \\ q'_0 c & \rightarrow q_1 c, R \\ q_1 a & \rightarrow q_2 a, R \\ q_1 b & \rightarrow q'_0 b, R \\ q_1 c & \rightarrow q_1 c, R \\ q_2 a & \rightarrow a'_0 a, R // caa \\ q_2 b & \rightarrow q_3 b, R // cab \\ q_2 c & \rightarrow q_1 c, R // cac \\ q_3 \alpha & \rightarrow q_3 \alpha, R // \alpha \in \{a, b, c\} \\ q_3 \square & \rightarrow q_4 \square, L \\ q_4 \alpha & \rightarrow q_4 \square, L \\ q_4 \otimes & \rightarrow q_f \otimes, S \\ r \square & \rightarrow q_5 \square, L // r \in \{q'_0, q_1, q_2\} \\ q_5 \alpha & \rightarrow q_5 \alpha, L \\ q_5 \otimes & \rightarrow q_f \otimes, S \end{aligned}$$

Для ошибочного решения ( $q'_0$  не вводится):

$$(a_1, \lambda, \otimes ab \square) \vdash (q_0, \otimes, ab \square) \vdash (q_0, \otimes a, b \square) \vdash (q_0, \otimes ab, \square) \vdash \underline{(q_f, \otimes a, b \# \square)}$$

**Определение 11.** Машина Тьюринга называется детерминированной, если из каждой ее конфигурации непосредственно выводится не более одной конфигурации.

**Теорема 1.1.** Машина Тьюринга называется детерминированной тогда и только тогда, когда в ее программе (системе команд) нет двух (более) различных команд с одинаковыми левыми частями.

**Соглашение.** Во всех дальнейших суждениях машина Тьюринга будет считаться детерминированной. ДМТ - детерминированная машина Тьюринга.

Допустим машина Тьюринга с алфавитом  $V$ , то мы говорим, что это машина Тьюринга в алфавите  $V$ . Но если  $V \supset V'$ , то мы говорим, что Машина Тьюринга над алфавитом  $V$ .

**Определение 12.** Вербальная функция  $f : V^* \rightarrow V^*$  называется вычислительной по Тьюрингу, если может быть построена МТ  $\mathcal{T}_f$  над алфавитом  $V$  такая, что

$$(\forall x \in V^*)(\mathcal{T}(x) \iff x \in D(f) \& \mathcal{T}_f(x) = f(x))$$

**Тезис Тьюринга.** Он гласит, что любая вербальная функция, вычисляемая в интуитивном смысле слова, вычислима по Тьюрингу.

Общие разделы:

1. Основная модель.
2. Понятие вычислимой функции. Основная гипотеза.
3. Эквивалентный алгоритм.
4. Теорема сочетания.
5. Универсальный алгоритм.
6. Разрешимые перечислимые множества (языки).
7. Анализ алгоритмически неразрешимых задач.

## 1.3 Нормальные алгоритмы Маркова

Предположим, что есть

$$V; x, y \in V^*; x \sqsubseteq y \iff (\exists y_1, y_2)(y = y_1 x y_2)$$

причем тройка слов  $(y_1, x, y_2)$  - вхождение слова  $x$  в слово  $y$ .

Некоторые свойства:

- $(\forall x)(\lambda \sqsubseteq x)$
- $(\forall x)(x \sqsubseteq x)$
- $(\forall x)(\forall y)(\forall z)(x \sqsubseteq y, y \sqsubseteq z \implies x \sqsubseteq z)$

Записывается иногда так:  $y_1 * x * y_2$  ( $x \notin V$ )

Пример:  $y = \underbrace{\quad}_x$ ; \*вход\*ит - корень

Еще один:  $\underbrace{\quad}_x \text{ абракадабра } \underbrace{\quad}_x$

Среди всех вхождений  $x$  в  $y$  выделяется первое, или главное, вхождение, а именно имеющую наименьшую длину левого крыла (самое левое вхождение).

**Определение 13.** Подстановка:

$$u, v \in V^* \quad \underbrace{u}_{\text{л.ч.}} \rightarrow \underbrace{v}_{\text{п.ч.}}; \rightarrow \notin V$$

**Определение 14.** Омега применима, или подходит, если ее левая часть входит в слово  $x$ .

$$\omega : u \rightarrow v$$

Тогда вхождение:

$$x = x_1 u x_2; \quad x_1 * u * x_2 - 1\text{-е вхождение } u \text{ в } x$$

Отсюда

$$y \Leftarrow \omega x \Leftarrow x_1 v x_2$$

Это можно представить так:

$$\begin{array}{lcl} x & = & \boxed{x_1} \boxed{u} \boxed{x_2} \\ & \downarrow & \\ y = \omega x & = & \boxed{x_1} \boxed{v} \boxed{x_2} \end{array}$$

**Пример.** Пусть дана замена:

$$\omega : B \rightarrow y$$

Тогда слово Входит превратится в слово уходит.  $\omega x = \text{уходит}$

**Определение 15.** Нормальный алгоритм  $\mathcal{A} = (V, S, \mathcal{P})$

**Пример.**

$$\mathcal{A} : \begin{cases} \#a \rightarrow a(1) \\ \#b \rightarrow b\# \\ \# \rightarrow \cdot aba \\ \rightarrow \# \end{cases}$$

Рассматриваем систему сверху вниз и ищем первую подходящую формулу. Пусть

$$x = bbab$$

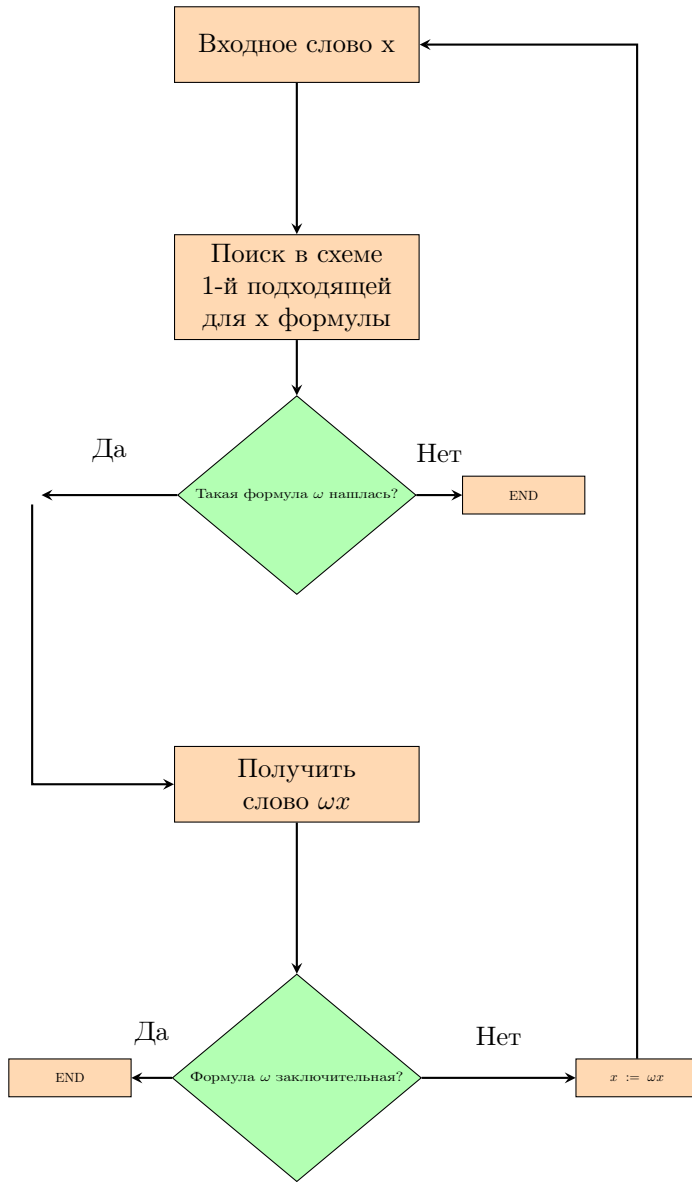
Отсюда получаем:

$$x = bbab \vdash \#bbab \vdash b\#bab \vdash bb\#ab \vdash bba\#b \vdash bbab\# \vdash \cdot bbab\underline{aba}$$

Общий вид:

$$\mathcal{A} : \begin{cases} u_1 \rightarrow [\cdot]v_1 \\ u_2 \rightarrow [\cdot]v_2 \\ \vdots \\ u_n \rightarrow [\cdot]v_n \end{cases}$$

Можно записать это в виде блок-схемы неформально:



Теперь формально опишем его. Распишем 5 разных ситуаций.

- 1)  $\mathcal{A} : x \vdash y \Leftrightarrow$  непосредственно просто переводит слово  $x$  в слово  $y \Leftrightarrow y = \omega x$ , где  $\omega$  - 1-я в схеме  $\mathcal{A}$  формула, которая оказывается простой
- 2)  $\mathcal{A} \vdash \bullet y \Leftrightarrow$  Алгоритм  $\mathcal{A}$  непосредственно заключительно переводит слово  $x$  в слово  $y \Leftrightarrow y = \omega x$ , где  $\omega$  - 1-я в схеме  $\mathcal{A}$ , которая оказывается заключительной
- 3)  $\mathcal{A}x \models y \Leftrightarrow$  Алгоритм  $\mathcal{A}$  переводит слово  $x$  в слово  $y$ , когда существует последовательность  $x = x_0, x_1, \dots, x_n = y$ , где  $(\forall i = \overline{0, n-1})(\mathcal{A} : x_i \vdash x_{i+1})$
- 4)  $\mathcal{A} : x \models \bullet y \Leftrightarrow$  Алгоритм  $\mathcal{A}$  заключительно переводит слово  $x$  в слово  $y \Leftrightarrow \mathcal{A} : x \vdash \bullet y \vee (\exists z)(\mathcal{A} : x \models z \vdash \bullet y)$
- 5)  $\sim \mathcal{A}(x) \Leftrightarrow$  в схеме  $\mathcal{A}$  нет ни одной подходящей формулы для  $x$ .

Процесс работы НА  $\mathcal{A} = (S, S, P)$  со словом  $x \in V^*$  : это последовательность слов  $x = x_0, x_1, \dots, x_n, \dots$  такая, что  $(\forall i \geq 0)(\mathcal{A} : x_i \vdash x_{i+1} \text{ или } \mathcal{A} : x_i \vdash \bullet x_{i+1})$ , если  $x_{i+1}$  определено в последовательности.

Слово  $x_{i+1}$  и каждое слово  $x_n, n > i + 1$  считается неопределенным, если  $\mathcal{A} : x_{i-1} \vdash \bullet x_i \text{ или } \sim \mathcal{A}(x_i)$

Если процесс работы НА  $\mathcal{A}$  со словом конечный, то есть  $x = x_0, x_1, \dots, x_n, n \geq 0$ , то  $!\mathcal{A}(x)$  и  $x_n \Leftrightarrow \mathcal{A}(x)$ . В противном случае пишем  $\neg !\mathcal{A}(x)$ , то есть алгоритм со словом  $x$  будет бесконечный, или не останавливается.



**Об алфавитах в НА.** Пусть НА алгоритм  $\mathcal{A} = (V, S, P)$ . Тогда мы говорим, что это НА в алфавите  $V$ . Пусть  $\mathcal{A}_1 = (V_1 \subset V, S_1, P_1)$  - нормальный алгоритм над алфавитом  $V$ .

**Определение 16.** Вербальная функция  $f : V^* \rightarrow V^*$  называется вычислимой по Маркову, если может быть построен нормальный алгоритм  $\mathcal{A}_f$  над алфавитом  $V$  такой, что

$$(\forall x \in V^*)(! \mathcal{A}_f(x) \iff x \in D(f)) \& (\mathcal{A}_f(x) = f(x))$$

**Гипотеза НА (Принцип нормализации).** Любая вербальная функция, вычисляемая в интуитивном смысле слова, вычислима по Маркову.

**Примеры НА.** Первый пример.

$$\mathcal{I}\alpha : \left\{ \rightarrow \bullet \right.$$

Получаем вот что:  $(\forall x)(\mathcal{I}\alpha(x) = x)$ , то есть вычисляет тождественную функцию в любом алфавите.

Второй пример.

$$Null : \left\{ \rightarrow \right.$$

Для любого слова будет работать бесконечно:  $(\forall x) \neg !Null(x)$

Третий пример.

$$Lc : \left\{ \rightarrow \bullet x_0, \text{ где } x_0 \in V^* - \underline{\text{фиксированное слово}} \right.$$

Получим:  $x \in V^* : x \vdash \bullet x_0 x$ , то есть  $Lc(x) = x_0 x$

Четвертый пример.

$$Rc : \left\{ \begin{array}{l} \# \xi \rightarrow \xi \# \\ \# \rightarrow \bullet x_0 (x_0 \in V^* - \text{фиксированное слово}) \\ \rightarrow \# \end{array} \right.$$

$$x \in V^*, x = x(1)x(2) \dots x(k) \vdash \# x(1)x(2) \dots x(k) \vdash x(1)\#x(2) \dots x(k) \models^{k-1} x\# \vdash \bullet x x_0$$

Пятый пример.

$$Double : \left\{ \begin{array}{l} \alpha \xi \rightarrow \xi \beta \xi \alpha \\ \beta \xi \eta \rightarrow \eta \beta \xi \\ \beta \rightarrow \\ \alpha \rightarrow \bullet \\ \rightarrow \alpha \end{array} \right.$$

Причем  $\alpha, \beta \notin V; \xi, \eta \in V$ .

Первый тест:  $\lambda \vdash \alpha \vdash \bullet \lambda$ .

Второй тест:  $a \vdash \alpha a \vdash a \beta a \alpha \vdash a a \alpha \vdash \bullet a a$

Третий тест:

$$\begin{aligned} abca \vdash \alpha abca \vdash a \beta a \alpha bca \vdash a \beta ab \beta b \alpha ca \vdash \\ \vdash a \beta ab \beta bc \beta c \alpha a \vdash a \beta ab \beta bc \beta ca \beta a \alpha \vdash \\ \vdash ab \beta a \beta bc \beta ca \beta a \alpha \vdash ab \beta ac \beta b \beta ca \beta a \alpha \vdash \\ \vdash abc \beta a \beta b \beta ca \beta a \alpha \vdash abc \beta a \beta ba \beta c \beta a \alpha \vdash \\ \vdash abc \beta aa \beta b \beta c \beta a \alpha \vdash abca \beta a \beta b \beta c \beta a \alpha \models^4 \\ \models^4 abca abca \alpha \vdash \bullet abca abca \end{aligned}$$

Можно строго доказать, что

$$(\forall x \in V^*)(Double(x) = xx = x^2)$$

## 1.4 Эквивалентность нормальных алгоритмов. Теорема о переводе.

Пусть даны  $\mathcal{A}, \mathcal{B} : V^* \rightarrow V^*$  над алфавитом  $V$ .

**Определение 17.** Алогрифмы  $\mathcal{A}, \mathcal{B}$  называются эквивалентными относительно алфавита  $V$ , если

$$(\forall x \in V^*)(! \mathcal{A}(x) \iff ! \mathcal{B}(x) \ \& \ (\mathcal{A}(x) = \mathcal{B}(x)))$$

Это называется условным равенством:

$$\mathcal{A}(x) \simeq \mathcal{B}(x)$$

**Рассмотрим такую конструкцию, называемую замыканием НА.**

$$\mathcal{A} : \begin{cases} u_1 \rightarrow [\cdot]v_1 \\ \vdots \\ u_n \rightarrow [\cdot]v_n \end{cases}$$

$$\mathcal{A}^* : \begin{cases} \text{Схема } \mathcal{A} \\ \rightarrow \cdot \end{cases}$$

То есть

$$(\forall x \in V^*) \mathcal{A}^*(x) \simeq \mathcal{A}(x)$$

Рассмотрим преобразования:

$\mathcal{A} : x \models \cdot y$ , то есть  $\mathcal{A}(x) = y$ ;  $\mathcal{A}^* : x \models y = \mathcal{A}(x)$ .

$\mathcal{A} : x \models y$ , то есть  $y = \mathcal{A}(x)$ ;  $\mathcal{A}^* : x \models y \vdash \cdot y = \mathcal{A}(x)$

**Замечка.** Переход к замыканию НА позволяет без ограничения общности не рассматривать ситуацию естественного обрыва процесса работы.

Если  $! \mathcal{A}(x)$ , то  $x \models \cdot \mathcal{A}(x)$  (система  $\mathcal{A}$  замкнутая)

**Естественное распространение НА на более широкий алгорифм.**  $\mathcal{A} = (V, S, P)$  и пусть  $V' \supset V$ . Тогда  $\mathcal{A}' = (V', S, P)$ . То есть просто означает, что рассматриваем тот же алгоритм в более широком алфавите. Из этого следует, что

$$(\forall x \in V^*)(\mathcal{A}(x) \simeq \mathcal{A}(x))$$

**Формальное распространение НА на более широкий алфавит.**  $\mathcal{A} = (V, S, P)$  в алфавите  $V$ .

$$\mathcal{A}^f : \begin{cases} \eta \rightarrow \eta \ // \ \eta \in V' \setminus V \\ \text{Схема } \mathcal{A} \end{cases}$$

Получаем:

$$(\forall x \in V^*)(\mathcal{A}^f(x) = \mathcal{A}(x)), \text{ но если } x \notin V^*, \text{ то } \neg ! \mathcal{A}^f(x)$$

Нам нужно расширить алфавит. Как это делается?

Рассмотрим алфавиты  $V = \{a_1, a_2, \dots, a_n\}$ ,  $V_\alpha = \{\alpha, \beta\}$  и  $V \cap V_\alpha = \emptyset$

Тогда считается

$$[a_i \Leftarrow \alpha \beta^i \alpha; \quad [\lambda = \lambda; \quad [x = [x(1)x(2) \dots x(k) \Leftarrow [x(1)[x(2) \dots [x(k)$$

**Пример.**

$$[abca = \underbrace{010}_{V_0} \underbrace{0110}_a \underbrace{0111}_b \underbrace{010}_c \underbrace{010}_a$$

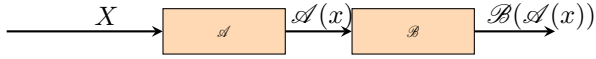
$$V_\alpha = \{\alpha, \beta\}$$

Чаще всего будет рассматривать такой алфавит:  $V_0 = \{0, 1\}$

**Теорема 1.2.** (О переводе). Каков бы ни был нормальный алгорифм  $\mathcal{A} = (V', S, P)$  над алфавитом  $V \subset V'$ , может быть построен НА  $\mathcal{B}$  в алфавите  $V \cup V_\alpha$  так, что  $(\forall x \in V^*)(\mathcal{B}(x) \simeq \mathcal{A}(x))$

## 1.5 Теорема сочетания

### 1.5.1 Композиция



**Теорема 1.3.** (О композиции). Каковы бы ни были НА  $\mathcal{A}, \mathcal{B}$  в алфавите  $V$  может быть построен НА алгоритм  $\mathcal{C}$  над алфавитом  $V$  такой, что

$$(\forall x \in V^*)(\mathcal{C}(x) \simeq \mathcal{B}(\mathcal{A}(x)))$$

**Доказательство.** Вводится алфавит двойников.

$$V = \{a_1, a_2, \dots, a_n\} \quad \bar{V} = \{\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n\}$$

Вводятся две буквы  $\alpha, \beta$  такие, что  $\alpha, \beta \notin V \cup \bar{V}$

$$\mathcal{C} : \begin{cases} \xi\alpha \rightarrow \alpha\xi // \xi \in V \\ \alpha\xi \rightarrow \alpha\bar{\xi} \\ \bar{\xi}\eta \rightarrow \bar{\xi}\bar{\eta} // \xi, \eta \in V \\ \bar{\xi}\beta \rightarrow \beta\bar{\xi} \\ \beta\bar{\xi} \rightarrow \beta\xi \\ \xi\bar{\eta} \rightarrow \xi\eta \\ \alpha\beta \rightarrow \bullet \\ \bar{\mathcal{B}}_\alpha^\beta \\ \mathcal{A}^\alpha \end{cases}$$

$\mathcal{A}^\bullet$	$\mathcal{A}^\alpha$
$u \rightarrow v$	$u \rightarrow v$
$u \rightarrow \bullet v$	$u \rightarrow \alpha v$

$\mathcal{B}^\bullet$	$\bar{\mathcal{B}}_\alpha^\beta$
$u \rightarrow v$	$\bar{u} \rightarrow \bar{v}$
$u \neq \lambda$	
$\rightarrow v$	$\alpha \rightarrow \alpha\bar{v}$
$u \rightarrow \bullet v$	$\bar{u} \rightarrow \beta\bar{v}$
$\rightarrow \bullet v$	$\alpha \rightarrow \alpha\beta\bar{v}$

**Примерно идея доказательства.**  $x \in V^*$

$$\mathcal{C} : x \models_{(9)}^{\mathcal{A}^\bullet(x)} y_1 \alpha y_2, \text{ где } y_1 y_2 = \mathcal{A}^\bullet(x)$$

Если  $\neg \mathcal{A}^\bullet(x)$ , то и  $\neg \mathcal{C}(x)$ , заметим. Отсюда

$$y_1 \alpha y_2 \models_{(1)} \alpha y_1 y_2 = \alpha y = \alpha y(1) y(2) \dots y(m),$$

где  $y_1 y_2 = y$ . Далее получаем

$$\alpha y(1) y(2) \dots y(m) \vdash_{(2)} \overline{\alpha y(1) y(2) \dots y(m)} \models_{(3)} \overline{\alpha y(1) y(2) \dots y(m)} = \alpha \bar{y}$$

Следующий, третий шаг

$$\alpha \bar{y} \models_{(8)} \alpha \bar{z}_1, \beta \bar{z}_2 z, \text{ где } z_1, z_1 = z = \mathcal{B}^\bullet(y), \text{ если } \mathcal{B}(y)$$

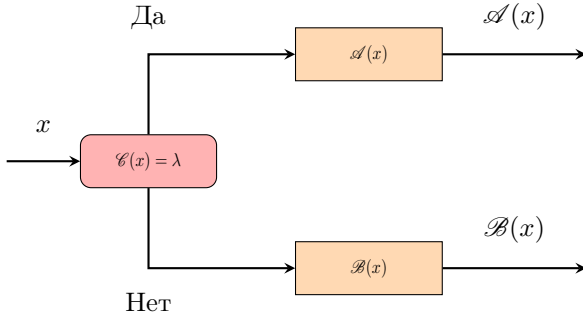
Заметим, что если  $\neg \mathcal{B}^\bullet(y) \implies \neg \mathcal{C}(y) \implies \neg \mathcal{C}(x)$ . Получаем

$$\alpha \bar{z}_1 \beta \bar{z}_2 \models_{(4)} \alpha \beta \bar{z}_1 \bar{z}_2 = \alpha \beta z \models_{(5),(6)} \alpha \beta z \vdash \bullet z = \mathcal{B}^\bullet(y) = \mathcal{B}^\bullet(\mathcal{A}^\bullet(x)) = \mathcal{B}(\mathcal{A}(x))$$

□



### 1.5.3 Разветвление



Записать в виде псевдокода можно так:

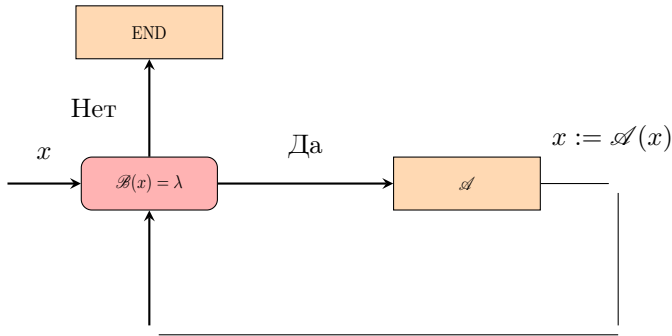
$$if(C(x) = \lambda) \text{ then } y := A(x) \text{ else } y := B(x);$$

**Теорема 1.5.** (О разветвлении). Каковы бы ни были НА  $A, B, C$  в алфавите  $V$ , может быть построен НА  $D$  над алфавитом  $V$  так, что

$$(\forall x \in V^*)(D(x) = A(x), \text{ если } C(x) = \lambda) \text{ и } (D(x) = B(x), \text{ если } C(x) \neq \lambda)$$

$$D \rightleftharpoons C(A \vee B)$$

### 1.5.4 Повторение



В виде псевдокода:

- Для цикла с условием, пока правда:

$$\underline{while} B(x) = \lambda \underline{do} x := A(x) \underline{end}; \text{ Записывается так: } {}_{\beta}\{A\}$$

- Для цикла с условием, пока неправда:

$$\underline{while} B(x) \neq \lambda \underline{do} x := A(x) \underline{end}; \text{ Записывается так: } {}_{\beta}\langle A \rangle$$

**Теорема 1.6.** (Повторения). Каковы бы ни были НА  $A, B$  в алфавите  $V$ , может быть построен НА  $C$  над алфавитом  $V$  такой, что  $!C(x) \rightleftharpoons (B(x) \neq \lambda)$  и тогда  $C(x) = x$  или существует последовательность  $x = x_0, x_1, \dots, x_n$ , где  $(\forall i = \overline{0, n-1}) (B(x_i) = \lambda) \text{ и } x_{i+1} = A(x_i); B(x_n) \neq \lambda \text{ и } C(x) = x_n$

**Примеры использования теоремы сочетания.**

#### 1) Проецирующие НА

Дано  $V, \$ \notin V$ . Векторное слово в алфавите  $V : x_1 \$ x_2 \$ \dots \$ x_n, n \geq 1$ , где  $(\forall i = \overline{1, n})(x_i \in V^*)$

Нужен алгоритм, который вычисляет его  $x_i$

$$\prod_i (x_1 \$ x_2 \$ \dots \$ x_n) = x_i, \quad i = 1 \dots n$$

$$\mathcal{P}_1 : \begin{cases} \$\eta \rightarrow // \eta \in V \\ \$ \rightarrow \\ \rightarrow \bullet \end{cases}$$

Результат работы  $\mathcal{P}_1(x_1\$x_2\$ \dots \$x_n) = x_1$

$$\mathcal{P}_2 : \begin{cases} \eta \rightarrow \# // \eta \in V, \# \notin V \\ \# \rightarrow \bullet \\ \$ \rightarrow \# \end{cases}$$

То есть  $\mathcal{P}_2(x_1\$x_2\$ \dots \$x_n) = x_2\$ \dots \$x_n$

Получаем  $\prod_i = \mathcal{P}_1 \circ \mathcal{P}_2^{i-1}, \quad 1 \leq i \leq n$

i = 1:  $\mathcal{P}_2^{i-1} = \mathcal{P}_2^0 = \mathcal{I}\alpha$

i = n:  $\mathcal{P}_2^{n-1}(x_1\$x_2\$ \dots \$x_n) = x_w; \quad \mathcal{P}_1(x_n) = x_n$

2) НА распознавания равенства слов

$EQ(x\$y) = \lambda \iff x = y; \quad x, y \in V^*, \$ \notin V$

$EQ(x\$y) \simeq Comp(\mathcal{I}\alpha\$Inv(y))$

$Inv(y) = y^R$

$$Comp : \begin{cases} \eta\$ \eta \rightarrow \$ // \eta \in V \\ \$ \rightarrow \bullet \end{cases}$$

$x^R = (x(1)x(2) \dots x(k))^R = x(k) \dots x(2)x(1)$

3) НА определения центра слова

$\mathcal{C}(x) = x_1\$x_2$ , где  $x_1x_2 = x, \quad ||x_1| - |x_2|| \leq 1, x \in V^*; \quad \$ \notin V$

$\mathcal{C} = \mathcal{B} \circ \mathcal{A} \langle L \circ R \rangle$

$$L : \begin{cases} \alpha\beta \rightarrow \bullet\alpha\beta \\ \alpha\xi \rightarrow \bullet\xi\alpha // \xi \in V, \alpha \notin V \\ \rightarrow \alpha \end{cases}$$

$$R : \begin{cases} \gamma\xi \rightarrow \xi\gamma // \xi \in V; \beta, \gamma \notin V \\ \xi\gamma \rightarrow \bullet\beta\xi \\ \xi\beta \rightarrow \bullet\beta\xi \\ \rightarrow \gamma \end{cases}$$

$$\mathcal{A} : \begin{cases} \alpha\beta\xi \rightarrow \alpha\beta \\ \xi\alpha\beta \rightarrow \alpha\beta \\ \alpha\beta \rightarrow \bullet \\ \rightarrow \bullet \end{cases}$$

$$\mathcal{B} : \begin{cases} \alpha\beta \rightarrow \bullet\$ \\ \rightarrow \bullet\$ \end{cases}$$

Пример 1.  $\lambda, \quad \mathcal{B}(\lambda) = \$$

$\mathcal{A}(\lambda) = \lambda \implies$  тело цикла не выполнилось

Пример 2.  $x = a \in V$

$\mathcal{A}(a) = a \neq \lambda$

$$R : a \vdash \gamma a \vdash a \gamma \vdash \bullet \beta a$$

$$L : \beta a \vdash \alpha \beta a \vdash \bullet \alpha \beta a$$

$$\mathcal{A}(\alpha \beta a) = \lambda$$

$$\mathcal{B}(\alpha \beta a) = \$a$$

Пример 3.  $x = ab$

$$\mathcal{A}(ab) = ab \neq \lambda$$

$$R : ab \vdash \gamma ab \vdash^2 ab \gamma \vdash \bullet \alpha \beta b$$

$$L : \alpha \beta b \vdash \alpha \alpha \beta b \vdash \bullet \alpha \alpha \beta b$$

$$\mathcal{A}(a \alpha \beta b) = \lambda$$

$$\mathcal{B}(a \alpha \beta b) = a \$b$$

Пример 4.  $x = abcde$

$$\mathcal{A}(x) = x \neq \lambda$$

1 Итерация:

$$R : abcde \vdash \gamma abcde \vdash^5 abcde \gamma \vdash \bullet abcde \beta e$$

$$L : abcd \beta e \vdash \alpha abcd \beta e \vdash \bullet \alpha abcd \beta e$$

2 Итерация:

$$R : a \alpha abcd \beta e \vdash \bullet a \alpha bc \beta de$$

$$L : a \alpha bc \beta de \vdash \bullet a b \alpha c \beta de$$

3 Итерация:

$$R : ab \alpha c \beta de \vdash \bullet ab \alpha \beta cde$$

$$L : ab \alpha \beta cde \vdash \bullet ab \alpha \beta cde$$

$$\mathcal{A}(ab \alpha \beta cde) = \lambda$$

$$\mathcal{B}(ab \alpha \beta cde) = ab \$cde$$

## 1.6 Универсальный нормальный алгоритм.

Пусть дан НА:

$$\mathcal{A} : \begin{cases} u_1 \rightarrow [\cdot] v_1 \\ \vdots \\ u_n \rightarrow [\cdot] v_n \end{cases}$$

$$A^{\mathcal{A}} \Leftrightarrow u_1 \alpha [\beta] v_1 \gamma u_2 \alpha [\beta] v_2 \gamma \dots \gamma u_n \alpha [\beta] v_n, \text{ где } \alpha, \beta, \gamma \notin V$$

Пусть

$$\mathcal{A}_0 : \begin{cases} \#a \rightarrow a\# \\ \#b \rightarrow b\# \\ \# \rightarrow \bullet aba \rightarrow \# \end{cases}$$

Отсюда

$$A_0^{\mathcal{A}} = \#a\alpha a\#\gamma\#b\alpha b\#\gamma\#\alpha\beta aba\gamma\alpha\#$$

$$\mathcal{E}A_0^3 = \underbrace{01110}_{\#} \underbrace{010}_a \underbrace{011110}_{\alpha} \underbrace{010}_a \underbrace{01110}_{\#} \underbrace{0111110}_{\gamma}$$

$a$	$b$	$\#$	$\alpha$	$\beta$	$\gamma$
1	2	3	4	5	6

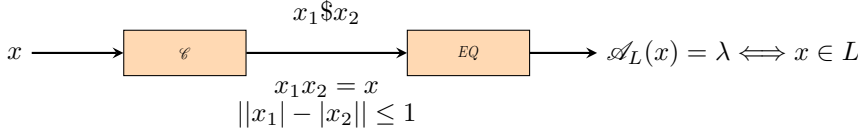
**Теорема 1.7.** (Об универсальном НА). Пусть  $V$  - произвольный алфавит. Может быть построен НА  $U$  над алфавитом  $V \cup V_0$  такой, что для любых НА  $\mathcal{A}$  в алфавите  $V$  и слова  $x \in V^*$  имеет место  $U(\mathcal{E}A_0^3 x) \simeq \mathcal{A}(x)$ , где  $\$ \notin V \cup V_0$

## 1.7 Разрешимые и перечислимые языки.

**Определение 19.** Язык  $L \subseteq V^*$  называется алгоритмически разрешимым, если может быть построен НА  $\mathcal{A}_L$  над алфавитом  $V$  такой, что

$$(\forall x \in V^*)(!\mathcal{A}_L) \text{ и } \mathcal{A}_L(x) = \lambda \iff x \in L$$

**Пример.** Пусть  $L = \{\omega\omega : \omega \in V^*\}$



Также стоит заметить, что здесь  $\mathcal{A}_L = C + EQ$ . Запись формальная и Белоусов может не понять, что здесь написано. А написано здесь то, что алгоритм  $\mathcal{A}_L$  состоит из  $C$  и  $EQ$ .

$$\underbrace{C \rightarrow EQ}_{\mathcal{A}_L}$$

**Определение 20.** НА  $\widetilde{\mathcal{A}}_L$  называется полурешимым для языка  $L \subseteq V^*$ , если

$$!\widetilde{\mathcal{A}}_L(x) \iff x \in L$$

**Теорема 1.8.** Если для языка  $L$  невозможен полурешающий НА, то невозможен и разрешающий.

**Доказательство.** От противного. Предполагаем, что для языка  $L$  невозможен полурешающий, то возможен разрешающий НА.

Пусть  $\mathcal{A}_L$  - разрешающий НА для  $L \subseteq V^*$

По теореме о разветвлении строим

$$\mathcal{B}_L = \mathcal{A}_L(\mathcal{A}_L \vee \text{Null}),$$

где

$$\text{Null} : \left\{ \begin{array}{l} \rightarrow \end{array} \right.$$

Если  $\mathcal{A}_L(x) = \lambda$ , то есть  $x \in L$ , то  $\mathcal{B}_L(x) = \mathcal{A}_L(x) = \lambda$ .

Если  $\mathcal{A}_L(x) \neq \lambda$ , то есть  $x \notin L$ , отсюда  $\neg !\mathcal{B}_L(x)$ , так как  $\neg !\text{Null}(x)$

Итак,  $!\mathcal{B}_L(x) \iff x \in L$ , то есть  $\mathcal{B}_L$  - полурешающий НА для  $L$  вопреки условию теоремы.  $\square$

**Теорема 1.9.** Если язык  $L$  разрешим, то и разрешимо его дополнение.

$$\mathcal{A}_L(x) = \lambda \iff x \in L, \text{ то есть } \mathcal{A}_L \neq \lambda \iff x \notin L \text{ при } (\forall x)!\mathcal{A}_L(x)$$

Для универсального языка:

$$L = V^* \quad \mathcal{A}_{V^*} : \left\{ \begin{array}{l} \xi \rightarrow // \xi \in V \\ \rightarrow \cdot \end{array} \right.$$

Отсюда следует, что и пустой язык тоже разрешим, потому что он - дополнение универсального.

**Определение 21.** Конструктивное натуральное число (КНЧ) - это слово вида  $0 \underbrace{11 \dots 1}_{n \geq 0}$ . Ноль кодирует ноль,

01 кодирует 1 и так далее. КНЧ  $x \in V_0^*$

$$0 \rightarrow 0; \quad 01 \rightarrow 1; \quad 011 \rightarrow 2; \quad \dots$$

**Определение 22.** Конструктивное целое число (КЦЧ) - это слово вида  $[-]n$ , где  $n$  - КНЧ.

**Определение 23.** Конструктивное рациональное число (КРЧ):  $m/n$ , где  $m, n$  - КЦЧ, то есть слово в  $\{0, 1, -, /\}$  и  $n \neq 0$

**Определение 24.** Язык  $L \subseteq V^*$  называется алгоритмически перечислимый, если может быть построен НА  $N_L$  такой, что для любого КНЧ  $n$   $!N_L(n)$  и  $N_L(n) \in L$ , и  $(\forall x \in L)$  существимо КНЧ  $n$  такое, что  $x = N_L(n)$



**Определение 25.**  $A, \nu : \mathbb{N}_0 \rightarrow A$  сюръективно, то есть  $(\forall x \in A)(\exists n \in \mathbb{N}_0)(x = \nu(n))$ . Это называется нумерацией множества  $A$ .

Далее будем предполагать, что отображение  $\nu$  будет биективной.

Проведем нумерацию целых чисел

рис1

Можно записать в виде формулы:

$$\gamma(n) = \begin{cases} -\frac{n}{2}, & \text{если } n \text{ четное} \\ \frac{n+1}{2}, & \text{если } n \text{ нечетное} \end{cases}$$

Сначала сделаем 3 алгоритма, нужных для следующей задачи (?)

$$\mathcal{C} : \begin{cases} 11 \rightarrow \\ 0 \rightarrow \bullet \end{cases}$$

Можем заметить, что  $\mathcal{C}(n) = \lambda \iff n \text{ четное}$

$$N_L = \mathcal{C}(\mathcal{A} \vee \mathcal{B})$$

Схема  $\mathcal{A}$ :

$$\mathcal{A} : \begin{cases} \alpha 11 \rightarrow 1\alpha \\ \alpha \rightarrow \bullet \\ 01 \rightarrow -0\alpha 1 \\ 0 \rightarrow \bullet 0 \end{cases}$$

Причем  $\alpha \notin V_0$

Схема  $\mathcal{B}$

$$\mathcal{B} : \begin{cases} \alpha 11 \rightarrow 1\alpha \\ \alpha \rightarrow \bullet \\ 01 \rightarrow 0\alpha 11 \\ \rightarrow \bullet \end{cases}$$

Нужно пронумеровать рациональные числа. Это по факту пары двух целых. Значит, учимся упорядочивать пары.

рис2

**Определение 26.** Область применимости НА  $\mathcal{A}$  относительно алфавита  $V$ : пусть  $\mathcal{A} = (V' \supset V, S, P)$  - НА над  $V$ ; Тогда область применимости НА относительно алфавита  $V$  есть множество  $\mathcal{M}_{\mathcal{A}}^V \Leftarrow \{x : x \in V^* \text{ и } !\mathcal{A}(x)\}$ , причем  $\mathcal{A} : V^* \rightarrow V^*$ .  $\mathcal{M}_{\mathcal{A}}^V$  и есть область применимости.

**Теорема 1.10.** Язык  $L \subseteq V^*$  перечислим тогда и только тогда, когда он является областью применимости относительно алфавита  $V$  некоторого НА.

**Следствие.** Всякий разрешимый язык перечислим.

**Доказательство.** (следствия). Пусть  $L$  - разрешимый язык и  $\mathcal{A}_L$  - разрешающий НА.

Строим такой НА  $\mathcal{B}_L = \text{Empty} \circ \mathcal{A}_L$ , где  $\text{Empty}$  применим только к пустому слову.

$$\text{Empty} : \begin{cases} \xi \rightarrow \xi // \xi \in V \\ \rightarrow \bullet \end{cases}$$

Отсюда получаем

$$!\mathcal{B}_L(x) \iff !\mathcal{A}_L(x) \text{ и } \mathcal{A}_L(x) = \lambda,$$

то есть  $L = \mathcal{M}_{\mathcal{B}_L}^V$

Однако обратное неверно!

□

## 1.8 Проблема применимости нормальных алгоритмов Маркова

**Частная проблема применимости.** Дан НА  $\mathcal{A}$  в алфавите  $V$ . Можно ли построить НА  $\mathcal{B}$  над алфавитом  $V$  такой, что  $(\forall x \in V^*) !\mathcal{B}(x)$  и  $\mathcal{B}(x) = \lambda \iff \neg !\mathcal{A}(x)$ . Алгоритм Б задуман для того, чтобы расширить область применимости алгоритма А.

**Общая проблема применимости.** Дан алфавит  $V$ ,  $\$ \notin V \cup V_0$ . Можно ли построить НА  $\mathcal{B}$  над алфавитом  $V \cup V_0$  так, что для любых НА  $\mathcal{A}$  в алфавите  $V$  и слова  $x \in V^*$

$$!\mathcal{B}(\mathcal{A}\$x) \text{ и } \mathcal{B}(\mathcal{A}\$x) = \lambda \iff \neg !\mathcal{A}(x)$$

### 1.8.1 Проблема самоприменимости.

Рассмотрим проблему самоприменимости. Мы хотим, чтобы алгоритм работал со своей собственной записью.

**Соглашение.** В дальнейшем, не оговаривая это особо, мы считаем, что алгоритм в алфавите  $V$  заменяем его в алфавит  $V \cup V_0$

$$V \rightarrow V \cup V_0$$

$$V_1 = V \cup V_0 \cup \{\alpha, \beta\}; \alpha, \beta \notin V \cup V_0$$

$$\mathcal{A} : (V \cup V_0)^* \subset \rightarrow (V \cup V_0)^*$$

$$V_2 = V_0 \cup \{\alpha, \beta\}$$

Дан алфавит  $V$ . Можно ли построить НА  $\mathcal{B}$  над алфавитом  $V_0$  такой, что для любого НА  $\mathcal{A}$  в  $V \cup V_0$  будет верно

$$!\mathcal{B}(\mathcal{A}\$) \text{ и } \mathcal{B}(\mathcal{A}\$) = \lambda \iff \neg !\mathcal{A}(\mathcal{A}\$)$$

**Примеры.** Построим как самоприменимые, так и несамоприменимые НА.

$$\mathcal{A}_0 : \begin{cases} \#a \rightarrow a\# \\ \#b \rightarrow b\# \\ \# \rightarrow \bullet aba \\ \rightarrow \# \end{cases}$$

Дадим ему на вход свою же запись:

$$\mathcal{A}_0 : \mathcal{A}_0\$ \vdash \# \mathcal{A}_0\$ \vdash \bullet aba \mathcal{A}_0\$$$

Причем  $V_0 \cap \{\#, a, b\} = \emptyset$ . Этот алгоритм самоприменим.

$$\mathcal{A}_0^f : \begin{cases} 0 \rightarrow 0 \\ 1 \rightarrow 1 \\ \text{Схема } \mathcal{A}_0 \end{cases}$$

Дадим ему на вход свою же запись:

$$\mathcal{A}_0^f : \mathcal{A}_0^f\$ \vdash \mathcal{A}_0^f\$ \vdash \dots$$

То есть  $\neg !\mathcal{A}_0^f(\mathcal{A}_0^f\$)$

**Лемма.** Невозможен НА  $\mathcal{B}$  в алфавите  $V \cup V_0$  такой, что для любого НА  $\mathcal{A}$  в алфавите  $V \cup V_0$  имело бы место

$$!\mathcal{B}(\mathcal{A}\$) \iff \neg !\mathcal{A}(\mathcal{A}\$)$$

**Доказательство.** Пусть алгоритм  $\mathcal{B}$  построен. Тогда при  $\mathcal{A} = \mathcal{B}$  имеем:

$$!\mathcal{B}(\mathcal{B}\$) \iff \neg !\mathcal{B}(\mathcal{B}\$)$$

что является противоречием. То есть он применим тогда, когда не применим?) □

**Теорема 1.11.** Невозможен НА  $\mathcal{B}$  над алфавитом  $V_0$  так, что для любого НА  $\mathcal{A}$  в алфавите  $V_1$  имело бы место

$$!\mathcal{B}(\mathcal{A}3) \iff \neg!\mathcal{A}(\mathcal{A}3)$$

**Доказательство.** По теореме о переводе может быть построен НА  $\mathcal{B}_1$  в алфавите  $V_0 \cup \{\alpha, \beta\}$  так, что  $(\forall x \in V_0^*) \mathcal{B}_1(x) \simeq \mathcal{B}(x)$ .

Строим НА  $\mathcal{B}_2$  как естественное распространение НА  $\mathcal{B}_1$  на алфавит  $V_1$ .

Пусть

$$!\mathcal{B}(\mathcal{A}3) \iff \neg!\mathcal{A}(\mathcal{A}3),$$

но тогда  $!\mathcal{B}(\mathcal{A}3) \iff !\mathcal{B}_1(\mathcal{A}3) \iff !\mathcal{B}_2(\mathcal{A}3) \iff \neg!\mathcal{A}(\mathcal{A}3)$ , что невозможно в силу самой леммы.  $\square$

Итак, мы доказали невозможность полуразрешимости самоприменимости.

Проблема самоприменимости для алгорифмов алгорифмически неразрешима.

**Теорема 1.12.** Язык записей несамоприменимых НА неперечислим.

**Доказательство.** Пусть указанный язык  $L = \{\mathcal{A}3 : \neg!\mathcal{A}(\mathcal{A}3)\}$  перечислим. Тогда  $L$  есть область применимости относительно алфавита  $V_0$  некоторого НА  $\mathcal{B}$ , то есть

$$!\mathcal{B}(\mathcal{A}3) \iff \neg!\mathcal{A}(\mathcal{A}3),$$

что невозможно!  $\square$

**Один вспомогательный НА.** Нам нужен такой НА:

$$Double^\$(x) = x\$x, \quad x \in V^*, \quad \$ \notin V$$

Его схема:

$$Double^\$ : \begin{cases} \alpha\xi \rightarrow \xi\beta\xi\alpha \\ \beta\xi\eta \rightarrow \eta\beta\xi \\ \alpha \rightarrow \$ \\ \beta\xi\$ \rightarrow \$\xi \\ \$ \rightarrow \bullet\$ \\ \rightarrow \alpha \end{cases}$$

причем  $\alpha, \beta, \# \notin V; \quad \xi, \eta \in V$

**Пример его работы.** Несколько примеров.

$$\textcircled{1} \lambda \vdash \alpha \vdash \$ \vdash \bullet\$$$

$$\textcircled{2} a \vdash \alpha a \vdash a\beta a \vdash a\beta a\$ \vdash a\$a \vdash \bullet a\$a$$

$$\begin{aligned} abc &\vdash \\ &\vdash \alpha abc \vdash a\beta a\alpha bc \vdash a\beta ab\beta b\alpha c \vdash \\ &\vdash \dots \vdash abc\$abc \\ &\vdash \bullet abc\$abc \end{aligned}$$

**Теорема 1.13.** Может быть построен НА  $\mathcal{A}$  в алфавите  $V_2$  так, что невозможен НА  $\mathcal{B}$  над алфавитом  $V_2$ , для которого выполнялось бы

$$!\mathcal{B}(y) \iff \neg!\mathcal{A}(y), y \in V_2^*$$

**Доказательство.** По теореме об универсальном НА построим НА  $U$  над алфавитом  $V_2$  так, что для любых НА  $D$  в алфавите  $V_2$  и слово  $y \in V_2^*$  выполняется

$$U(\mathcal{A}D3\$y) \simeq D(y).$$

Определим НА  $U_1$  так, что

$$(\forall y \in V_2^*)(U_1(y) \simeq U(y\$y)),$$

то есть  $U_1 = U \circ Double^\$$ .

Тонкий момент здесь! Алгоритм  $U_1$  будучи НА над алфавитом  $V_2$  тем самым является и НА над алфавитом  $V_0$  ( $V_2$  - расширение  $V_0$ ). По теореме о переводе он может быть заменен вполне эквивалентным ему относительно алфавита  $V_0$  НА  $U_2$  в алфавите  $V_2$  (то есть в двухбуквенном расширении  $V_0$ ).

$$U_2(x) \simeq U_1(x), \text{ где } x \in V_0^*, U_2 - \text{НА в } V_2 = V_0 \cup \{\alpha, \beta\}$$

Предположим, что такой НА  $\mathcal{B}$  нашелся.

$$!B(\&D3) \iff \neg!U_2(\&D3) \iff \neg!U_1(\&D3) \iff \neg!U(\&D3\&D3) \iff \neg!D(\&D3)$$

Он будет полуразрешающим НА для несамоприменимых НА в языке  $V_2$ , что невозможно.  $\square$

**Следствие.** Может быть построен НА с неразрешимой частной проблемой применимости, следовательно его область применимости будет перечислимая, но неразрешимая (множество?).

**Примеры неразрешимых проблем.** Проблема соответствия Поста.

$$\rho = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \subseteq V^{+2}$$

Существует ли

$$(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \dots, (x_{im}, y_{im}) : x_{i1}x_{i2} \dots x_{im} = y_{i1}y_{i2} \dots y_{im}?$$

## 1.9 Порождающие грамматики

**Определение 27.**  $\mathcal{J} = (V, N, S \in N, \Phi), V \cap N = \emptyset$

Правило вывода:  $\alpha \rightarrow \beta, \rightarrow \notin V \cup N$

Левая часть  $\alpha \in (V \cup N)^* N (V \cup N)^*, N$  - детерминал.

Пусть  $\gamma, \delta \in (V \cup N)^*$ . Тогда

$$\gamma \vdash_{\mathcal{J}} \delta \iff \text{сущ правило вывода } \alpha \rightarrow \beta \text{ в системе } \Phi \text{ и } \gamma = \gamma_1 \alpha \gamma_2, \delta = \gamma_1 \beta \gamma_2$$

**Определение 28.** Вывод в порождающей грамматике  $\mathcal{J}$  - это последовательность  $\alpha_0, \alpha_1, \dots, \alpha_n, \dots$ , где  $(\forall i \geq 0)(\alpha_i \in (V \cup N)^*)$  и  $(\forall i \geq 0)(\alpha_i \vdash_{\mathcal{J}} \alpha_{i+1})$ , если  $\alpha_{i+1}$  определен в последовательности.

**Определение 29.**  $\gamma \vdash_{\mathcal{J}}^* \delta \iff$  существует вывод

$\gamma = \alpha_0 \vdash \alpha_1 \vdash \dots \vdash \alpha_n = \delta, n \geq 0$  - длина вывода (к-рая конечна).

**Определение 30.**  $L(\mathcal{J}) \iff \{x : x \in V^*, S \vdash_{\mathcal{J}}^* x\}$

**Примеры грамматик.**

$$1) S \rightarrow aSb \mid \lambda$$

$$S \vdash aSb \vdash aaSbb \vdash \dots \vdash a^n S b^n \vdash a^n b^n$$

$$\mathcal{J}_1 = (\{a, b\}, \{S\}, S, \Phi_1)$$

Тогда язык, порожденный такой грамматикой

$$L(\mathcal{J}_1) = \{a^n b^n : n \geq 0\}$$

$$2) \Phi_2 : S \rightarrow aSa \mid bSb \mid aa \mid bb \mid a \mid b \mid \lambda$$

$$S \vdash aSa \vdash aba$$

$$S \vdash aSa \vdash abSba \vdash abbSbba \vdash abbbba$$

$$L(\mathcal{J}_2) = \{x : x = x^R, x \in \{a, b\}^*\} - \text{палиндром}$$

$$3) S \rightarrow () \mid (S) \mid SS - \text{правильная скобочная структура}$$

$$4) \mathcal{J}_4 = (\{a, b\}, \{S, A, B, C, D\}, S, \Phi_4)$$

$$\Phi_4 : \begin{cases} S \rightarrow CD \\ c \rightarrow aCA|bcD|\lambda \\ AD \rightarrow aD \\ BD \rightarrow bD \\ Aa \rightarrow aA \\ Ab \rightarrow bA \\ Ba \rightarrow aB \\ Bb \rightarrow bB \\ D \rightarrow \end{cases}$$

$$S \vdash CD \vdash \lambda D \vdash \lambda \lambda = \lambda$$

$$\begin{aligned} S \vdash CD \vdash aCAD \vdash abcBAD \vdash abbCBBAD \vdash abbBBAD \vdash \\ \vdash abbBBaD \vdash abbBaBD \vdash abbaBBD \vdash abbaBbD \vdash abbabBD \vdash \\ \vdash abbabbD \vdash abbabb \end{aligned}$$

$L(\mathcal{J}_4) \supseteq \{\omega\omega : \omega \in \{a, b\}^*\}$ . Можно доказать, что такой язык будет состоять только из двойных слов.

$$L(\mathcal{J}_4) = \{\omega\omega : \omega \in \{a, b\}^*\}$$

## 1.10 Классификации грамматик

- 1) Грамматики типа 0
- 2) Неукорачивающие грамматики (НК-)
- 3) Контекстно зависимые грамматики (КЗ-)
- 4) ОКЗ-грамматики (ограниченно КЗ)
- 5) Контекстно свободные (КС-)
- 6) Линейные грамматики
- 7) Праволинейные грамматики
- 8) Леголинейные грамматики
- 9) Регулярные (автоматные) грамматики

**Определение 31.** Грамматики называются эквивалентными, если они порождают один и тот же язык

$$G_1 \simeq G_2 \Leftrightarrow L(G_1) = L(G_2)$$

**Определение 32.** Грамматики называют почти эквивалентными, если порождаемые ими языки совпадают с точностью до пустого слова, то есть

$$G_1 \approx G_2 \Leftrightarrow L(G_1) \nabla L(G_2) \subseteq \{\lambda\}$$

**Теорема 1.14.**

- 1) Для каждой грамматики типа 0 может быть построена эквивалентная ей ОКЗ-грамматика
- 2) Для каждой неукорачивающей грамматики может быть построена эквивалентная ей КЗ-грамматика
- 3) Для каждой КС-грамматики может быть построена почти эквивалентная ей КС-грамматика, не содержащая правил с пустой правой частью (т.н. лямбда-правил)
- 4) Для каждой леголинейной грамматики может быть построена эквивалентная ей праволинейная грамматика и наоборот.
- 5) Для каждой праволинейной грамматики может быть построена эквивалентная ей регулярная грамматика

**Теорема 1.15.** Язык перечислим тогда и только тогда, когда он порождается грамматикой типа 0. Всякий КС-язык разрешим, но обратное неверно.

## 1.11 МП-автоматы (Pushdown machine)

рис1

$qaZ \rightarrow r\gamma$ , где  $q, r \in Q$ ,  $Z \in \Gamma$ ,  $\gamma \in \Gamma^*$ ,  $a \in V \cup \{\lambda\}$

рис2

**Пример**

$$q_0 a Z \rightarrow q_0 a Z$$

$$q_0 a a \rightarrow q_0 a a$$

$$q_0 b a \rightarrow q_1 \lambda$$

$$q_1 b a \rightarrow q_1 \lambda$$

$$q_1 \lambda Z \rightarrow q_2 \lambda$$

Машинный автомат может быть описан тоже в виде конфигураций. Начальное:

$$(q, ay, Z\alpha) \quad \alpha \in \Gamma^*, \text{ то есть может быть пустой}$$

$Z$  - все, что есть в магазине.

$$(q_0, aabb, Z) \vdash (q_0, abb, aZ) \vdash (q_0, bb, aaZ) \vdash (q_1, b, aZ) \vdash (q_1, \lambda, Z) \vdash (q_1, \lambda, \lambda)$$

**Определение 33.**  $\mathcal{M} = (Q, V, \Gamma, q_0, F, Z_0(\text{нач. маг. симв.}), \delta(\text{сист. перех.}))$  - магазинный автомат

**Определение 34.** Конфигурация МП-авт:  $(Q, ay, Z\alpha)$ , где  $q \in Q$ ,  $a \in V \cup \{\lambda\}$ ,  $y \in V^*$ ,  $z \in \Gamma$ ,  $\alpha \in \Gamma^*$

$$(q, ay, Z\alpha) \vdash_{\mathcal{M}} (r, y, \gamma\alpha) \Leftrightarrow qaZ \rightarrow r\gamma$$

Далее отношение непосредственной выводимости на мн-стве конфигурации рефлексивно-транзитивно замыкается подобно тому, как это было сделано на конфигурации машины Тьюринга.

**Определение 35.** Язык, допускаемый магазинным автоматом, - это

$$L(\mathcal{M}) \Leftrightarrow \{x : (q_0, x, Z_0)\} \vdash^* (q_f, \lambda, \alpha),$$

где  $q_f \in F$ .

Мы можем немного переопределить наш язык так:

$$L(\mathcal{M}) = \{x : (q_0, x, Z_0) \vdash^* (q_f, \lambda, \lambda); x \in V^*\}$$

**Теорема 1.16.** Язык является контекстно свободным тогда и только тогда, когда он допускается некоторым МП-автоматом.

Дано: КС-грамматика  $\mathcal{J} = (V, N, S, \mathcal{P})$

Строим: МП-автомат  $\mathcal{M} = (Q, V, \Gamma, q_0, F, z_0, \delta)$

$$\boxed{L(\mathcal{M}) = L(\mathcal{J})}$$

$$\mathcal{M} = (\{q\}, V, V \cup N, q, \{q\}, S, \delta_{\mathcal{P}})$$

$$\text{Причем } q\lambda A \rightarrow q\alpha \in \delta_{\mathcal{P}} \Leftrightarrow A \rightarrow \alpha \in \mathcal{P}$$

$$(\forall a \in V)(qa a \rightarrow q\lambda \in \delta_{\mathcal{P}})$$

**Пример 1.**

$\mathcal{J} : S \rightarrow aSa|bSb|aa|bb|a|b|$   
 То есть  $L(\mathcal{J}) = \{x : x = x^R, x \neq \lambda\}$   
 То есть система команд такая:

$$\delta_{\mathcal{J}} : \begin{cases} q \rightarrow qaSa|qbSb|qaa|qbb|qa|qb \\ qaa \rightarrow q\lambda \\ qbb \rightarrow q\lambda \end{cases}$$

$\mathcal{J} : S \vdash aSa \vdash abSba \vdash ababa$

Для автомата:

$(q, ababa, S) \vdash (q, ababa, aSa) \vdash (q, baba, Sa) \vdash (q, baba, bSba) \vdash (q, aba, Sba) \vdash (q, aba, aba) \models^3 (q, \lambda, \lambda)$  - допуск

**Пример 2.**

$S \rightarrow ab|aSb|SS$

$$\delta : \begin{cases} qaS \rightarrow qb|qsb \\ q\lambda S \rightarrow qSS \\ qaa \rightarrow q\lambda \\ qbb \rightarrow q\lambda \end{cases}$$

$S \vdash SS \vdash aSbS \vdash aabbS \vdash aabbab$

Как автомат ее разберет:

$(q, aabbab, S) \vdash (q, aabbab, SS) \vdash (q, abbab, SbS) \vdash (q, bbab, bbS) \models^2 (q, ab, S) \vdash (q, b, b) \vdash (q, \lambda, \lambda)$  - допуск

# Булевы функции

## 2.1 Булева алгебра

Свойства симметричного полукольца:

- $a + (b + c) = (a + b) + c$
- $a + b = b + a$
- $a + a = a$
- $a + 0 = a$
- $a * (b * c) = (a * b) * c$
- $a * 1 = 1 * a = a$
- $a * (b + c) = ab + ac$
- $a * 0 = 0 * b = 0$
- $ab = ba$
- $aa = a$
- $a + 1 = 1$
- $a + bc = (a + b)(a + c)$

Симметричное полукольцо:  $\mathcal{S} = (S, +, \cdot, 0, 1)$

Симметричное ему полукольцо:  $\mathcal{S}^* = (S, \cdot, +, 1, 0)$

$(\forall a)(a^* = 1)$

**Принцип двойственности симметрического полукольца.** Любое тождество, доказанное для симметрического полукольца, останется справедливым, если в нем произвести взаимные замены операции сложения и умножения, а также взаимные замены нуля и единицы.

**Пример.**

$$(a + b)(a + c) = a^2 + ac + ab + bc = a + ac + ab + bc = a \underbrace{(1 + c + b)}_1 + bc = a + bc$$

**Свойство 1.**  $a + ab = a(a + b) = a$

**Доказательство.**  $a(a + b) = a^2 + ab = a + ab = a(1 + b) = a * 1 = a$

□

**Свойство 2.**  $a \leq b \iff ab = a$

**Доказательство.**

$$a \leq b \implies a + b = b \implies ab = a(a + b) = a$$

$$ab = a \implies a + b = ab + b = ab + 1 * b = (a + 1)b = 1 * b = b$$

□



**Свойство 3.**  $(\forall a)(a \leq 1)$ , то есть  $(\forall a)(0 \leq a \leq 1)$

**Определение 36.** Дополнение элемента  $a$ :  $\bar{a} * a = 0$  и  $\bar{a} + a = 1$

**Теорема 2.1.** Если дополнение элемента симметрического полукольца определено, то оно определено однозначно.

**Доказательство.** Пусть  $(\exists x)(a + x = 1, ax = 0)$

Тогда

$$x = x + a * \bar{a} = (x + a)(x + \bar{a}) = 1(x + \bar{a}) = (a + \bar{a})(x + \bar{a}) = ax + \bar{a} = 0 + \bar{a} = \bar{a}$$

□

**Следствие.**  $\bar{\bar{a}} = a$

**Определение 37.** Булева алгебра - это симметричное полукольцо, в котором каждый элемент имеет дополнение.

**Примеры.**

$$\mathcal{B} = (\{0, 1\}, +, *, 0, 1)$$

$$\mathcal{S}_M = (2^M, \cup, \cap, \emptyset, M)$$

Булева алгебра обозначается так:

$$\mathcal{D} = (B, \vee, \wedge, \Theta, I, \bar{\phantom{x}})$$

**Теорема 2.2.** В любой булевой алгебре имеет место:

$$\overline{a \vee b} = \bar{a} \wedge \bar{b}; \quad \overline{a \wedge b} = \bar{a} \vee \bar{b}$$

**Доказательство.**

$$(a \vee b) \vee (\bar{a} \wedge \bar{b}) = (a \vee b \vee \bar{a}) \wedge (a \vee b \vee \bar{b}) = I$$

$$(a \vee b) \wedge (\bar{a} \wedge \bar{b}) = (\bar{a} \wedge \bar{b} \wedge a) \vee (\bar{a} \wedge \bar{b} \wedge b) = \Theta \vee \Theta = \Theta$$

Отсюда  $\overline{a \vee b} = \bar{a} \wedge \bar{b}$

□

Пусть дана булева алгебра  $\mathcal{B} = (B, \vee, \wedge, \Theta, I, \bar{\phantom{x}})$

$$\mathcal{B}^n = (B^n, \vee, \wedge, \tilde{\Theta}, \tilde{I})$$

Тогда пусть  $\tilde{\alpha}, \tilde{\beta} \in \mathcal{B}^n$ ;  $\alpha = (\alpha_1, \dots, \alpha_n)$

$$\beta = (\beta_1, \dots, \beta_n)$$

Отсюда

$$\tilde{\alpha} \vee \tilde{\beta} \Leftrightarrow (\alpha_1 \vee \beta_1, \dots, \alpha_n \vee \beta_n)$$

Аналогично и для  $\tilde{\alpha} \wedge \tilde{\beta}$ .

Также  $\tilde{\Theta} = (\Theta, \dots, \Theta)$  и  $\tilde{I} = (I, \dots, I)$

**Определение 38.** Булев куб размерности  $n$ :  $\mathcal{B}^n = (\{0, 1\}^n, \vee, \wedge, \tilde{0}, \tilde{1})$

Рассмотрим всевозможные отображения  $X$  в носитель булевой алгебры

$$f : X \rightarrow B$$

Тогда можно сказать такое:

$$1) (f \vee g)(x) \Leftrightarrow f(x) \vee g(x)$$

$$2) (f \wedge g)(x) \Leftrightarrow f(x) \wedge g(x)$$

$$3) \bar{f}(x) \Leftrightarrow \overline{f(x)}$$

$$\bullet (4) \sigma(x) \Leftrightarrow \Theta \quad (\forall x)$$

$$5) \xi(x) = I(\forall x)$$

**Определение 39.** Так обозначается булева алгебра функций:

$$\mathcal{B}^X = (B^X, \vee, \wedge, \sigma, \xi)$$

Булево кольцо, соответствующее булевой алгебре  $\mathcal{B}$

$$\mathcal{R}_B = (B, \oplus, \cdot, \Theta, I)$$

Отсюда

$$a \oplus b \Leftrightarrow a\bar{b} \vee \bar{a}b$$

$$a \cdot b \Leftrightarrow a \wedge b$$

$$\mathcal{S}_M = (2^M, \cup, \cap, \emptyset, M)$$

$$\mathcal{R}_M = (2^M, \Delta, \cap, \emptyset, M)$$

## 2.2 Булевы функции. Основные понятия

**Определение 40.** Булева функция от  $n$  переменных:

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

Булева переменная - это  $x_1, x_2, \dots, x_n$ . Функция выглядит обычно:  $y = f(x_1, \dots, x_n)$

Множество всех булевых функций:

$$\mathcal{P}_2 = \mathcal{P}_2^{(0)} \cup \mathcal{P}_2^{(1)} \cup \dots \cup \mathcal{P}_2^{(n)} \cup \dots$$

Нам известно определение  $n$ -арной операции:  $\omega : A^n \rightarrow A$ . То есть булевы функции своего рода  $n$ -арные операции.

	$f_1$	$f_2$	$f_3$	$f_4$
0	0	1	0	1
1	0	1	1	0

Можно заметить, что  $\bar{x} = x \oplus 1 = x \sim 0$

$$h = (0011111010101110) \iff h = \{2, 3, 4, 5, 6, 8, 10, 12, 13, 14\}$$

## 2.3 Равенство булевых функций. Фиктивные переменные

**Определение 41.** Пусть есть  $f, g : \{0, 1\}^n \rightarrow \{0, 1\}$ . Тогда функции равны, если

$$f = g \Leftrightarrow (\forall \tilde{\alpha} \in \{0, 1\}^n)(f(\tilde{\alpha}) = g(\tilde{\alpha}))$$

$$f(x_1, x_2) = x_1 \vee x_2$$

$$g(x_1, x_2, x_3) = x_1 x_3 \vee x_1 \bar{x}_3 \vee x_2 x_3 \vee x_2 \bar{x}_3 = x_1(x_2 \vee \bar{x}_3) \vee x_2(x_3 \vee \bar{x}_3) = x_1 \vee x_2$$

**Определение 42.** Булевы функции считаются равными, если они отличаются друг от друга, может быть, только фиктивными переменными.

Можно переформулировать так предыдущее определение.

**Определение 43.** Булевы функции равны, если они существенно зависят от одних и тех же переменных и на каждом наборе значений этих переменных принимают одинаковые значения

Пусть дан набор значений  $X = \{x_1, \dots, x_n\}$ . Тогда селектор  $pr_i(x_1, \dots, x_i, \dots, x_n) = x_n$  и иногда называется  $i$ -селектором.

Так можно добавит фиктивные переменные:

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

## 2.4 Суперпозиции и формулы

**Определение 44.** Пусть у нас есть  $f \in \mathcal{P}_2^{(n)}$ ,  $g_1, \dots, g_n \in \mathcal{P}_2^{(m)}$

$$f(g_1, \dots, g_n)(\tilde{\alpha}) = f(g_1(\tilde{\alpha}), \dots, g_n(\tilde{\alpha})), \quad \tilde{\alpha} \in \{0, 1\}^m$$

и это называется суперпозицией.

## 2.5 Дизъюнктивная и конъюнктивная нормальные формы (ДНФ и КНФ)

**Определение 45.** Литерал - это формула, в которой есть либо переменная, либо отрицание переменной.

$$x^\sigma \Leftrightarrow \begin{cases} x_i, & \text{если } \sigma = 1 \\ \overline{x_i}, & \text{если } \sigma = 0 \end{cases}$$

Обозначение  $\tilde{x}_i$  - это возможное отрицание.

**Определение 46.** Элементарная конъюнкция - это конъюнкция каких-то литералов.

$$\tilde{x}_{i_1} \tilde{x}_{i_2} \dots \tilde{x}_{i_k}$$

**Определение 47.** ДНФ - это  $k_1 \vee k_2 \vee \dots \vee k_m$  от  $x_1, x_2, \dots, x_n$ , где  $k_i$  - элементарная конъюнкция.

**Определение 48.** В СДНФ в каждую элементарную конъюнкцию входит каждый из  $x_1, x_2, \dots, x_n$  либо сам, либо как отрицание.

$$\text{ДНФ: } \{x_1, x_2, x_3\} : \quad \overline{x_1}x_2 \vee x_2 \vee x_1\overline{x_2}x_3$$

$$\text{СДНФ: } \{x_1, x_2, x_3\} : \quad x_1x_2x_3 \vee x_1\overline{x_2}x_3 \vee \overline{x_1}x_2x_3$$

**Определение 49.** Элементарная дизъюнкция - это дизъюнкция каких-то литералов.

**Определение 50.** КНФ от  $x_1, x_2, \dots, x_n$ :  $D_1 * D_2 * \dots * D_m, m \geq 1$

**Определение 51.** В СКНФ в каждую элементарную дизъюнкцию входит каждый из  $x_1, x_2, \dots, x_n$  либо сам, либо как отрицание.

**Теорема 2.3.** Любая функция, отличная от константы 0, может быть представлена в виде ДНФ. Любая функция, отличная от константы 1, может быть представлена в виде КНФ.

**Доказательство.** 1) Так как  $f \neq 0$ , то  $\exists \tilde{\alpha} \in \{0, 1\}^n : f(\tilde{\alpha}) = 1$  - называется это конституента 1 функции  $f$ .

Тогда

$$C_f^1 \Leftrightarrow \{\tilde{\alpha} : f(\tilde{\alpha}) = 1\} \neq \emptyset, \tilde{\alpha} = (\alpha_1, \dots, \alpha_n).$$

$$K_{\tilde{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$$

Заметим, что

$$K_{\tilde{\alpha}}(\tilde{\beta}) = 1 \iff \tilde{\beta} = \tilde{\alpha}$$

Отсюда получаем:

$$f(x_1, \dots, x_n) = \bigvee_{\tilde{\alpha} \in C_f^1} K_{\tilde{\alpha}}$$

Заметим, что если

$$f(x_1, \dots, x_m) = 1 \implies (\exists \tilde{\alpha} \in C_f^1)(f(\tilde{\alpha}) = 1) \implies k_{\tilde{\alpha}} = 1 \implies \bigvee_{\tilde{\alpha} \in C_f^1} k_{\tilde{\alpha}} = 1,$$

то есть  $f(\tilde{\alpha}) = 1$ . Аналогично для КНФ. □

**Следствие.** Любая булева функция может быть представлена некоторой формулой над стандартным базисом. То есть стандартным базисом является полным множеством булевых функций.

## 2.6 Полином Жегалкина

$$\mathcal{F}_1 = \{\oplus, *, 1\}$$

Отсюда  $\bar{x} = x \oplus 1$  и  $x_1 \vee x_2 = x_1 x_2 \oplus x_1 \oplus x_2$ .

**Определение 52.** Полиномом Жегалкина является

$$P(x_1, x_2, \dots, x_n) = \sum (mod 2) a_{i_1 i_2 \dots i_k} x_{i_1} x_{i_2} \dots x_{i_k}, \quad \{i_1, i_2, \dots, i_k\} \subseteq \{1, 2, \dots, n\}.$$

Здесь  $2^n$  слагаемых.  $a_{i_1 i_2 \dots i_k} \in \{0, 1\}$

Общий вид полинома Жегалкина от двух переменных:

$$P(x_1, x_2) = a_{12} x_1 x_2 \oplus a_1 x_1 \oplus a_2 x_2 \oplus a_0$$

Общий вид от трех:

$$P(x_1, x_2, x_3) = a_{123} x_1 x_2 x_3 \oplus a_{12} x_1 x_2 \oplus a_{13} x_1 x_3 \oplus a_{23} x_2 x_3 \oplus a_1 x_1 \oplus a_2 x_2 \oplus a_3 x_3 \oplus a_0$$

**Теорема 2.4.** Каждая булева функция однозначно представима в виде полинома Жегалкина.

**Метод неопределенных коэффициентов.**

$$f = (00010111)$$

$$f(0, 0, 0) = a_0 = 0$$

$$f(1, 0, 0) = a_1 \oplus a_0 = 0 \implies a_1 = 0$$

$$f(0, 1, 0) = a_2 \oplus a_0 = 0 \implies a_2 = 0$$

$$f(0, 0, 1) = a_3 \oplus a_0 = 0 \implies a_3 = 0$$

$$f(1, 1, 0) = a_{12} \oplus a_2 \oplus a_1 \oplus a_0 = 1 \implies a_{12} = 1$$

$$f(1, 0, 1) = a_{13} \oplus a_1 \oplus a_3 \oplus a_0 = 1 \implies a_{13} = 1$$

$$f(0, 1, 1) = a_{23} \oplus a_2 \oplus a_3 \oplus a_0 = 1 \implies a_{23} = 1$$

$$f(1, 1, 1) = a_{123} \oplus a_{12} \oplus a_{13} \oplus a_{23} \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_0 = 1 \implies a_{123} \oplus 1 = 1 \implies a_{123} = 0$$

**Определение 53.** Булева функция называется линейной, если она может быть представлена полиномом Жегалкина первой степени.

$$f \in L \iff f(x_1, \dots, x_n) = \sum_{i=1}^n (mod 2) a_i x_i \oplus a_0$$

## 2.7 Классы Поста

Всего 5 классов.

$$1) \mathcal{T}_0 \Leftarrow \{f : f(0, \dots, 0) = 0\}$$

$$2) \mathcal{T}_1 \Leftarrow \{f : f(1, \dots, 1) = 1\}$$

$$3) \mathcal{S} \Leftarrow \{f : (\forall \tilde{\alpha})(f(\tilde{\alpha}) = \overline{f(\tilde{\alpha})})\}$$

$$f \notin \mathcal{S} \iff (\exists \tilde{\alpha})(f(\tilde{\alpha}) \neq \overline{f(\tilde{\alpha})})$$

$$f, \quad f^*(\tilde{\alpha}) = \overline{f(\tilde{\alpha})} \iff \overline{f^*(\tilde{\alpha})} = f(\tilde{\alpha})$$

$$4) \mathcal{M} \Leftarrow \{f : (\forall \tilde{\alpha}, \tilde{\beta})(\tilde{\alpha} \leq \tilde{\beta} \implies f(\tilde{\alpha}) \leq f(\tilde{\beta}))\}$$

$$\tilde{\alpha} = (\alpha_1, \dots, \alpha_n) \leq \tilde{\beta} = (\beta_1, \dots, \beta_n) \iff (\forall i = \overline{1, n})(\alpha_i \leq \beta_i)$$

$$\overline{\mathcal{T}_0} \cap \overline{\mathcal{T}_1} \subseteq \overline{\mathcal{M}}$$

$$5) \mathcal{L} \Leftarrow \{f : f = \sum_{i=1}^n (mod 2) a_i x_i \oplus a_0\}$$

$$x_1 \sim x_2 = x_1 \oplus x_2 \oplus a_0 \in \mathcal{L}$$

Есть функции, которые принадлежат всем классам Поста, и есть такие, которые не принадлежат никакому.

**Лемма 1** (О несамодвойственной функции). Пусть  $f_S \notin \mathcal{S}$ . Тогда обе константы (0 и 1) представимы формулами над множеством  $\{f_S, \overline{\phantom{x}}\}$

**Доказательство.** Так как  $f_S \notin \mathcal{S}$ , то  $(\exists \tilde{\alpha} = (\alpha_1, \dots, \alpha_n))(f(\tilde{\alpha}) = f(\overline{\tilde{\alpha}}))$

Определим

$$h(x) \Leftrightarrow f_S(x^{\alpha_1}, \dots, x^{\alpha_n}); \quad h(x) = \text{const} \in \{0, 1\}.$$

Подставим 1 или 0:

$$\begin{aligned} h(0) &= f_S(0^{\alpha_1}, \dots, 0^{\alpha_n}) = f_S(\overline{\tilde{\alpha}}) \\ h(1) &= f_S(1^{\alpha_1}, \dots, 1^{\alpha_n}) = f_S(\tilde{\alpha}) \end{aligned}$$

То есть

$$h(0) = h(1) = f_S(\tilde{\alpha}) = f(\overline{\tilde{\alpha}}) \in \{0, 1\}.$$

Представим ее как отрицание:  $\overline{h(x)} \in \{0, 1\}$  - и получим вторую константу.  $\square$

**Лемма 2** (О немонотонной функции). Если функция  $f_M \notin \mathcal{M}$ , то существует два набора (вектора)  $\tilde{\alpha} = (\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n)$  и  $\tilde{\beta} = (\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n)$ , и  $f(\tilde{\alpha}) = 1, f(\tilde{\beta}) = 0$

Рассмотрим такую функцию:  $f_M = (1000\ 0011\ 1111\ 1100) \in \overline{\mathcal{T}_0} \cap \overline{\mathcal{T}_1} \implies f_M \notin \mathcal{M}$

**Лемма 3** (О немонотонной функции). Отрицание может быть представлено формулой над множеством  $\{f_M, 0, 1\}$ , где  $f_M \notin \mathcal{M}$

**Доказательство.** В силу леммы 2 берем два набора  $\tilde{\alpha}$  и  $\tilde{\beta}$ . Тогда очевидно отрицание представимо формулой

$$\overline{x} = f_M(\alpha_1, \dots, \alpha_{i-1}, x, \alpha_{i+1}, \dots, \alpha_n)$$

$$f_M(\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n) = 1 \text{ и } 0 \text{ иначе.} \quad \square$$

**Лемма 4** (О нелинейной функции). Пусть  $f_L \notin \mathcal{L}$ . Тогда конъюнкция может быть представлена формулой над множеством  $\{f_L, 0, \overline{\phantom{x}}\}$

**Доказательство.** Поскольку  $f_L$  нелинейная функция, в ее полиноме Жегалкина обязательно будет нелинейное слагаемое. Среди всех нелинейных слагаемых функции  $f_L$  выбираем самое короткое. Пусть это самое короткое слагаемое будет  $x_{i1}, x_{i2}, \dots, x_{ik}$ . ( $k \geq 2$ )

Строим новую функцию

$$f'_L = f_L \Big|_{x_j=0 \text{ при } j \neq \{i_1, i_2, \dots, i_k\}} = x_{i1}x_{i2} \dots x_{ik} \oplus a_{i1}x_{i1} \oplus a_{i2}x_{i2} \oplus \dots \oplus a_{ik}x_{ik} \oplus a_0$$

Произвольно делим переменные на две части. Мы строим функцию от двух переменных. Первая часть переменных есть  $x$ , вторая -  $y$ .

$$\chi(x, y) = f'_L \Big|_{\substack{x_{i1} = \dots = x_{i_s} = x \\ x_{i_{s+1}} = \dots = x_{i_k} = y \\ 1 \leq s < k}} = xy \oplus ax \oplus by \oplus c,$$

$$\text{где } a = \sum_{j=1}^s (\text{mod} 2) a_{i_k}, \quad b = \sum_{l=s+1}^k (\text{mod} 2) a_{i_l}, \quad c = a_0$$

Утверждается, что конъюнкция  $xy = \chi(x \oplus b, y \oplus a) \oplus ab \oplus c$ .

Посмотрим:

$$\begin{aligned} (x \oplus b)(y \oplus a) &\oplus a(x \oplus b) \oplus b(y \oplus a) \oplus c \oplus ab \oplus c = \\ xy \oplus ax \oplus by \oplus ab \oplus ax \oplus ab \oplus by \oplus ab \oplus c \oplus ab \oplus c &= xy \end{aligned}$$

Что и требовалось доказать.  $\square$

**Теорема 2.5.** Каждый класс Поста замкнут.

## 2.8 Теорема Поста

**Теорема 2.6.** *Множество булевых функций полно тогда и только тогда, когда оно не содержится (целиком) ни в одном из классов Поста.*

**Доказательство.** Необходимость. Полагая, что множество булевых функций содержится в каком-то классе Поста, получим, в силу замкнутости каждого класса Поста, что формулами над этим множеством могут быть представлены только функции этого класса, а, стало быть, не может быть представлена ни одна функция, не содержащаяся ни в одном из классов Поста, например, штрих Шеффера. Значит, такое множество не может быть полным.

Достаточность. Достаточно показать, что формулами над множеством  $\mathcal{F}$ , удовлетворяющем условию теоремы, могут быть представлены функции какого-то уже известного полного множества. В качестве такого множества можно взять такое, состоящее из конъюнкции и дизъюнкции.

Так как множество  $\{*, \bar{\phantom{x}}\}$  является полным, достаточно указать способ построения формул для конъюнкции и отрицания над базисом  $\mathcal{F}$ , который удовлетворяет условию теоремы Поста, то есть не содержится ни в одном из классов Поста, что можно выразить следующим образом:

$$(\forall C \in \{T_0, T_1, S, M, L\})(\exists f_c \in F \setminus C)$$

1 случай) Представим константу 1:

$$1 = f_0(x, \dots, x),$$

а константу 0 представим с использованием какой-нибудь функции  $g_1 \in F \setminus T_1$ :

$$0 = g(1, \dots, 1) = g(f_0(x, \dots, x), \dots, f_0(x, \dots, x))$$

Имея формулы для обеих констант, отрицание представим формулой, используя немонотонную функцию.

2 случай) Всякая функция  $f_0 \in F \setminus T_0$  не сохраняет и константу 1, а всякая функция  $f_1 \in F \setminus T_1$  не сохраняет и константу 0. В этом случае сразу получаем формулу для отрицания.

$$\bar{x} = f_0(x, \dots, x)$$

Тут используется лемма о несамодвойственной функции. □

# Элементы математической логики

## 3.1 Предпосылки возникновения математической логики

Пример Гиберта.

$Y = \{x : |x| \geq 3\}$  х - множество

То есть возьмем такие примеры и получим:

$$\{1, 2, 3\} \in Y, \{1, 2, 3, 4\} \in Y, \{1, 2, 3, 4, 5\} \in Y \implies Y \in Y$$

**Определение 54.** Нормальные множества - это такие множества, которые не содержат самих себя.

Пусть мы хотим найти все Нормальные множества:  $Z = \{x : x \notin x\}$   $Z \notin Z \implies Z \in Z \implies Z \notin Z$ .  
Это называется парадокс Рассела.

## 3.2 Понятие формальной аксиоматической теории

**Определение 55.**  $\mathcal{T} = ( \underbrace{V}_{\text{алфавит}}, \underbrace{\mathcal{F}}_{\text{формулы}}, \underbrace{\mathcal{A} \subseteq \mathcal{F}}_{\text{Мн. аксиом}}, \underbrace{\mathcal{P}}_{\text{Мн. правил вывода}} )$  называется теорией.

**Определение 56.** Фиксируется некоторое множество  $\Gamma \subseteq \mathcal{F}$  - гипотеза. Среди гипотез нет ни одной аксиомы:  $\Gamma \cap \mathcal{A} = \emptyset$ .

**Определение 57.** Вывод теории  $\mathcal{T}$  из множества гипотез  $\Gamma$  - это последовательность формул (конечная или бесконечная):  $\theta_0, \theta_1, \dots, \theta_n, \dots$ ,  $n \geq 0$ , где для каждого  $\forall i \geq 0$ : 1)  $\theta_i \in \Gamma$ , 2)  $\theta_i \in \mathcal{A}$ , 3) существует правило вывода в  $\mathcal{P}$ :  $\frac{\theta_{j_1} \dots \theta_{j_m}}{\theta_i}$ , где  $j_1, \dots, j_m < i$ .

Если  $\Phi = \theta_i$ , то  $\Gamma \vdash_{\mathcal{T}} \Phi$ . Если  $\Gamma = \emptyset$ , то пишем  $\vdash_{\mathcal{T}} \Phi$ .

**Теорема 3.1.** Если формула  $\Phi$  выводима из гипотезы  $(\Gamma \vdash_{\mathcal{T}} \Phi)$ , то для любого  $\Gamma' \supset \Gamma$  верно  $\Gamma' \vdash_{\mathcal{T}} \Phi$ .

**Следствие.** Если  $\vdash_{\mathcal{T}} \Phi$ , то для любого  $\Gamma$ :  $\Gamma \vdash_{\mathcal{T}} \Phi$ .

## 3.3 Алгебра высказываний. Тавтологии

У нас есть высказывания  $p, q, r, \dots$  и они могут принимать значения Ложь (Л) или Истина (И). Указываются по-русски, однако для упрощения разметки буду использовать F, T (False, True).

$p$	$q$	$\vee$	$\&$	$\rightarrow$	$\oplus$
F	F	F	F	T	F
F	T	T	F	T	T
T	F	T	F	F	T
T	T	T	T	T	F

Функции также аналогичны тем, что описаны в математической логике:

$$G : \{F, T\}^n \rightarrow \{F, T\}$$
$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

**Определение 58.** Тавтология - это то, что говорит само за себя

### 3.4 Исчисление высказываний

Мы строим ее на основе Теории  $L$ .

**Определение 59.** Теория  $L = (V_L, \mathcal{F}_L, \mathcal{A}_L, \mathcal{P}_L)$ . Причем  $V_L = Var \cup \{\neg, \rightarrow\} \cup Aux$ ,  $\mathcal{F}_L$  : 1) Каждая переменная есть формула, 2) Если  $\Phi$  - формула, то  $(\neg\Phi)$  - формула, 3) если  $\Phi$  и  $\Psi$  - формулы, то  $(\Phi \rightarrow \Psi)$  - формула, 4) Никаких других формул нет.

Наше "подсахаривание" формул: 1)  $\Phi \vee \Psi = \neg\Phi \rightarrow \Psi$ , 2)  $\Phi \& \Psi = \neg(\Phi \rightarrow \neg\Psi)$

Схем аксиом всего три:

$$\begin{array}{l} (1) \quad A \rightarrow (B \rightarrow A) \\ \mathcal{A}_L : (2) \quad (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) \\ (3) \quad (\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B) \end{array}$$

И наши правила вывода:

$$\mathcal{P}_L : \frac{A, A \rightarrow B}{B} \quad \text{modus ponens (MP)}$$

**Пример Тавтологии.**

$$\vdash (A \rightarrow A)$$

**Доказательство.**

1.  $A \rightarrow ((A \rightarrow A) \rightarrow A) \rightarrow ((A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A))$  - схема (2) при  $B := A \rightarrow A$ ,  $C := A$
2.  $A \rightarrow ((A \rightarrow A) \rightarrow A)$  - схема (1) при  $B := A \rightarrow A$
3.  $(A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)$  - Modus ponens к шагам (1) и (2)
4.  $A \rightarrow (A \rightarrow A)$  - схема (1) при  $B := A$
5.  $A \rightarrow A$  - modus ponens шагов (3) и (4) □

### 3.5 Теорема дедукции

**Теорема 3.2.** (Эрбрам). Пусть дано некоторое множество формул,  $A$  - произвольная формула, тогда если из  $\Gamma, A$  выводится формула  $B$  ( $\Gamma, A \vdash B$ ), то  $\Gamma \vdash (A \rightarrow B)$ .

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash (A \rightarrow B)}$$

**Пример применения.**

$$\vdash (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$$

1.  $\neg B \rightarrow \neg A$  - гипотеза
2.  $A$  - гипотеза
3.  $(\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B)$  - схема 3
4.  $(\neg B \rightarrow A)$  - МР, (1) и (3)
5.  $A \rightarrow (\neg B \rightarrow A)$  - схема 1 при  $B := \neg B$
6.  $\neg B \rightarrow A$  - МР, (2) и (5)
7.  $B$  - МР, (4) и (6)

То есть  $\neg B \rightarrow \neg A, A \vdash B$  по теореме дедукции  $\neg B \rightarrow \neg A \vdash A \rightarrow B$  по теореме дедекции  $\vdash (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$

asd