Московский государственный технический университет
имени Н. Э. Баумана
Кафедра «Программное обеспечение ЭВМ и информационные технологии»


конспект лекций

# Логика и теория алгоритмов

лектор
Алексей Иванович Белоусов

Вёрстка: Р. И. Инфлянскас
Иллюстрации: А. С. Никичкин

26 мая 2013 г.

Данный конспект записывался в «режиме реального времени» на лекциях. Лектор не несёт ответственности за неправильное толкование его лекций, ошибки и опечатки.

# Организационные вопросы

**Форма сдачи:** зачёт

**Шкала оценок:**

| | |
|---|---|
| Модули | $3 \cdot 30 = 90$ |
| Прилежание | 10 |

**Аудитория:** 226л

## Литература

1. Мендельсон. Введение в математическую логику.
2. Непейвода. Прикладная логика.
3. Катленд. Вычислимость.

# Содержание

# 1. Теория алгоритмов

## Предтечи

**Парадокс Рассела** Пусть множество $Y$ определяется следующим образом:

$$Y = \{X : |X| \geqslant 3\}$$

Это множество содержит, к примеру, множества

$$X_1 = \{a, b, c\}$$
$$X_2 = \{a, b, c, d\}$$
$$X_3 = \{a, b, c, d, e\}$$

Но тогда оно само имеет как минимум 3 элемента, а значит: $Y \in Y$.
Гилберт предложил следующее:

$Z = \{X : X \notin X\}$
$Z \in Z \Rightarrow Z \notin Z$
$Z \notin Z \Rightarrow Z \in Z$
$Z \notin Z \Leftrightarrow Z \in Z$
$Z \notin Z \Rightarrow Z \neq Z$, то есть $Z \in Z \Rightarrow Z \notin Z \Rightarrow (Z \in Z)\&(Z \notin Z)$ — *противоречие!*

**Самоприменимые прилагательные:** Самоприменимые прилагательные — прилагательные, которые описывают сами себя.

1. Трёх-слож-ный — три слога, слово описывает само себя.
2. Несамоприменимый — *противоречие*!

**Теорема Гёделя** Гёдель показал, что в теории могут быть утверждения, которые нельзя ни доказать, ни опровергнуть.

Чтобы полностью проанализировать математику, надо выйти за её пределы.

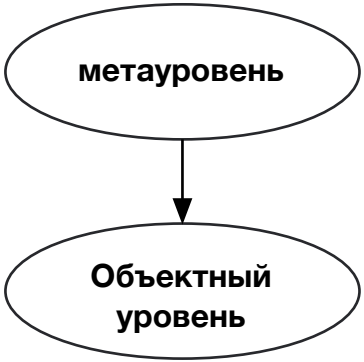


**Рис. 1.** Выход за пределы математики

Основатель теории алгоритмов — Тьюринг (работал шифровальщиком в I мировую войну). Теория алгоритмов тесно связана с криптографией.

## 1.1. Понятие алгоритма в интуитивном смысле слова

Входные данные и результат — конструктивные объекты.

Конструктивный объект — слово в конечном алфавите.

Множество $X$ — множество входных слов, $Y$ — множество выходных слов. Причём $X \subseteq V^*, Y \subseteq W^*$

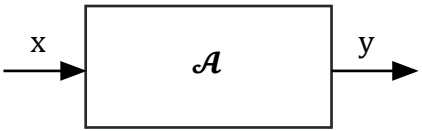


**Рис. 2.** Схема работы алгоритма

$A$ — алгоритм типа $XY$ : $\quad A : X \to Y$

$A$ — частичный алгоритм типа $XY$ : $\quad A : X \dot{\to} Y$.

Частичный алгоритм определяет частичную функцию, которая в качестве области определения использует подмножество $X$, а значения — подмножество $Y$.

Признаки алгоритма:

1. **Признак детерминированности** Алгоритм определяет детерминированный процесс.

Детерминированный процесс осуществляется за конечное число шагов, и на каждом шаге однозначно определено продолжение процесса или его прекращение.

2. **Признак массовости** Любой алгоритм может осуществлять преобразования в достаточно широком множестве слов.
3. **Признак результативности** Алгоритм должен через конечное число шагов дать определённый результат.

Словарная (вербальная) функция: $V, W \quad f : V^* \dot{\to} W^*$
Пример: $V = W \quad f(x) \rightleftharpoons xx = x^2 \quad f : V^* \to V^*$

**Функции идентификации** $x \sqsubseteq y \Leftrightarrow (\exists y_1, y_2)(y = y_1 x y_2)$ — слово $x$ входит в слово $y$.

$$g(y) = \begin{cases} \lambda, \text{ если } x \sqsubseteq y \\ y, \text{ иначе} \end{cases}$$

Пусть $A : V^* \dot{\to} W^*$. Тогда $(x \in V^*)!A(x)$ означает, что алгоритм $A$ применим к слову $x$. $\neg!A(x)$ — алгоритм $A$ не применим к слову $x$.

Результат алгоритма: $A(x) \in W^*$

**Определение 1** Вычислимость в интуитивном смысле слова. Вербальная функция называется вычислимой в интуитивном смысле слова, если существует алгоритм $A_f : V^* \to W^*$, что $(\forall x \in V^*)(!A_f(x) \Leftrightarrow x \in D(f)) \& (A_f(x) = f(x))$

## 1.2. Машина Тьюринга

Алан Тьюринг — английский математик.

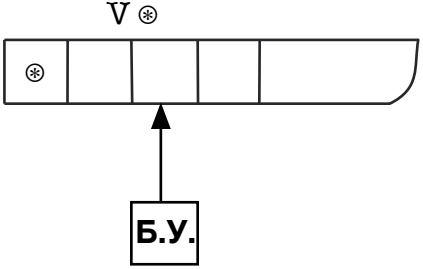


**Рис. 3.** Машина Тьюринга

Машина Тьюринга — полубесконечная лента, разделённая на буквы.

⊛ — маркер начала ленты.

□ — символ пробела.

Блок управления может находиться в любом состоянии из множества состояний $Q = q_0, \ldots, q_f$

Запись команды

$$qa \to rb, \left\{ \begin{matrix} S \\ L \\ R \end{matrix} \right\} \quad q, r \in Q,\ a, b \in V \cup \{⊛, □\}$$

означает следующее: если в состоянии $q$ обозреваемый символ $a$, то перейти в состояние $r$, записать $b$ и сдвинуться ($L$ — на символ влево, $R$ — на символ вправо, $S$ — остаться на месте).

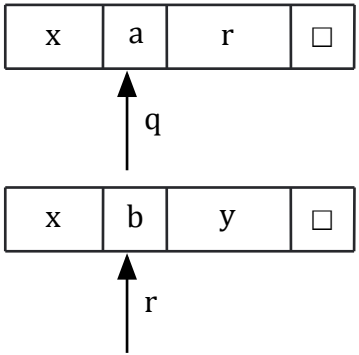


**Рис. 4.** Запись команды

Входное слово записывается на ленте без всяких пробелов буква за буквой. Первый пробел — конец слова. Потом идёт бесконечное число пробелов.

Когда машина Тьюринга даёт результат, головка останавливается на маркере начала ленты в заключительном состоянии. Сразу после этого идёт результат, потом — пробелы.

Вместо буквы после состояния может идти параметр, к примеру: $\alpha \in \{a, b, c\}$ — любой из символов $\{a, b, c\}$.

**Пример 1.** Что делает машина Тьюринга со следующей системой команд?

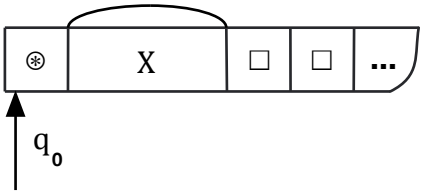


**Рис. 5.** Машина Тьюринга со входным словом

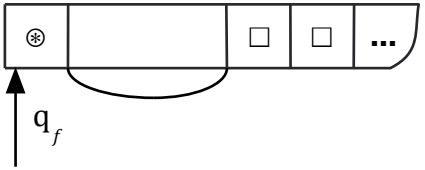


**Рис. 6.** Окончание работы машины Тьюринга

$$
\begin{array}{rl}
q_0\circledast \to q_0\circledast, R & q_2 b \to q_2 b, L \\
q_0 a \to q_0 a, R & q_2\circledast \to q_f\circledast, L \\
q_0 b \to q_0 b, R & q_1\square \to q_3\square, L \\
q_0 c \to q_1 c, R & q_3 a \to q_3\square, L \\
q_1 a \to q_1 a, R & q_3 b \to q_3\square, L \\
q_1 b \to q_1 b, R & q_3 c \to q_3\square, L \\
q_1 c \to q_1 c, R & q_3\circledast \to q_3\square, L \\
q_0\square \to q_2\square, L & q_3\circledast \to q_f\circledast, S \\
q_2 a \to q_2 a, L &
\end{array}
$$

*Ответ*: стирает слова, которые содержат букву *c*.

Формально машина Тьюринга определяется как следующий кортеж:

$$T = (V, Q, q_0, q_f, ⊛, \square, S, L, R, \delta),$$

где $\delta$ — система команд:

$$\delta : Q \times V' \to 2^{Q \times V' \times \{S,L,R\}}, \text{ где } V' = V \cup \{⊛, \square\}$$

В дальнейшем мы будем иметь дело только с детерминированными машинами Тьюринга.

В детерминированной машине Тьюринга $\delta : Q \times V' \dot{\to} Q \times V' \times \{S, L, R\}$ не может быть двух и более команд с одной и той же левой частью.

**Определение 2** Конфигурация.

$$C = (q, x, ay) \in Q \times V'^* \times V'V'^*, \text{ то есть } q \in Q,\ x, y \in V'^*,\ a \in V', \text{ где}$$

$q$ — текущее состояние, $x$ — цепочка левее головки, $a$ — символ, обозреваемый головкой, $y$ — цепочка, стоящая сразу после $a$ (с точностью до любой цепочки пробелов, стоящих в конце).

В любой машине выделяется начальная конфигурация:

$$C_0 = (q_0, \lambda, ⊛x\square)$$

и конечная конфигурация:

$$C_f = (q_f, \lambda, \circledast y \square)$$

**Определение 3** Отношение непосредственной выводимости.

$$C = (q, x, ay) \vdash_{\mathcal{T}} \begin{cases} (r, x, by), & \text{если в системе команд } \delta \text{ есть команда } qa \to rb, S \\ (r, x', cby), & \text{если в системе команд } \delta \text{ есть команда } qa \to rb, L\,(x = x'c \neq \lambda) \\ (r, xb, dy'), & \text{если в системе команд } \delta \text{ есть команда } qa \to rb, R\,(dy' = y) \end{cases}$$

**Определение 4** Выводимость на множестве конечных конфигураций машины Тьюринга.

$C_1, C_2, \ldots, C_n, \ldots \; n \geqslant 1$
$(\forall i \geqslant 1)(C_i \vdash_{\mathcal{T}} C_{i+1})$, если $C_{i+1}$ определена в последовательности
$C_1 \vdash_{\mathcal{T}} C_2 \vdash_{\mathcal{T}} \ldots \vdash_{\mathcal{T}} C_n$ — длина $= n - 1 (n \geqslant 1)$
$C \vdash^*_{\mathcal{T}} C' \rightleftharpoons$ существует вывод $C_1 \vdash_{\mathcal{T}} C_2 \vdash_{\mathcal{T}} \ldots \vdash_{\mathcal{T}} C_n = C', n \geqslant 1)$

Пусть дана машина Тьюринга $\mathcal{T}$ и слово $x \in V^*$

$$!\mathcal{T}(x) \rightleftharpoons (q_0, \lambda, \circledast x \square) \vdash^*_{\mathcal{T}} (q_f, \lambda, \circledast y \square)\; y \rightleftharpoons \mathcal{T}(x) \text{Ⓕ} \neg !\mathcal{T}(x) \rightleftharpoons [(q_0, \lambda, \ldots$$

**Определение 5.** Вербальная функция

$$f : V^* \dot{\to} V^*$$

вычислима по Тьюрингу, если может быть построена $\mathcal{T}_f$ с рабочим алфавитом $V_1 \supseteq V (\forall x \in V^*)(!\mathcal{T}_f(x) \Leftrightarrow x \in D(f)) \& (\mathcal{T}_f(x) = f(x))$

**Теорема 1** Тезис Тьюринга. *Всякая функция, вычислимая в интуитивном смысле слова вычислима по Тьюрингу.*

**Пример 2** Машина Тьюринга, стирающая всё, если есть вхождение слова.

$$\mathcal{T}_1 : \mathcal{T}_1(x) = \begin{cases} \lambda, \text{ если } aab \sqsubseteq x \\ x \text{ иначе} \end{cases} \quad , \text{где } V = \{a, b\}$$

$$\begin{aligned} q_0 \textcircled{*} &\to q_0 \textcircled{*}, R \\ q_0 a &\to q_1 a_1, R \\ q_0 b &\to q_0 b_1, R \\ q_1 a &\to q_2 a_1, R \\ q_1 b &\to q_0 b, R \\ q_2 a &\to q_2 a, R \\ q_2 b &\to q_3 b, R \end{aligned}$$

$$
\begin{aligned}
q_3\alpha &\to q_3\alpha, R, \alpha \in \{a, b\} \\
q_3\square &\to q_4\square, L \\
q_4\alpha &\to q_4\square, L, \alpha \in \{a, b\} \\
q_4\circledast &\to q_f\circledast, S \\
q_0\square &\to q_5\square, L \quad q_5 \text{ — движение в случае неуспешного поиска} \\
q_1\square &\to q_5\square, L \\
q_2\square &\to q_5\square, L \\
q_5\alpha &\to q_5\alpha, L, \alpha \in \{a, b\} \\
q_5\circledast &\to q_5\circledast, S
\end{aligned}
$$

Прогонка:

$$
\begin{aligned}
&(q_0, \lambda, \circledast aaaabab\square) \vdash (q_0, \circledast, aaaabab\square) \vdash (q_1, \circledast a, aaabab\square) \vdash (q_2, \circledast aa, aabab\square) \vdash \\
&\vdash (q_2, \circledast aaa, abab\square) \vdash (q_2, \circledast aaaa, bab\square) \vdash (q_3, \circledast aaaab, ab\square) \vdash^2 (q_3, \circledast aaaabab, \square\square) \vdash \\
&\vdash (q_4, \circledast aaabab, b\square\square) \vdash^6 (q_4, \circledast, \square) \vdash (q_4, \lambda, \circledast\square) \vdash (q_f, \lambda, \circledast\square)
\end{aligned}
$$

**Пример 3.**

$$
\mathcal{T}_2 : (q_0, \lambda, \circledast x\square) \vdash^* (q_f, \lambda, \circledast \# x\square), \quad V = V_0 \cup \{\#\}, \# \notin V_0,\ x \in V_0^*
$$

$$
\begin{aligned}
q_0\circledast &\to q_0\circledast, R \\
q_0\square &\to q_f\#, L \\
q_0\alpha &\to q_\alpha\#, R \quad \alpha \in V_0 \\
q_\alpha\beta &\to q_\beta\alpha, R \quad \alpha, \beta \in V_0 \\
q_\alpha\square &\to q_1\alpha, L \\
q_1\gamma &\to q_1\gamma, L \quad \beta \in V_0 \cup \{\#\} \\
q_1\circledast &\to q_f\circledast, S
\end{aligned}
$$

Прогонка:

$V_0 = \{a, b\} \quad (q_0, \lambda, \circledast ab\square) \vdash (q_0, \circledast, ab\square) \vdash (q_a, \circledast\#, b\square) \vdash (q_b, \circledast\#a, \square) \vdash (q_1, \circledast\#, ab\square) \vdash$
$\vdash (q_1, \circledast, \#ab\square) \vdash (q_1, \lambda, \circledast\#ab\square) \vdash (q_f, \lambda, \circledast\#ab\square)$

**Пример 4.**

$$\mathcal{T}_3 : (q_0, \lambda, \circledast\#x\square) \vdash^* (q_f, \lambda, \circledast x\square), \text{ где } x \in V_0^*,\ V = V_0 \cup \{\#\}$$

$$
\begin{aligned}
q_0\circledast &\to q_0\circledast, R \\
q_0\# &\to q_\#\#, R \\
q_\#\alpha &\to q_\alpha\#, L \\
q_\alpha\# &\to q_0\alpha, R \\
q_\#\square &\to q_1\square, L
\end{aligned}
$$

$$q_1\alpha \to q_1\alpha, L \quad \alpha \in V_0$$
$$q_1\circledast \to q_f\circledast, S$$
$$q_1\# \to q_1\square, L$$

Прогонка:

$(q_0, \lambda, \circledast\#abc\square) \vdash (q_0, \circledast, \#abc\square) \vdash (q_\#, \circledast, \#, abc\square) \vdash (q_a, \circledast, \#\#bc\square) \vdash (q_o, \circledast a, \#bc\square) \vdash$
$\vdash (q_\#, \circledast a\#, bc\square) \vdash (q_b, \circledast a, \#\#c\square) \vdash (q_0, \circledast ab, \#c\square) \vdash (q_\#, \circledast ab\#, c\square) \vdash (q_c, \circledast ab, \#\#\square) \vdash$
$\vdash (q_0, \circledast abc, \#\square) \vdash (q_\#, \circledast abc\#, \square) \vdash (q_1, \circledast abc, \#\square) \vdash (q_1, \circledast ab, c\square\square) \vdash^3 (q_1, \lambda, \circledast abc\square) \vdash$
$\vdash (q_f, \lambda, \circledast abc\square)$

**Пример 5.** Пусть требуется сдвинуть на заранее заданное количество символов влево.
Вместо:

$$q_1\alpha \to q_1\alpha, L \quad \alpha \in V_0$$
$$q_1\circledast \to q_f\circledast, S$$
$$q_1\# \to q_1\square, L$$

из предыдущего примера вставим:

$$q_1\#\square \to q_2\square, L$$
$$q_2\alpha \to \alpha, L \quad (\alpha \in V_0)$$

$$q_2\# \to q_\#\#, R$$
$$q_\#\# \to q_\#\#, R$$
$$q_2\textcircled{*} \to q_f\textcircled{*}, S$$

**Свойства модели алгоритмов** Любая модель алгоритмов должна иметь:

1. Описание модели.
2. Понятие эквивалентных алгоритмов.
3. Способы сочетания алгоритмов.
4. Универсальный алгоритм.
5. Понятие разрешимого и перечислимого множества.
6. Алгоритмически неразрешимых проблем.

## 1.3. Нормальные алгорифмы Маркова

**Определение 6** Вхождение слова.

$$V,\ x, y \in V^* \quad x \sqsubseteq y \rightleftharpoons (\exists y_1, y_2)(y = \underbrace{y_1}_{\text{левое крыло}}\ \underbrace{x}_{\text{основа}}\ \underbrace{y_2}_{\text{правое крыло}})$$

$(\forall x)(\lambda \sqsubseteq x)\&(x \sqsubseteq x)$

$x \sqsubseteq y, y \sqsubseteq z \Rightarrow x \sqsubseteq z$

$(y_1, x, y_2)$, где $y = y_1 x y_2 \quad y_1 \star x \star y_2,\ \star \notin V$

Самое левое вхождение слова пустого слова в слово $x$: $\star\star x$.

Первое (главное) вхождением слова $x$ в слово $y$ имеет наименьшую длину левого крыла среди всех вхождений.

**Определение 7** Формула подстановки.

$$\omega : u \to v, \quad u, v \in V^*,\ \to \notin V$$

**Определение 8** Применимость. Если $u \sqsubseteq x$, то говорят, что $\omega$ применима к $x$ (подходит для слова $x$).

Первое вхождение $u$ в $x$: $x_1 \star u \star x_2$, тогда

$$y \equiv x_1 v x_2 \equiv \omega x \ —$$

результат применения формулы к слову $x$, полученный путём замены первого вхождения левой части формулы правой частью.

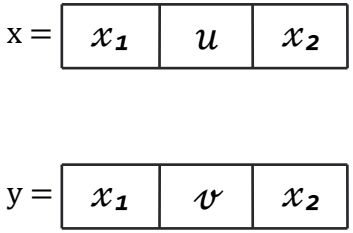


**Рис. 7.** Формула подстановки

Например, $x =$ входит, $\omega$ : вход $\rightarrow$ уход. $\omega x =$ уходит.

**Определение 9** Нормальный алгорифм.

$$\mathcal{A} = (V, \mathcal{S}, \mathcal{P})$$

$V$ — алфавит, $\mathcal{S}$ — схема, $\mathcal{P}$ — заключительные формулы.

Схема нормального алгорифма (квадратные скобки означают необязательность вхождения):

$$\begin{cases} u_1 \to[\cdot]\, v_1 \\ u_2 \to[\cdot]\, v_2 \\ \vdots \\ u_n \to[\cdot]\, v_n \end{cases}$$

**Пример 6** Добавление $aba$ в конец слова.

$$\mathcal{A}_0 : \begin{cases} \#a \to a\# \quad V = \{a, b, \#\} \\ \#b \to b\# \\ \# \to\cdot\, aba \\ \to \# \end{cases}$$

Прогонка:

$$\mathcal{A}_0 : x = aba \vdash \#aba \vdash a\#ba \vdash ab\#a \vdash aba\# \vdash\cdot\, abaaba$$

В общем случае:

$$\mathcal{A}_0 : x = x(1)x(2)\ldots x(k) \vdash \#x(1)x(2)\ldots x(k) \vdash x(1)\#x(2)\ldots x(k) \vdash$$
$$\vdash x(1)x(2)\#\ldots x(k) \vdash \ldots \vdash x(1)x(2)\ldots x(k)\# \vdash\cdot\, x(1)x(2)\ldots x(k)aba \quad (k \geqslant 1)$$

**Пример 7** Добавление $aba$ в начало слова.

$$\mathcal{A}_1 : \begin{cases} \to \cdot\, aba \\ (\forall x \in \{a,b\}^*)(\mathcal{A}_1 : x \vdash \cdot\, abax) \end{cases}$$

Пусть есть нормальный алгорифм:

$$\mathcal{A} = (V, \mathcal{S}, \mathcal{P})$$

Алгорифм $\mathcal{A}$ просто непосредственно переводит $x$ в $y$: $x \vdash y \rightleftharpoons y = \omega x$, где $\omega$ — первая входящая в $\mathcal{S}$ подходящая для $x$ формула, не являющаяся заключительной.

Алгорифм $\mathcal{A}$ непосредственно заключительно переводит $x$ в $y$: $A : x \vdash \cdot\, y \rightleftharpoons y = \omega x$, где $\omega$ — первая входящая в $\mathcal{P}$ подходящая для $x$ формула.

Алгорифм $\mathcal{A}$ просто переводит $x$ в $y$: $\mathcal{A} : x \vDash y \rightleftharpoons$ Существует последовательность слов

$$x = x_0,\ x_1,\ x_2,\ \ldots,\ x_n = y, \text{ где } (\forall i = \overline{0, n-1})(\mathcal{A} : x_i \vdash x_{i+1})$$
$$\mathcal{A} : x \vDash y \rightleftharpoons (\mathcal{A} : \vdash \cdot\, y) \rightleftharpoons (\mathcal{A} : x \vdash \cdot\, y) \vee (\exists z)(\mathcal{A} : x \vDash z \vdash \cdot\, y)$$

$!\mathcal{A}(x)$ — алгорифм $\mathcal{A}$ применим к слову $x$.

$\neg!\mathcal{A}(x)$ — алгорифм $\mathcal{A}$ не применим к слову $x$.

$\mathcal{A} : \neg x$ — слово $x$ не поддаётся схеме нормального алгорифма (нет ни одной подходящей формулы).

**Определение 10** Процесс работы нормального алгорифма со словом $x$. Это конечная или бесконечная последовательность слов:

$$x = x_0,\ x_1,\ x_2,\ \ldots,\ x_n,\ \ldots \text{ такая, что } (\forall i \geqslant 0)(\mathcal{A} : x_i \vdash x_{i+1}) \vee (\mathcal{A} : x_i \vdash\cdot\ x_{i+1}),$$

если $x_{i+1}$ определено в последовательности.

При этом слово $x_n$ не определено тогда и только тогда (по определению):

1. $n - 1 = 0$ и $\mathcal{A} : \neg x_0 = x$
2. $n > 0$ т. е. $x_{n-1}$ определено, но $\mathcal{A} : \neg x_{n-1}$
3. $\mathcal{A} : x_{n-2} \vdash\cdot\ x_{n-1}, n \geqslant 2$

Пусть $x_n = x_0, x_1, \ldots, x_n$ — процесс работы $\mathcal{A}$ с $x$ (является конечным). Тогда $x_n$ называется процессом работы нормального алгорифма $\mathcal{A}$ со словом $x$ и обозначается $\mathcal{A}(x)$.

**Определение 11.** Вербальная функция

$$f : V^* \dot{\to} V^*$$

называется вычислимой по Маркову, если может быть построен нормальный алгорифм $\mathcal{A}_f$ в алфавите $V$ такой, что

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

**Теорема 2** Принцип нормализации. *Любая вербальная функция, вычислимая в интуитивном смысле слова, вычислима по Маркову.*

**Пример 8** Правое присоединение.

$$V = \{a_1, \ldots, a_n\},\ \# \notin V$$

$$R_c : \begin{cases} \#\xi \to \xi\# & (\xi \in V) \\ \# \to\cdot\, x_0 & x_0 \text{ — произвольное фиксированное слово в } V \\ \dot{\to}\# & \end{cases}$$

**Пример 9** Удвоение слова.

$$V,\ \alpha, \beta \notin V$$

$$\mathcal{A}_2 : \begin{cases} \alpha\xi & \to \xi\beta\xi\alpha \quad (\xi \in V) \\ \beta\xi\eta & \to \eta\beta\xi \quad (\eta, \xi \in V) \\ \beta & \to \\ \alpha & \to\cdot \\ & \to \alpha \end{cases}$$

Прогонка:

$\lambda \vdash \alpha \vdash\cdot \lambda, \quad a \in V$

$a \vdash \alpha a \vdash a\beta a\alpha \vdash aa\alpha \vdash\cdot aa$

$abca \vdash \alpha abca \vdash a\beta a\alpha bca \vdash a\beta ab\beta b\alpha ca \vdash a\beta ab\beta bc\beta c\alpha a \vdash a\beta ab\beta bc\beta ca\beta a\alpha \vdash ab\beta a\beta bc\beta ca\beta a\alpha \vdash$

$\vdash ab\beta ac\beta b\beta ca\beta a\alpha \vdash abc\beta a\beta b\beta ca\beta a\alpha \vdash abc\beta a\beta ba\beta c\beta a\alpha \vdash abc\beta aa\beta b\beta c\beta a\alpha \vdash abca\beta a\beta b\beta c\beta a\alpha \vdash^4$

$\vdash^4 abcaabca\alpha \vdash \cdot abcaabca$

Таким образом действие вышеописанной модели Маркова:

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

## 1.4. Эквивалентность нормальных алгорифмов. Теорема о переводе

**Определение 12** Условное равенство.

$$\mathcal{A}, \mathcal{B} : V^* \dot{\to} V^* \quad (\forall x \in V^*)(!\mathcal{A}(x) \Leftrightarrow !\mathcal{B} \& (\mathcal{A}(x) = \mathcal{B}(x)) \rightleftharpoons \mathcal{A}(x) \approx \mathcal{B}(x)$$

**Замыкание схемы нормального алгорифма** Исходная схема:

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

Новая схема:

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

называется замыканием нормального алгорифма $\mathcal{A}$.

**Утверждение 1.**

$$(\forall x \in V^*)(A(x) \approx A^{\cdot}(x))$$

**Доказательство.** Пусть $!A(x)$, то есть

1. $A : x \vDash y,\ A : \neg y$ (есть обрыв) или
2. $\mathcal{A} : x \vDash\cdot\, y$

1. $\mathcal{A}^{\cdot} : x \vDash y \vdash\cdot\, y$, то есть $\mathcal{A}^{\cdot} : x \vDash y$;
2. $\mathcal{A}^{\cdot} : x \vDash\cdot\, y$. Если $!\mathcal{A}(x)$, то $!\mathcal{A}^{\cdot}(x)$, причём $\mathcal{A}^{\cdot} : x \vDash\cdot\, \mathcal{A}(x) = \mathcal{A}^{\cdot}(x)$. Если же $\neg!\mathcal{A}(x)$, то $\neg!\mathcal{A}^{\cdot}(x)$, то есть $!\mathcal{A}^{\cdot}(x) \Rightarrow !\mathcal{A}(x)$.

Итак, $\mathcal{A}(x) \approxeq \mathcal{A}^{\cdot}(x)$ □

Переход к замыканию нормального алгорифма позволяет без ограничений общности считать применимость алгорифма к слову означает что на последнем шаге процесса работы была применена заключительная формула, то есть исключить естественный обрыв.

### 1.4.1. Естественное и формальное распространение нормального алгорифма на более широкий алгорифм

$$A = (V, S, P), V' \supset V$$
$$A' = (V', S, P) \text{ — естественное распространение}$$
$$(\forall x \in V^*)(A(x) \approxeq A'(x))$$
$$A^f = (V', S^f, P)$$

$$S^f = \begin{cases} \xi \to \xi \quad (\xi \in V' \setminus V) \\ S \end{cases}$$

$(\forall x \in V^*)(A^f(x) \approxeq A(x))$, но $(\forall x \notin V^*)(\neg!A^f(x))$

Пусть есть алфавиты $V, V_0$:

$$V = \{a_1, a_2, \ldots, a_n\}, \quad V_0 = 0, 1, \quad V_0 \cap V = \emptyset$$

Можно закодировать буквы и слова алфавита $V$ буквами алфавита $V_0$.

$$[a_i \rightleftharpoons 0\underbrace{11\ldots1}_{i}0 \quad x \in V^*[\lambda = \lambda, [x(1)x(2)\ldots x(k) \rightleftharpoons [x(1)\ldots[x(k)$$

$V = \{a, b, c\}[abca = 0100110011100010$

**Теорема 3** о переводе. *Каков бы ни был нормальный алгорифм $A = (V', S, P)$ над алфавитом $V$ (то есть $V' \supset V$), может быть построен нормальный алгорифм $B$ в алфавите $V \cup V_0$ такой, что $(\forall x \in V^*)(A(x) \approxeq B(x))$*

## 1.5. Способы сочетаний нормальных алгорифмов

### 1.5.1. Композиция

**Теорема 4** о композиции. *Каковы бы ни были нормальные алгорифмы $A, B$, может быть построен нормальный алгорифм $C$, такой что* $(\forall x \in V^*)(C(x) \approxeq B(A(x)))$.

**Доказательство.** Определим $\overline{V} = \{\overline{a_1}, \ldots, \overline{a_n}\}$, где $V = \{a_1, \ldots, a_n\}$, и $\overline{V} \cap V = \emptyset \quad \alpha, \beta \notin V \cup \overline{V}$

$$C : \begin{cases} (1)\ \xi\alpha & \to \alpha\xi \quad (\xi \in V) \\ (2)\ \alpha\xi & \to \alpha\bar{\xi} \\ (3)\ \bar{\xi}\eta & \to \bar{\xi}\bar{\eta} \quad (\eta \in V) \\ (4)\ \bar{\xi}\beta & \to \beta\bar{\xi} \\ (5)\ \beta\bar{\xi} & \to \beta\xi \\ (6)\ \xi\overline{\eta} & \to \xi\eta \\ (7)\ \alpha\beta & \to\cdot \\ (8)\ \overline{\mathbb{B}}_{\alpha}^{\beta} & \\ (9)\ \mathbb{A}^{\alpha} & \end{cases}$$

В систему формул включаются $\overline{\mathbb{B}}_{\alpha}^{\beta}$ и $\mathbb{A}^{\alpha}$. Они получаются следующим образом:

| $\mathbb{A}^{\cdot}$ | $\mathbb{A}^{\alpha}$ |
|---|---|
| $u \to v$ | $u \to v$ |
| $u \to\cdot\, v$ | $u \to \alpha v$ |

| $\mathbb{B}^{\cdot}$ | $\overline{\mathbb{B}}^{\beta}_{\alpha}$ |
|---|---|
| $\to v$ | $\alpha \to \alpha\overline{v}$ |
| $u \underset{(u\neq\lambda)}{\to} v$ | $\overline{u} \to \overline{v}$ |
| $u \underset{(u\neq\lambda)}{\to\cdot} v$ | $\overline{u} \to \beta\overline{v}$ |
| $\to\cdot\, v$ | $\alpha \to \alpha\beta\overline{v}$ |

$x \in V^*(x \neq \lambda)$

$C : x \vDash y_1\alpha y_2$, где $y_1y_1 = A^{\cdot}(x) \quad (I)$

$y_1\alpha y_2 \vDash_{(1)} \alpha y_1y_2 = \alpha y = \alpha y(1)y(2)\dots y(m),\ m > 0 \quad (II)$

$\alpha y(1)y(2)\dots y(m) \vdash_{(2)} \alpha y(\bar{1})y(2)\dots y(m) \vDash \alpha y(\bar{1})y(\bar{2})\dots y(\bar{m}) = \alpha\bar{y} \quad (III)$

$\alpha\bar{y} \vDash_{(8)} \alpha\bar{z_1}\beta\bar{z_2}$, где $z_1z_2 \rightleftharpoons z = B^{\cdot}(y) = B^{\cdot}(A^{\cdot}(x)) \quad (IV)$

□

Если слово $y = \mathcal{A}(x) = \lambda$, то второй этап пропадёт.

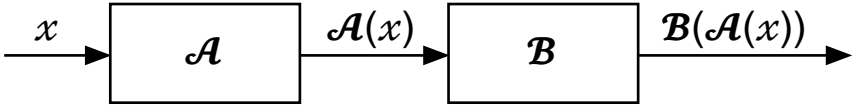


**Рис. 8.** Композиция

Композиция может обозначаться: $\mathcal{C} = \mathcal{B} \circ \mathcal{A}$.
Степень:

$$A^0 \rightleftharpoons Id,\ \mathcal{A}^n \rightleftharpoons \mathcal{A}^{n-1} \circ \mathcal{A}$$

**Пример 10.**

$$A^{\cdot}: \begin{cases} \#a \to a\# \\ \#b \to b\# \\ \# \to \cdot aba \\ \to \# \\ \to \cdot \end{cases}$$

$$B^{\cdot}: \begin{cases} \to \cdot baba \\ \to \cdot \end{cases}$$

$$C: \begin{cases} \xi\alpha \to \alpha\xi \\ \alpha\xi \to \alpha\overline{\xi} \\ \overline{\xi}\eta \to \xi\overline{\eta} \\ \overline{\xi}\beta \to \beta\overline{\xi} \\ \beta\overline{\xi} \to \beta\xi \\ \xi\overline{\eta} \to \xi\eta \\ \alpha\beta \to \cdot \\ \alpha \to \alpha\beta\overline{b}\overline{a}\overline{b}\overline{a} \\ \alpha \to \alpha\beta \\ \#a \to a\# \\ \#b \to b\# \\ \# \to \alpha aba \\ \to \# \\ \to \alpha \end{cases}$$

Работа алгорифма:

$$C: x = baa \vdash \#baa \vDash baa\# \vdash baa\alpha aba \vDash \alpha baaaba \vdash \alpha\overline{b}\overline{a}aaba \vDash \alpha\overline{b}\overline{aaa}\overline{b}\overline{a} \vdash \alpha\beta\overline{b}\overline{a}\overline{b}\overline{a}\overline{b}\overline{aaa}\overline{b}\overline{a} \vdash$$
$$\vdash \alpha\beta b\overline{a}\overline{b}\overline{a}\overline{b}\overline{aaa}\overline{b}\overline{a} \vdash \alpha\beta ba\overline{b}\overline{a}\overline{b}\overline{aaa}\overline{b}\overline{a} \vDash \alpha\beta bababaaaba \vdash \cdot\, bababaaaba$$

### 1.5.2. Объединение

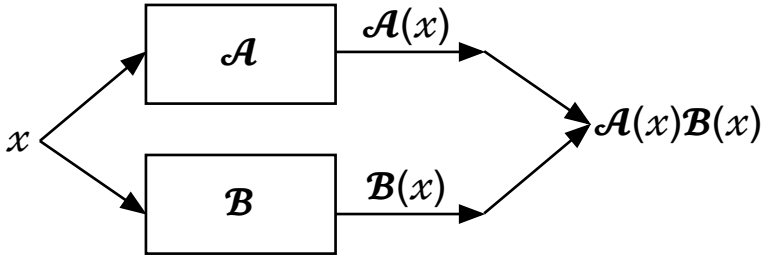


**Рис. 9.** Объединение

**Теорема 5** Объединение. *Каковы бы ни были нормальные алгорифмы $\mathcal{A}$ и $\mathcal{B}$ в алфавите $V$ может быть построен нормальный алгорифм $\mathcal{C}$ над алфавитом $V$ такой что*

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

$$\mathcal{C}(x\$y) \approxeq \mathcal{A}(x)\$\mathcal{B}(y)$$
$$x, y \in V^*, \$ \notin V$$

### 1.5.3. Разветвление

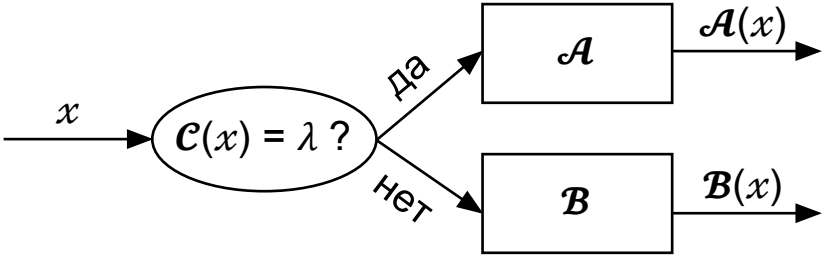


**Рис. 10.** Разветвление

```
1  if C(x) = λ then
2     y ← A(x)
3  else
4     y ← B(x)
```

**Теорема 6** Разветвление. *Каковы бы ни были нормальные алгорифмы $\mathcal{A}, \mathcal{B}, \mathcal{C}$ в алфавите $V$, может быть построен нормальный алгорифм $\mathcal{D}$ над алфавитом $V$ такой, что*

$$(\forall x \in V^*)\mathcal{D} \approx \begin{cases} \mathcal{A}(x), \text{ если } \mathcal{C}(x) = \lambda \\ \mathcal{B}(x) \text{ иначе} \end{cases}$$

### 1.5.4. Повторение

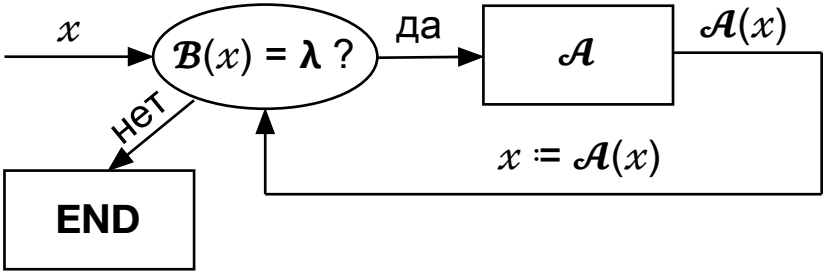


**Рис. 11.** Повторение алгорифма $A$, управляемого алгорифмом B

**Теорема 7** Повторение. *Каковы бы ни были нормальные алгорифмы $\mathcal{A}, \mathcal{B}$ в алфавите $V$, может быть построен нормальный алгорифм $\mathcal{C}$ над алфавитом $V$ такой что*

$(\forall x \in V^*)!\mathcal{C}(x) \rightleftharpoons (\mathcal{B}(x) \neq \lambda)\&(\mathcal{C}(x) = x) \vee$ (*существует последовательность слов* $x = x_0, x_1, x_2, \ldots, x_{n-1}, x_n$, *где* $[(\forall i = \overline{0, n-1})(\mathcal{B}(x_i) = \lambda)\&(x_{i+1} = \mathcal{A}(x_i))]\&(\mathcal{B}(x_n) \neq \lambda)\&$ $\&(\mathcal{C}(x) = x_n)$)

Обозначение: $\mathcal{C} = {}_{\mathcal{B}}\{A\}$

```
1  while (B(x) = λ) do
2    x ← A(x)
```

```
3   end
```

Другой вид повторения:
Обозначение: $\mathcal{C} = {}_{\mathcal{B}} < A >$

```
1   while (B(x) ≠ λ) do
2     x ← A(x)
3   end
```

**Определение 13.** Векторное слово в алфавите $V$:

$$x_1\$x_2\$\ldots\$x_n,\ n \geqslant 1,\ \$ \notin V \text{ n-ка слов}$$

**Пример 11** Проектирующие алгорифмы. $V$ — алфавит.

$$\prod_i(x_1\$x_2\$\ldots\$x_n) = x_i,\ i = \overline{1,n}$$

$$\mathcal{P}_1 = \begin{cases} \$\eta \to \$\,(\eta \in V) \\ \$ \to \\ \dot{\to} \end{cases} \qquad \mathcal{P}_1(x_1\$x_2\$\ldots\$x_n) = x_1$$

$$\mathcal{P}_2 = \begin{cases} \eta\# \to \#\,(\eta \in V,\ \# \notin V,\ \eta \neq \#) \\ \#\dot{\to} \\ \$ \to \# \end{cases} \qquad \mathcal{P}_2(x_1\$x_2\$\ldots\$x_n) = x_2\$\ldots\$x_n$$

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

**Пример 12** Распознавание равенства слов.

$$EQ(x\$y) = \lambda \Leftrightarrow x = y, \text{ где } x, y \in V^*, \$ \notin V$$
$$Inv(y) = y^R$$
$$EQ(x\$y) \cong Comp(Id(x)\$Inv(y))$$
$$Comp : \begin{cases} \eta\$\eta \to \$ \quad (\eta \in V) \\ \$ \to \cdot \end{cases}$$

## 1.6. Универсальный нормальный алгорифм

Пусть есть нормальный алгорифм $\mathcal{A}$ в алфавите $V$.

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

$$\mathcal{A}^n \rightleftharpoons u_1\alpha[\beta]v_1\gamma u_2\alpha[\beta]v_2\gamma\ldots\gamma u_n\alpha[\beta]v_n, \quad \alpha,\beta,\gamma \notin V, \text{ где}$$

$\alpha$ — стрелки, $\beta$ — подточки, $\gamma$ — разделитель между формулами.

**Пример 13.**

$$\mathcal{A}_0 : \begin{cases} \#a \to a\# & V = \{a, b, \#\} \\ \#b \to b\# & \\ \# \to\cdot\, aba & \\ \to \# & \end{cases}$$

$$\mathcal{A}_0^n = \#a\alpha a\#\gamma\#b\alpha b\#\gamma\#\alpha\beta aba\gamma\alpha\#$$

**Определение 14.** Запись нормального алгорифма — это перевод его изображения в алфавит $V_0 = \{0, 1\}$.

Обозначение: $[\![\mathcal{A}]\!]$

$$[\![\mathcal{A}_0]\!] = \langle a-1, b-2, \#-3, \alpha-4, \beta-5, \gamma-6\rangle = \underbrace{01110}_{\#}\underbrace{010}_{a}\underbrace{011110}_{\alpha}\underbrace{010}_{a}\underbrace{01110}_{\#}\underbrace{01111110}_{\gamma}\ldots$$

**Теорема 8** об универсальном нормальном алгорифме. *Каков бы ни был алфавит $V$, может быть построен нормальный алгорифм $U$ над алфавитом $V \cup V_0 \cup \{\$\}$, такой, что для любых слов $x \in V^*$ и нормального алгорифма $\mathcal{A}$ в алфавите $V$ имеет место*

$$U([\![\mathcal{A}]\!]\$x) \approxeq \mathcal{A}(x)$$

## 1.7. Разрешимые и перечислимые множества (языки)

**Определение 15.** Язык $L$ в алфавите $V^*$ называется алгорифмически разрешимым, если может быть построен нормальный алгорифм $\mathcal{A}_L$ такой, что

$$(\forall x \in V^*)(!\mathcal{A}_L(x)\&(\mathcal{A}_L(x) = \lambda \Leftrightarrow x \in L))$$

**Определение 16** Полуразрешающий алгорифм.

$$\tilde{\mathcal{A}}_L : \quad !\tilde{\mathcal{A}}_L(x) \Leftrightarrow x \in L$$

**Теорема 9.** *Если для языка невозможен полуразрешающий нормальный алгорифм, то невозможен и разрешающий.*

**Доказательство.** Пусть построен разрешающий нормальный алгорифм $\mathcal{A}_L$ для языка $L$, но невозможен полуразрешающий.

$\mathcal{B}_L \rightleftharpoons_{\mathcal{A}_L} (\mathcal{A}_L \vee Null)$, откуда $!\mathcal{B}_L(x) \Leftrightarrow x \in L$

□

**Пример 14** Язык двойных слов.

$$L = ww : w \in V^*$$

Докажем, что язык является разрешимым.

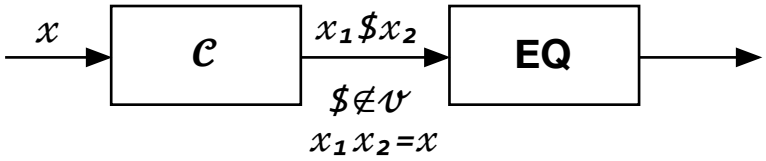


**Рис. 12.** Язык двойных слов

$x_1x_2 = x$
$|x_1| = |x_2|$, если $|x| = 2k$
$\big||x_1| - |x_2|\big| = 1$ иначе
$EQ \circ \mathcal{C}(x_1 = \lambda \Leftrightarrow x = ww$ для некоторого $w \in V^+)$

**Пример 15.**

$$R: \begin{cases} \gamma\xi \to \xi\gamma,\ \xi \in V,\ \gamma \notin V,\ \beta \notin V \\ \xi\gamma \to\cdot\ \beta\xi \\ \xi\beta \to\cdot\ \beta\xi \\ \ \to \gamma \end{cases}$$

$$L : \begin{cases} \alpha\beta \to \cdot\, \alpha\beta \ (\alpha \notin V) \\ \alpha\xi \to \cdot\, \xi\alpha \\ \to \alpha \end{cases}$$

$$\mathrm{A} : \begin{cases} \xi\alpha\beta \to \alpha\beta \\ \alpha\beta\xi \to \alpha\beta \\ \alpha\beta \to \cdot \end{cases}$$

$$\mathcal{B} : \{\alpha\beta \to \cdot\, \$$$

$$\mathcal{C} = \mathcal{B} \circ_{\mathcal{A}} < L \circ R >$$

### 1.7.1. Конструктивные числа

**Определение 17** Конструктивное натуральное число. Это слово в алфавите $V_0 = \{0, 1\}$.

1. 0 — конструктивное натуральное число.
2. Если $n$ — конструктивное натуральное число, то $n1$ — конструктивное натуральное число.
3. Других конструктивных натуральных чисел нет.

**Определение 18** Конструктивное целое число. Это слово вида $[-]n$, где $n$ — конструктивное натуральное число, то есть слово в алфавите $V_0 \cup \{-\}$

**Определение 19** Конструктивное рациональное число. Это слово вида $m/n$, где $m, n$ — конструктивное целое число ($n \neq 0$), то есть слово в алфавите $V_0 \cup \{-, /\}$

**Определение 20** Алгорифмически перечислимый язык. Язык $L \subseteq V^*$ называется алгорифмически перечислимым, если может быть построен нормальный алгорифм $\mathcal{N}_L$ такой, что $(\forall n \in$ конструктивное нормальное число$)(!\mathcal{N}_L(n) \& \mathcal{N}_L(x) \in L)$ и $\forall x \in L$ осуществимо конструктивное натуральное число $n$ такое, что $\mathcal{N}_L(n) = x$.

Нумерация:

$$\nu : \quad \mathbb{N}_0 \to A \qquad (\forall n \in \mathbb{N}_0)(\nu(n) \in A) \qquad \nu^{-1} \quad A \to \mathbb{N}_0$$

**Рис. 13.** Нумерация рациональных чисел

**Пример 16.**

$$\nu(n) = \begin{cases} -\frac{n}{2}, \text{ если } n \text{ чётное} \\ \frac{n+1}{2}, \text{ если } n \text{ нечётное} \end{cases} \qquad \nu^{-1}(n) = \begin{cases} -2x, \text{ если } x \leqslant 0 \\ 2x - 1, \text{ если } x > 0 \end{cases}$$

$$\mathcal{N}_L = {}_{\mathcal{C}}(\mathcal{A} \vee \mathcal{B})$$

$$\mathcal{C} : \begin{cases} 011 \to 0 \\ 0 \to \cdot \end{cases}$$

$\mathcal{C}(n) = \lambda \Leftrightarrow n = 2k$ ($k$ — конструктивное натуральное число)

$$\mathcal{A} \begin{cases} \alpha 11 \to 1\alpha \\ \alpha \to \cdot \\ 0 \to -0\alpha \end{cases}$$

$$\mathcal{A}(n) = -\frac{n}{2}$$

$$\mathcal{B} : \begin{cases} \alpha 11 \to 1\alpha \\ \alpha \to \cdot \\ 0 \to 0\alpha 1 \end{cases}$$

$$\mathcal{B}(n) = \frac{n+1}{2}$$

**Определение 21** Область применимости нормального алгорифма.

$$\mathcal{A} : \quad V^* \dot{\to} V^* \text{ — нормальный алгорифм над } V$$

Область применимости:

$$\mathcal{M}_{\mathcal{A}}{}^{V} \rightleftharpoons \{x :!\mathcal{A}(x),\ x \in V^*\}$$

**Теорема 10.** *Всякий алгорифмически разрешимый язык является алгорифмически перечислимым (но обратное — неверно).*

**Теорема 11** Характеристика. *Язык $L \subseteq V^*$ является перечислимым $\Leftrightarrow$ $L$ является областью применимости относительно алфавита $V$ некоторого нормального алгорифма.*

## 1.8. Проблема применимости для нормальных алгорифмов

**Частная проблема применимости** Фиксирован нормальный алгорифм $\mathcal{A}$ в алфавите $V$. Может ли быть построен нормальный алгорифм $\mathcal{B}$ над $V$ такой, что

$$(\forall x \in V^*)(!\mathcal{B}(x)\ \&\ \mathcal{B}(x) = \lambda \Leftrightarrow \neg!\mathcal{A}(x))?$$

**Общая проблема применимости** Фиксирован алфавит $V$. Может ли быть построен нормальный алгорифм $\mathcal{B}$ над $V \cup V_0$ такой, что для любых нормального алгорифма $\mathcal{A}$ в алфавите $V$ и слова $x \in V^*$

$$!\mathcal{B}([\![\mathcal{A}]\!]\$x)\ \&\ \mathcal{B}([\![\mathcal{A}]\!]) = \lambda\ \Leftrightarrow\ \neg!\mathcal{A}(x)?$$

**Проблема самоприменимости для нормальных алгорифмов** Фиксирован алфавит $V$. Может ли быть построен нормальный алгорифм $\mathcal{B}$ над $V_0$ такой, что для любого нормального алгорифма $\mathcal{A}$ в алфавите $V$

$$!\mathcal{B}([\![\mathcal{A}]\!]) \,\&\, \mathcal{B}([\![\mathcal{A}]\!]) = \lambda \;\Leftrightarrow\; \neg!\mathcal{A}([\![\mathcal{A}]\!])?$$

**Определение 22** Самоприменимый нормальный алгорифм. Нормальный алгорифм называется самоприменимым, если он применим к собственной записи. Иначе он называется несамоприменимым. В дальнейшем, будем предполагать, что алгорифмы будут рассматриваться на алфавите $V \cup V_0$.

**Пример 17** Самоприменимый алгорифм.

$$\mathcal{A}_0 : \begin{cases} \#a \to a\# \\ \#b \to b\# \\ \# \to\cdot\, aba \\ \to \# \\ \to\cdot \end{cases}$$

$$V = \{a, b, \#\}$$

$$\mathcal{A}_0 :\ [\![\mathcal{A}_0]\!] \vdash \#[\![\mathcal{A}_0]\!] \vdash\cdot\, aba[\![\mathcal{A}_0]\!]$$

**Лемма 1.** *Невозможен нормальный алгорифм $\mathcal{B}$ в алфавите $V \cup V_0$ такой, что для любого нормального алгорифма $\mathcal{A}$ в $V \cup V_0$ имеет место условие:*

$$\mathcal{B}([\![\mathcal{A}]\!]) \Leftrightarrow !\mathcal{A}([\![\mathcal{A}]\!])$$

**Доказательство.** При $\mathcal{A} = \mathcal{B}$ получаем

$$!\mathcal{B}([\![\mathcal{B}]\!]) \Leftrightarrow \neg !\mathcal{B}([\![\mathcal{B}]\!])$$

$$V, V_0 = \{0,1\},\ V_0 \cap V_0 = \emptyset \quad f : (V \cup V_0)^* \dot{\to} (V \cup V_0)^*,\ V_1 = V \cup V_0 \cup \{\alpha, \beta\}$$

□

Можно ли построить нормальный алгорифм $\mathcal{B}$ над $V_0$ такой, что для любого нормального алгорифма $\mathcal{A}$ в $V_1$ имеет место $!\mathcal{B}([\mathcal{A}]) \Leftrightarrow \neg !\mathcal{A}([\mathcal{A}])$?

**Теорема 12.** *Невозможен нормальный алгорифм $\mathcal{B}$ над алфавитом $V_0$ такой, что для любого нормального алгорифма $\mathcal{A}$ в алфавите $V_1$ имеет место*

$$!\mathcal{B}([\mathcal{A}]) \Leftrightarrow \neg !\mathcal{A}([\mathcal{A}])$$

**Доказательство.** Допустим, что алгорифм $\mathcal{B}$ может быть построен.

По теореме о переводе может быть построен нормальный алгорифм $\mathcal{B}_1$ в алфавите $V_2 \rightleftharpoons V_0 \cup \{\alpha, \beta\}$ $(\alpha, \beta \notin V_0)$ такой, что $(\forall x \in V_0^*)(\mathcal{B}_1(x) \cong \mathcal{B}(x))$

Тогда, если $\mathcal{B}_1'$ — распространение $\mathcal{B}_1$ на алфавите $V_1 = V_2 \cup V$, то оказывается, что может быть построен нормальный алгорифм $\mathcal{B}_\infty$ в $V_1$ такой, что для любого нормального алгорифма $\mathcal{A}$ в $V_1$ имеет место

$$!\mathcal{B}_1'([\mathcal{A}]) \Leftrightarrow \neg!\mathcal{A}([\mathcal{A}])$$

$$V_1 = \underbrace{V \cup \{\alpha, \beta\}}_{V'} \cup V_0$$

Итак, алгорифм $\mathcal{V}_\infty'$ решает проблему самоприменимости в алфавите $V_1$ тем самым в некотором фиксированном алфавита $V_0$, что невозможно в силу ранее доказанной леммы. □

**Следствие 1.** *Проблема самоприменимости нормальных алгорифмов алгорифмически неразрешима. Язык, состоящий из самоприменимых записей не разрешим алгорифмически.*

**Теорема 13.** *Может быть построен нормальный алгорифм $\mathcal{A}$ в алфавите $V_2 = V_0 \cup \{\alpha, \beta\}$ такой, что невозможен нормальный алгорифм $\mathcal{B}$ над $V_2$, для которого имеет место условие:*

$$(\forall x \in V_2^*)(!\mathcal{B}(x) \Leftrightarrow \neg!\mathcal{A}(x))$$

**Доказательство.** По теореме об универсальном нормальном алгорифме построим нормальный алгорифм $\mathcal{U}$ так, что $(\forall y \in V_2^*)$ и любого нормального алгорифма $\mathcal{C}$ в алгорифме $V_2$ имеет место $\mathcal{U}([\mathcal{C}]\$y) \approx \mathcal{C}(y)(\$ \notin V_2))$. Строим нормальный алгорифм $\mathcal{U}_1$ так, что $(\forall y \in V_2^*)(\mathcal{U}_1(y) \approx \mathcal{U}(y\$y))$. Можно определить $\mathcal{U}_1 = \mathcal{U} \circ Double^{\$}(Double^{\$}(y) = y\$y)\, \mathcal{U}, \mathcal{U}_1$ — алгорифмы над алфавитом $V_2$. Всякое расширение алфавита $V_2$ есть расширение алфавита $V_0$, а по теореме о переводе оно может быть сведено к двухбуквенному расширению алфавита $V_0$ то есть к алфавиту $V_2$. Тем самым любой алгорифм над $V_2$ может быть заменён вполне эквивалентным ему относительно

алфавита $V_0$ нормальным алгорифмом в алфавите $V_2$. Может быть построен нормальный алгорифм $\mathcal{A}$ в $V_2$ так, что $(\forall x \in V_0^*)(\mathcal{A}(x) \approxeq \mathcal{U}_1(x))$. Утверждается, что нормальный алгорифм $\mathcal{A}$ и есть искомый алгорифм.

Рассуждаем от противного. Предположим, что может быть построен нормальный алгорифм $\mathcal{B}$ такой, что: $(\forall x \in V_2^*)(!\mathcal{B}(x) \Leftrightarrow \neg!\mathcal{A})$. $x = [\![\mathcal{C}]\!]$, тогда

$$!\mathcal{B}([\![\mathcal{C}]\!]) \Leftrightarrow \neg!\mathcal{A}([\![\mathcal{C}]\!]) \Leftrightarrow \neg!\mathcal{U}_1([\![\mathcal{C}]\!]) \Leftrightarrow \neg!\mathcal{U}([\![\mathcal{C}]\!]\$[\![\mathcal{C}]\!]) \Leftrightarrow \neg!\mathcal{C}([\![\mathcal{C}]\!])$$

Поскольку алгорифм $\mathcal{B}$ может быть заменён вполне эквивалентным ему относительно $V_0$ нормальным алгорифмом, то алгорифм $\mathcal{B}$ решает проблему самоприменимости в алфавите $V_2$, что невозможно. □

**Следствие 2.** *Проблема применимости для нормальных алгорифмом алгорифмически неразрешима.*

**Следствие 3.** *Существуют алгоритмически перечислимые языки не являющиеся алгоритмически разрешимыми. Таковыми будут области применимости нормальных алгорифмов с неразрешимой проблемой применимости.*

**Проблема соответствий Поста** Предположим $V = \{a_1, \ldots, a_n\}$. Рассмотрим конечное бинарное отношение $\rho \subseteq V^+ \times V^+$. $\$, \# \notin V$.

$$L_\rho \rightleftharpoons \{x_1\#y_1\$x_2\#y_2\$\ldots\$x_n\#y_n : n \geqslant 1,\ (\forall i = \overline{1,n})((x_i, y_i) \in \rho),\ x_1x_2\ldots x_n = y_1y_2\ldots y_n\}$$

Проблема соответствий Поста ставится так: для любого заданного наперёд отношения $\rho$ выяснить, является ли пустым язык $L_\rho$.

$V = \{a, b\}, \quad \rho = \{(aba, ab), (b, ab)\}$. Удовлетворяет: $aba\#ab\$b\#ab$.

**Теорема Райса** Предположим, что есть множество $\mathcal{F}$ — нормально вычислимые по Маркову словарные функции. Причём, $\mathcal{F} \neq \emptyset,\ \mathcal{F} \neq U$ (оно нетривиально). Рассмотрим записи нормальных алгорифмов:

$V, \quad [\![\mathcal{A}]\!], \quad \varphi_{[\![\mathrm{A}]\!]}$ — функция, которая вычисляет нормальный алгорифм $\mathcal{A} \quad L \rightleftharpoons \{[\![\mathcal{A}]\!] : \varphi_{[\![\mathcal{A}]\!]} \in \mathcal{F}\}$

**Теорема 14** Райса. *Язык $L$ алгоритмически неразрешим.*

## 1.9. Рекурсивные функции

Базовые функции:

1. $(\forall x \in \mathbb{N})\ \mathbb{O}(x) = 0)$
2. $(x \in \mathbb{N})\ +\!\!\not|(x) \rightleftharpoons x + 1$
3. $\prod_i(x_1, \ldots, x_i, \ldots, x_n) \rightleftharpoons x_i,\ i = \overline{1,n}$

Правила:

1. Подстановка. $f(x_1, \ldots, x_n),\ g_1, \ldots, g_n; \quad f(g_1, \ldots, g_n)$, где $f : \mathbb{N}^n \to \mathbb{N} \quad g_i : \mathbb{N}^{M_i} \to \mathbb{N}$.
2. Рекурсия. $\tilde{x} = (x_1, \ldots, x_n) \quad f(\tilde{x})(f : \mathbb{N}^n \to \mathbb{N}).\ g = g(\tilde{x}, y, z) \quad h(\tilde{(x)}, 0) = f(\tilde{x}) \quad g(\tilde{x}, y, h(\tilde{x}, y)) \rightleftharpoons h(\tilde{x}, y + 1)$

$$f(x_1, \ldots, x_n) \rightleftharpoons f(\tilde{x}),\ \tilde{x} = (x_1, \ldots, x_n) \quad g(\tilde{x}, y, z))$$

$$h(\tilde{x}, y+1) \rightleftharpoons g(\tilde{x}, y, h(\tilde{x}, y))$$

При $n = 0$:

$$f(\tilde{x} \rightleftharpoons a, \quad g(y, z) \quad h(y+1) \rightleftharpoons g(y, h(y)))$$

**Сложение**

$$x + y = ?$$

1. $x + 0 \rightleftharpoons x = h(x, 0)$
2. $x + (y+1) \rightleftharpoons (x+y) + 1$

**Усечённое вычитание**

$$y \dot{-} 1 \rightleftharpoons \begin{cases} y - 1, \text{ если } y > 0 \\ 0 \text{ иначе} \end{cases}$$

$$x \dot{-} (y+1) \rightleftharpoons (x \dot{-} y) \dot{-} 1, \quad x \dot{-} 0 \rightleftharpoons x \quad g(x, y, z) = z \dot{-} 1$$

**Модуль разности**

$$|x - y| \rightleftharpoons (x \dot{-} y) + (y \dot{-} x)$$

**Умножение**

$$x \cdot 0 \rightleftharpoons 0, \quad x \cdot (y+1) \rightleftharpoons x \cdot y + x, \quad g(x, y, z) = z + x$$

**Факториал**

$$0! = 1, \quad (y+1)! \rightleftharpoons y!(y+1) \quad g(x,y,z) = z(y+1)$$

$$f(\tilde{x}, y) \quad g(\tilde{x}) \rightleftharpoons \mu y \cdot (f(\tilde{x}, y) = 0) \ (\mu \text{ — оператор минимизации})$$

$$g(x) \rightleftharpoons \mu y \cdot (\left|x - y^2\right| = 0) \quad g(x) = \begin{cases} \sqrt{x}, \text{ если } x \text{ — полный квадрат} \\ \text{не определено иначе} \end{cases}$$

**Определение 23.** Рекурсивной называется функция типа

$$f : \mathbb{N}^p \dot{\to} \mathbb{N} \quad (p \geqslant 1)$$

которая может быть получена из исходных (базовых) функций при помощи подстановки, рекурсии, минимизации. Если не используется $\mu$ оператор, то получим примитивно рекурсивную функцию.

В теории рекурсивных функций вычислимость в интуитивном смысле слова обожествляется с частичной рекурсивностью.

$$0 : \begin{cases} 01 \to 0 \\ 0 \to \cdot\, 0 \end{cases}$$

$$(\forall n)(0(n) = 0)$$

$$+1 : \left\{ 0 \to \cdot\, 01 \right.$$

$$\prod_i (x_1 \$ x_2 \$ \ldots \$ x_n) = x_i$$
$$(\forall i = \overline{1,n})(x_i - \text{КНЧ})$$

**Композиция (подстановка)**

$$f(x_1, \ldots, x_n), \quad g_i : \mathbb{N}^{m_i} \dot{\to} \mathbb{N} \quad (i = 1, \ldots, n$$

Если предположить что

$$f \mapsto A_f, \quad g_i \mapsto A_{y_i}$$

то

$$f(g_1, \ldots, g_n) \mapsto A_f(A_{g_1}\tilde{x}_1 \$ \ldots \$ A_{g_n}(\tilde{x}_n)), \quad \$ \notin V_0$$

**Рекурсия**

$$f \mapsto A_f; \quad g \mapsto A_g$$

$$h(\tilde{x}, 0) \rightleftharpoons A_f(\tilde{x}\$0) \approxeq A_n(\tilde{x}\$0)$$
$$h(\tilde{x}, y+1) \rightleftharpoons A_g(\tilde{x}\$y\$A_h(\tilde{x}\$y))$$

**Минимизация**

```
1  while (A_f(x̃$y) ≠ 0) do
2      y ← y + 1
3  end
```

**Теорема 15.** *Всякая рекурсивная функция нормально вычислима.*

**Теорема 16.** *Всякая нормально вычислимая функция на множестве натуральных чисел может быть определена как рекурсивная.*

**Следствие 4.**

$$\mathcal{N} = \mathcal{R},$$

*где $\mathcal{N}$ — класс нормально вычислимых функций, $\mathcal{R}$ — класс рекурсивных функций.*

## 1.10. λ-исчисление

Пусть есть функция $f(x, y)$. Фиксируем значение $x$ ($x$ — параметр). Это обозначается: $\lambda y \cdot f(x, y) \approxeq \lambda y \cdot f_x(y)$. Получаем отображение $x \mapsto f_x$ и оператор $f^*(x) = f_x$. $f^*(x)(y) \approxeq f(x, y)$.

**Примеры**

$$\lambda x \cdot (x^2+3)a \rhd_\beta a^2+3; \quad \lambda x \cdot (x^2+3) \rhd_\beta 12$$
$$\lambda y \cdot (x^2+y^2+3)a \rhd_\beta x^2+a^2+3$$
$$\lambda xy \cdot f(x,y) \equiv \lambda x \cdot (\lambda y \cdot f(x,y))$$

**$\lambda$-терм** $X$ — множество переменных. $X = \{x_1, \ldots, x_n, \ldots\}$.

**Определение 24** $\lambda$-терм.

1. Всякая переменная из $X$ есть $\lambda$-терм.
2. Если $M$ и $N$ — $\lambda$-терм, то $MN$ — $\lambda$-терм (аппликация).
3. Если $x \in X$, $M$ — $\lambda$-терм, то $\underbrace{\lambda x}_{\lambda\text{-астракция}} \cdot \underbrace{M}_{\text{область действия}}$ — $\lambda$-терм (абстракция).
4. Других $\lambda$-термов не существует.

**Определение 25** Свободное вхождение. Вхождение переменной $x$ в терм $M$ называется свободным, если оно не лежит в области действия $\lambda$-оператора по этой переменной

Пример:

$$P \equiv (\lambda v \cdot x)(\lambda y \cdot yx(\lambda x \cdot yvx))$$

Говорят, что терм $N$ свободен для переменной $x$ в терме $P$, если ни одно свободное вхождение $x$ в $P$ не лежит в области действия $\lambda$-оператора по переменной терма $N$.

**Рис. 14**

$$P \equiv \lambda y \cdot y \underbrace{x}_{\text{своб}} \quad N = y$$

$$\lambda xyz \cdot M \equiv \lambda x \cdot (\lambda y \cdot (\lambda z \cdot M)) \quad MNPQ \equiv ((MN)P)Q$$

**Подстановка** $M$ — $\lambda$-терм, $N$ — $\lambda$-терм, $x \in X$.

Результат подстановки терма $N$ на место всех свободных переменной $x$ в терм $M$:

$$[N|x]M \text{или} [x := N]M$$

$$M = \lambda x \cdot (xyz) \quad N = uv \quad [N|y]M \equiv \lambda x \cdot (x \overbrace{uv}^{N} z)$$

Множество свободных переменных терма $N$: $FV(N)$.

Правила:

1. $[N|x]x \equiv N$
2. $[N|x]y \equiv (y \not\equiv x)$
3. $[N|x]PQ \equiv [N|x]P[N|x]Q$
4. $[N|x](\lambda x \cdot P) \equiv \lambda x \cdot P$
5. $[N|x](\lambda y \cdot P) \equiv \lambda y \cdot [N|x]P$ при $y \notin FV(N)$ или $x \notin FV(P)$. Если $x \notin FV(P)$, то $\lambda y[N|x] \equiv \lambda y \cdot P$

**Пример 18.**

$$\lambda y \cdot x \text{ и } \lambda v \cdot x$$

$$(\lambda y \cdot x)A \rhd_\beta x$$

$$[v|x]\lambda y \cdot x \equiv \lambda y \cdot v, \quad [v|x]\lambda v \cdot v \equiv \lambda v \cdot v$$

$$f)[N|x]\lambda y \cdot P \equiv \lambda z \cdot [N|x][z|y]P, \text{ если } x \in FV(P) \text{и} y \in FV(N)$$

**Определение 26** Понятие $\lambda$-конвертируемости. Терм $M$ $\alpha$ конвертируется в терм $N$, тогда и только тогда, когда два терма различаются только обозначением связанных переменных.

$$M =_\alpha N$$

$$\lambda x \cdot M = \alpha\lambda y \cdot [y|x]M$$

### 1.10.1. $\beta$-редукция

**Определение 27** $\beta$-редекс. Применить функцию $M$ к терму $N$.

$$(\lambda x \cdot M)N \rhd_\beta [N|x]M$$

**Теорема 17** Чёрча-Росса. *Если для двух термов $M$ имеет место $M \rhd_\beta P$ и $M \rhd_\beta Q$, то существует единственный (по модулю $=_\alpha$) терм $T$ такой, что $P \rhd_\beta T$ и $Q \rhd \beta T$.*

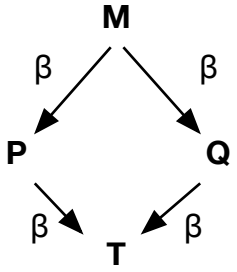


**Рис. 15.** Теорема Чёрча-Росса

**Определение 28.** Терм, не содержащий редексов называется приведённым к нормальной форме.

**Пример 19.**

$$(\lambda x \cdot xx)(\lambda y \cdot y)z \rhd_\beta (\lambda y \cdot y)(\lambda y \cdot y)z \rhd_\beta (\lambda y \cdot y)z \rhd_\beta z$$

**Пример 20** Неприводимый к нормальной форме терм.

$$(\lambda x \cdot xx)(\lambda x \cdot xx) \rhd_\beta (\lambda x \cdot xx)(\lambda x \cdot xx) \rhd_\beta \ldots$$

**Пример 21.**

$$(\lambda xy \cdot M)XY \equiv ((\lambda x \cdot (\lambda y \cdot M))X)Y \rhd_\beta ([X|x]\lambda y \cdot M)Y \equiv (\lambda y \cdot [X|x]M)Y \rhd_\beta$$
$$\rhd_\beta [Y|y][X|x]M$$

### 1.10.2. Комбинаторы

**Определение 29** Комбинатор. $\lambda$-терм, не содержащий свободных переменных, называется комбинатором (замкнутый терм).

1. $K \equiv \lambda xy \cdot x$

   $$KXY \equiv (\lambda xy \cdot x)XY \rhd_\beta [Y|y][X|x]x \rhd_\beta X$$

   Комбинатор истинности: $K \equiv T$
2. Ложь: $F \equiv \lambda xy \cdot y, \quad FXY \rhd_\beta Y$

   $$M_0 \equiv M\mathbb{T}, \quad M_1 \equiv M\mathbb{F}$$

3. $F \equiv \overline{0}$
4. Тождественная функция: $\mathbb{I} \equiv \lambda x \cdot x; \quad \mathbb{I}X \equiv (\lambda x \cdot x)X \rhd_\beta X$
5. $\overline{n} \equiv \lambda xyx^n y$, где $\underbrace{((\ldots(x)\ldots)x)x}_{n} = x^n$; $\overline{n}XY \equiv (\lambda xy \cdot x^n y)XY \rhd_\beta \mathrm{X}^n Y$
6. Прибавление единицы:Ⓕ

   $$\overline{\sigma} \equiv \lambda uxy \cdot x(uxy)$$

7. Упорядоченная пара

$$< M, N > \equiv \lambda z \cdot (zMN)$$
$$< M, N >_0 \equiv (\lambda z \cdot (zMN))\mathbb{T} \rhd_\beta \mathbb{T}MN \rhd_\beta \mathrm{M}$$
$$< M, N >_1 \equiv (\lambda z \cdot (zMN))\mathbb{F} \rhd_\beta \mathbb{F}MN \rhd_\beta N$$

8. Кортеж Ⓕ

$$< M_0, M_1, \ldots, M_n > \equiv \lambda z \cdot (zM_0M_1 \ldots M_n)$$
$$P_i^n < M_0, M_1, \ldots,$$

9. $\mathbb{B} \equiv \lambda xyz \cdot x(yz)Ⓕ$

**Пример 22.**

$$(\lambda xy \cdot Love(x, y))JohnMary$$

**Определение 30** $\lambda$-определимая функция. $f : \mathbb{N}^p \dot{\to} \mathbb{N}$ называется $\lambda$-определимой, если может быть построен $\lambda$-терм $M$ такой, что для любых $n_1, \ldots, n_p \in \mathbb{N}$, $\overline{n}_1\overline{n}_2 \ldots \overline{n}_p \rhd_\beta \overline{f}(n_1, n_2, \ldots, n_p)$, если $f(n_1, n_2, \ldots, n_p)$ определено; и терм $M\overline{n}_1\overline{n}_2 \ldots \overline{n}_p$ не имеет $\beta$-нормальной формы, если $f(n_1, n_2, \ldots, n_p)$ не определена.

**Теорема 18** Основная. *Функция $\lambda$-определима, если она рекурсивна.*

$$\lambda y \cdot P(x, y) = \lambda y \cdot P_x(y)$$

$$P^* : x \mapsto P_x$$

Условие корректности смешанного вычисления:

$$P(x, y) \cong P_x(y)$$

Режим работы чистого интерпретатора:

$$Int(P, x) \cong P(x)$$

Если данные неизвестны, то получаем объектный код:

$$\lambda x.Int(P, x) = \lambda x.Int_P(x) \quad Int^* : P \mapsto Int_P$$

Получение транслятора:

$$\lambda Px.mix(Int, (P, x)) = \lambda$$

$$\lambda IntPx.mix(mix, (Int, (P, x))) = \lambda IntPx.mix_{mix}(Int, (P, x)) \quad mix^* : mix \mapsto mix_{mix}$$

# 2. Булевы функции

## 2.1. Булевы алгебры

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

**Аксиомы симметричного полукольца**

1. $a + (b + c) = (a + b) + c$
2. $a + b = b + a$
3. $a + a = a$
4. $a + 0 = a$
5. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
6. $a \cdot 1 = 1 \cdot a = a$
7. $a(b + c) = ab + ac$
8. $a \cdot 0 = 0 \cdot a = 0$
9. $ab = ba$
10. $aa = a (a^2 = a)$
11. $a + 1 = 1$
12. $a + bc = (a + b)(a + c)$

| | | |
|---|---|---|
| 1 | $a+(b+c)=(a+b)+c$ | $a(b+c)=ab+ac$ |
| 2 | $a+b=b+a$ | $ab=ba$ |
| 3 | $a+a=a$ | $a^2=a$ |
| 4 | $a(b+c)$ | $a+bc=(a+b)(a+c)$ |
| 5 | $a+0=a$ | $a\cdot 1=1\cdot a=a$ |
| 6 | $a+1=1$ | $a\cdot 0=0\cdot a=0$ |

**Теорема 19** принцип двойтсвенности. *Каждое тождество симметричного полукольца остаётся справедливо, если в нём все знаки сложения заменить знаками умножения и наоборот, все нули заменить единицами и наоборот.*

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

**Примеры**

1. $\mathcal{S}_M=(2^M,\cup,\cap,\emptyset,M);\quad \mathcal{S}_M^*=(2^M,\cap,\cup,M,\emptyset)$
2. $\mathcal{S}_{[a,b]}=([a,b],max,min,a,b);\mathcal{S}_{[a,b]}^*=([a,b],min,max,b,a);$
3. $\mathcal{B}=(\{0,1\},+,\cdot,0,1);\mathcal{B}^*=(\{0,1\},\cdot,+,1,0);$
4. $\mathcal{D}_m=(Div(m),\text{НОК},\text{НОД},1,m);\mathcal{D}_m=(Div(m),\text{НОД},\text{НОК},m,1);$

$a(a+b)=a+ab=a$

**Доказательство.**

$$a(a+b) = a^2 + ab = a + ab = a(1+b) = a \cdot 1 = a$$

□

$a \leqslant b \Leftrightarrow ab = a$

**Доказательство.** $a \leqslant b \Rightarrow a + b = b \Rightarrow ab = a(a+b) = a$
$ab = a \Rightarrow a + b = ab + b = (a+1)b = 1 \cdot b = b$ □

$(\forall a)(a \leqslant 1)$

**Доказательство.** $a + 1 = 1 \Rightarrow a \leqslant 1$ □

**Определение 31** Дополнение.

$$a + a' = 1,\ a \cdot a' = 0$$

Не у всех элементов симметричного полукольца есть дополнения.

**Теорема 20.** *Если в симметричном полукольце элемент $a$ имеет дополнение, то оно определено однозначно.*

**Доказательство.**

$$\overline{a} : a + \overline{a} = 1, \quad a \cdot \overline{a} = 0$$

Пусть $(\exists x)(a + x = 1 \& a \cdot x = 0)$

$$x = x + 0 = x + a \cdot \overline{a} = (x + a)(x + \overline{a}) = 1 \cdot (x + \overline{a}) = (a + \overline{a})(x + \overline{a}) = ax + \overline{a} = 0 + \overline{a} = \overline{a}$$

□

**Определение 32** Булева алгебра. Симметричное полукольцо, в котором каждый элемент имеет дополнение, называется булевой алгебры.

Примеры: $\mathcal{S}_M, \mathcal{B}$ — булевы алгебры. $\mathcal{D}_m$ — булева алгебра $\Leftrightarrow q_1 q_2 \dots q_l$.
В булевой алгебре используются такие обозначения: $+ \to \vee, \cdot \to \wedge$,

$$\mathrm{B} = (B, \vee, \wedge, , 0, 1)$$

1. $(B, \vee, \wedge, 0, 1)$ — симметричное полукольцо.
2. $a \vee \overline{a} = 1, a \vee \overline{a} = 0$.

$\mathfrak{B} = (\{0,1\}, \vee, \wedge, 0, 1, )\ \overline{0} = 1, \overline{1} = 0$

$$\tilde{\alpha} \vee \tilde{\beta} = (\alpha_1 \vee \beta_1, \dots, \alpha_n \vee \beta_n)$$
$$\tilde{\alpha} \wedge \tilde{\beta} = (\alpha_1 \wedge \beta_1, \dots, \alpha_n \wedge \beta_n)$$
$$\overline{\tilde{\alpha}} \rightleftharpoons ()$$

**Определение 33** Булев куб. $\mathcal{B}^n = (\{0,1\}^n, \vee, \wedge, \tilde{0}, \tilde{1})$
$\tilde{\alpha} \leqslant \tilde{\beta} \rightleftharpoons \tilde{\alpha} \vee \tilde{\beta} = \tilde{\beta}$
$\tilde{\alpha} = (\alpha_1, \dots, \alpha_n) \leqslant \tilde{\beta} = (\beta_1, \dots, \beta_n) \Leftrightarrow (\forall i = \overline{1, n})(\alpha_i \leqslant \beta_i)$, где $0 < 1$

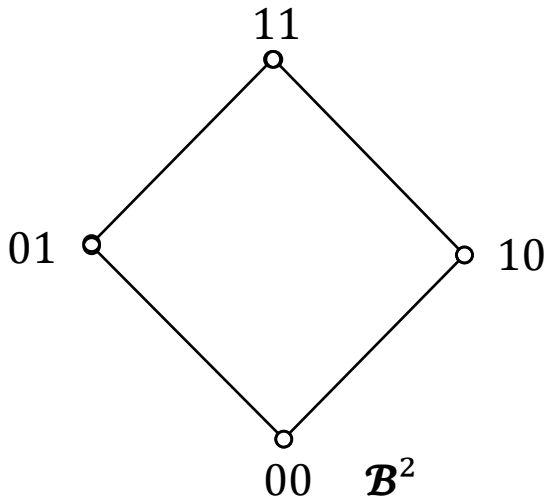


**Рис. 16.** Булев квадрат

$$\mathrm{B} = (B, \vee, \wedge, , 0, 1)$$

$$f : X \to B$$

$$(f \vee g)(x) \rightleftharpoons f(x) \vee g(x),\ (f \wedge g)(x) = f(x) \wedge g(x)$$

## 2.2. Булевы функции. Основные понятия, таблица булевой функции

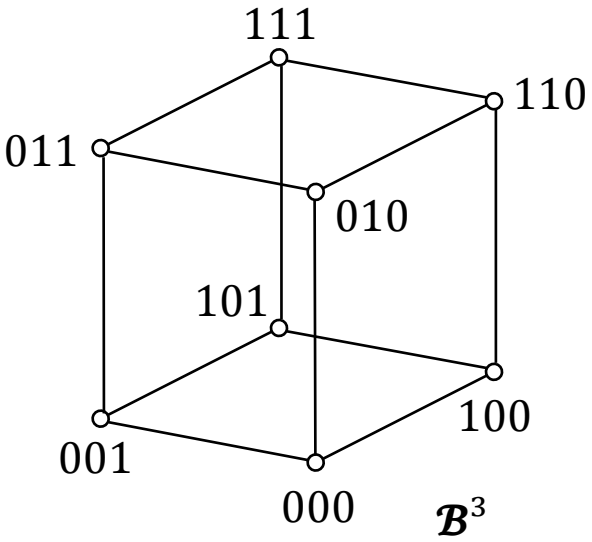


**Рис. 17.** Булев куб

**Определение 34** Булева функция от n переменных. Булева функция от n переменных — это отображение вида

$$f : \{0,1\} \to \{0,1\}$$

Может быть записана в виде: $y = f(x_1, \ldots, x_n)$.

Каждая булева функция от $n$ переменных — это $n$-арная функция на множестве $\{0,1\}$. Любая булева функция — конечная функция.

Количество булевых функций от $n$ переменных: $\left|\mathcal{P}^{(n)} = 2^{2^n}\right|$.

$n = 0 : 0, 1$

$n = 1 :$

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |

$n = 2 :$

| | $x_1$ | $x_2$ | $\vee$ | $\wedge$ | $\to$ | $\sim$ | $\oplus$ | $\mid$ | $\downarrow$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

**Определение 35.** *Любой набор, на котором функция принимает значение, равное 1, называется конституентой единицы.*

Сокращённый способ записи. Конституенты единицы:

$$f = \{3, 5, 6, 7\}$$

Минимальное число переменных $n = \begin{cases} \log_2]n_k[, \text{ если } (\overline{\exists} m)(n_k = 2^m) \\ \log_2 n_k + 1, \text{ если } (\exists m)(n_k = 2^m) \end{cases}$

## 2.3. Равенство булевых функций. Фиктивные переменные

$$f, g \in \mathcal{P}_2^{(n)}$$

**Определение 36** Равенство функций.

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

1. $x_1 \to x_2 = \overline{x_1} \vee x_2$
2. $x_1 \sim x_2 = (x_1 \to x_2) \cdot (x_2 \to x_1) = \overline{x_1 \oplus x_2}$
3. $x_1 | x_2 = \overline{x_1 \cdot x_2} = \overline{x_1} \vee \overline{x_2}$
4. $x_1 \downarrow x_2 = \overline{x_1 \vee x_2} = \overline{x_1} \cdot \overline{x_2}$
5. Формулы Моргана.

$$f = x_1 \vee x_2; \quad g = x_1 x_3 \vee x_1 \overline{x_3} \vee x_2 x_3 \vee x_2 \overline{x_3} = x_1(x_3 \vee \overline{x_3}) \vee x_2(x_3 \vee \overline{x_3}) = x_1 \vee x_2$$

**Определение 37** Фиктивная переменная. Переменная $x_i$ называется фиктивной переменной если $(\forall \tilde{\alpha}, \tilde{\beta})(\tilde{\alpha} = (\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n),\ \tilde{\beta} = (\beta_1, \ldots, \beta_{i-1}, 0, \beta_{i+1}, \ldots, \beta_n))(f(\tilde{\alpha}) = f(\tilde{\beta}))$

**Определение 38** Существенная переменная. Переменная булевой функции, не являющейся фиктивной, называется существенной.

**Определение 39** Равенство булевых функции. Две булевы функции называются равными, если они отличаются друг от друга быть может только своими фиктивными переменными.

**Определение 40** Равенство булевых функции. Две булевы функции называются равными, если они существенно зависят от одних и тех же переменных и при любом наборе этих переменных принимают одинаковое значение.

**Определение 41** $i$-селектор.

$$pr_i(x_1, \ldots, x_n) \rightleftharpoons x_i,\ 1 \leqslant i \leqslant n$$

Ввод фиктивных переменных для уравнивания количества переменных у двух функций:

$$y = f(x_1, \ldots, x_n) \quad y = (x_{n+1} \vee \overline{x_{n+1}}) f(x_1, \ldots, x_n) \rightleftharpoons \tilde{f}(x_1, \ldots, x_n, x_{n+1})$$

Используя возможность добавления фиктивных переменных к множеству переменных булевых функций, мы можем без ограничения общности считать, что две любые булевы функции, а следовательно любое число булевых функций, заданы как функции от одного и того же числа переменных.

## 2.4. Суперпозиции и формулы

$$f \in \mathcal{P}_2^n, \quad g_1, \ldots, g_n \in \mathcal{P}_2^m$$

**Определение 42.**

$$f(g_1, \ldots, g_n)(\tilde{\alpha}) \rightleftharpoons f(g_1(\tilde{\alpha}, \ldots, g_n(\tilde{\alpha}))),\ \tilde{\alpha} \in \{0,1\}^m$$

$$S(f; g_1, \ldots, g_n)$$

**Пример 23.**

**Рис. 18.** Суперпозиция

$$(x_1|x_2) \vee (x_1 \to x_2)$$

$$X = \{x_1, \ldots, x_n, \ldots\}$$

$\mathcal{F} = f_1, \ldots, f_m, \ldots$ — функциональные символы (уникальные имена булевых функций).
$\mathcal{F} = \mathcal{F}^{(0)} \cup F^{(1)} \cup \ldots \cup F^{(n)} \cup \ldots$ — множество функциональных символов арности $n$.

**Пример 24.** $\mathcal{F}_0 = \{\vee, \cdot, ^-\}$.

$$\mathcal{F}_0^{(0)} = \emptyset,$$

**Пример 25** Базис Жегалкина.

$$\mathcal{F}_1 = \{\oplus, \cdot, 1\} \quad \mathcal{F}_1^{(0)} = \{1\},\ \mathcal{F}_1^{(1)} = \emptyset,\ \mathcal{F}_1^{(2)} = \{\oplus, \cdot\}$$

**Определение 43** Формула над базисом $F$.

1. Всякая переменная из $X$ есть формула.
2. Если $\Phi_1, \ldots, \Phi_n$ — формулы, а $f^{(n)} \in \mathcal{F}^{(n)}$, то $f^{(n)}(\Phi_1, \ldots, \Phi_n)$ — формула.
3. Других формул нет.

**Пример 26.**

$$\overline{\cdot(\vee(\overline{x_1}, x_2), \vee(x_3, \overline{x_4}))} \mapsto \overline{(\overline{x_1} \vee x_2) \cdot (x_3 \vee \overline{x_4})}$$

Любая формула над базисом $\mathcal{F}$ представляет какую-то булеву функцию.

**Доказательство.**

1. Всякая переменная $x_i \in X$ представляет $i$-селектор.
2. Всякая константа из $\mathcal{F}^{(0)}$ представляет сама себя.
3. Если известно, что формула $\Phi_1$ представляет функцию $g_1$, …, формула $\Phi_n$ представляет функцию $g_n$, а $f^{(n)} \in \mathcal{F}^{(n)}$, то $f^{(n)}(\Phi_1, \ldots, \Phi_n)$ представляет $\textcircled{\cap} f^{(n)}(g_1, \ldots, g_n)$

□

**Определение 44** Полное множество. Множество булевых функций $F$ называется полным, если любая булева функция может быть представлена некоторой формулой над $F$.

**Определение 45** Замкнутое множество. Множество булевых функций $F$ называется замкнутым, если любая формула над $F$ представляет какую-то функцию из $F$.

## 2.5. Дизъюнктивные и конъюнктивные нормальные формы

**Определение 46** Литерал (буква). $x_i, \overline{x_i}$ — литерал (буква).

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

### ДНФ

**Определение 47** Элементарная конъюнкция. Элементарная конъюнкция — конъюнкция литералов $\tilde{x_{i_1}}, \tilde{x_{i_2}}, \ldots, \tilde{x_{i_k}}, i_1, \ldots, i_k \subseteq \{1, 2, \ldots, n\}$

**Определение 48** ДНФ. $K_1 \vee K_2 \vee \ldots \vee K_m,\ m \geqslant 1$

ДНФ называется совершенной, если каждая её элементарная конъюнкция содержит вхождение всех литералов.

**КНФ** Двойственным образом вводится конъюнктивная нормальная форма.

**Определение 49** Элементарная дизъюнкция. Элементарная дизъюнкция — дизъюнкция литералов $\tilde{x_{i_1}}, \tilde{x_{i_2}}, \ldots, \tilde{x_{i_k}}, i_1, \ldots, i_k \subseteq \{1, 2, \ldots, n\}$

**Определение 50** КНФ. $D_1 \cdot D_2 \cdot \ldots \cdot D_m,\ m \geqslant 1$

КНФ называется совершенной, если каждая её элементарная дизъюнкция содержит вхождение всех литералов.

**Теорема 21.** *Любая булева функция отличная от константы 0 (1) может быть представлена в виде ДНФ (КНФ).*

**Доказательство.** Пусть есть функция $y = f(x_1, \ldots, x_n) \not\equiv 0$.

$$C_f^{(1)} \rightleftharpoons \{\tilde{\alpha} : f(\tilde{\alpha}) = 1\} \neq \emptyset$$
$$K_{\tilde{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}, \text{ где } \tilde{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$$
$$K_{\tilde{\alpha}}(\tilde{\beta}) = 1 \Leftrightarrow \tilde{\beta} = \tilde{\alpha}$$
$$f(\tilde{\alpha}) = 1 \Leftrightarrow \tilde{\alpha} \in C_f^{(1)}$$

Таким образом:

Ⓕ □

см. учебник

**Лемма 2** о несамодвойственности функции. *Если $f_S \notin S$, то обе константы могут быть представлены формулой над $\{f_S, \bar{}\}$*

**Доказательство.** Так как $f_S \notin S$, то

$$(\exists \tilde{\alpha})(f(\tilde{\alpha}) = f(\overline{\tilde{\alpha}}))$$

Пусть $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$

$$h(x) \rightleftharpoons f_S(x^{\alpha_1}, \ldots, x^{\alpha_n});$$
$$h(0) = f_S(0^{\alpha_1}, \ldots, 0^{\alpha_n}) = f_S(\overline{\tilde{\alpha}}), \text{ так как } 0^1 = 0,\ 0^0 = 1$$
$$h(1) = f_S(1^{\alpha_1}, \ldots, 1^{\alpha_n}) = f_S(\tilde{\alpha}) \Rightarrow h(0) = h(1) = const \in \{0, 1\}$$

□

**Лемма 3** 1-я лемма о немонотонной функции. *Если $f_M \notin M$, то существуют наборы*

$$\tilde{\alpha} = (\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n)$$

*и*

$$\tilde{\beta} = (\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n)$$

*такие что*

$$f(\tilde{\alpha}) = 1,\ f(\tilde{\beta}) = 0$$

**Доказательство.** Так как $f_M \notin M$, то $\exists \tilde{\gamma}, \tilde{\delta};\ \tilde{\gamma} < \tilde{\delta}$, но $f(\tilde{\gamma}) = 1,\ f(\tilde{\delta}) = 0$.

$$\tilde{\gamma} = (\gamma_1, \ldots, \gamma_{i_1 - 1}, \underbrace{0}_{i_1}, \gamma_{i_1+1}, \ldots, \gamma_{i_k - 1}, \underbrace{0}_{i_k}, \gamma_{i_k+1}, \ldots, \gamma_n) \quad k \geqslant 1,\ k \leqslant n$$

$$\tilde{\gamma} = \tilde{\gamma}_0 < \tilde{\gamma}_1 < \tilde{\gamma}_2 < \ldots < \tilde{\gamma}_k = \tilde{\delta}$$

Набор $\tilde{\gamma}_l$ образуется из $\tilde{\gamma}_{l-1}$ путём замены компоненты: $(\tilde{\gamma}_{l-1})_{i_l} = 0 \Rightarrow (\tilde{\gamma}_l)_{i_l} = 1$

$$(\forall l = \overline{0, k-1})(\tilde{\gamma}_l \lhd \tilde{\gamma}_{l+1})$$

Где-то должен произойти скачок с 1 до 0.

$$(\exists l)(f(\tilde{\gamma}_l) = 1), f(\tilde{\gamma}_{i+1}) = 0$$

Тогда положим:

$$\tilde{\alpha} = \tilde{\gamma}_l, \tilde{\beta} = \tilde{\gamma}_{l+1}$$

□

**Лемма 4** 2-я лемма о немонотонной функции. *Если $f_M \notin M$, то отрицание может быть представлено формулой над $\{f_M, 0, 1\}$.*

**Доказательство.**

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

где

$$\tilde{\alpha} = (\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n)$$

Смотри лемму 3. □

**Лемма 5** о нелинейной функции. *Если $f_L \notin L$, то конъюнкцию можно представить формулой над $\{f_L, 0, {}^-\}$*

**Доказательство.** Если функция нелинейная, то в её полиноме Жегалкина найдётся нелинейное слагаемое. Выбираем самое короткое:

$x_{i_1}, x_{i_2}, \ldots, x_{i_k},\ \{i_1, i_2, \ldots, i_k\} \subseteq \{1, 2, \ldots, n\},\ k \geqslant 2$

$f'_L = f_L|_{(\forall j \notin \{i_1, i_2, \ldots, i_k\})(x_j = 0)} = x_{i_1}x_{i_2} \ldots x_{i_k} \oplus a_{i_1}x_{i_1} \oplus a_{i_2}x_{i_2} \oplus \ldots \oplus a_{i_k}x_{i_k} \oplus a_0$

$\chi(x, y) = f'_L\Big|_{\substack{x_{i_1}=\ldots=x_{i_s}=x \\ x_{s+1}=\ldots=x_{i_k}=y \\ 1 \leqslant s \leqslant k-1}} = xy \oplus ax \oplus by \oplus c$, где

$$a = \sum_{j=1}^{s} (mod\ 2) a_{ij}, \quad b = \sum_{j=s+1}^{k} (mod\ 2) a_{ij}, \quad c = a_0$$

$\chi(x \oplus b, y \oplus a) \oplus ab \oplus c = (x \oplus b)(y \oplus a) \oplus a(x \oplus b) \oplus b(y \oplus a) \oplus c \oplus ab \oplus c = xy \oplus ax \oplus by \oplus ab \oplus$
$\oplus\ ax \oplus ab \oplus by \oplus ab \oplus c \oplus ab \oplus c = xy$

$f'_L = f_L(0, \ldots, 0, x_{i_1}, 0, \ldots, 0, x_{i_2}, 0, \ldots, 0, x_{i_k}, 0, \ldots, 0)$

$$\chi(x,y) = f_L(0,\ldots,0,x,0,\ldots,0,x,0,\ldots,0,\underbrace{x}_{x_{is}},0,\ldots,0,\underbrace{y}_{i_{s+1}},0,\ldots,0,y,0)$$

$$x \cdot y = \tilde{f}_L(0,\ldots,0,\tilde{x},0,\ldots,0,\tilde{x},\ldots,0,\tilde{x},0,0,\ldots,0,\tilde{y},0,\ldots,0,\ldots,\tilde{y},0,\ldots,0)$$

□

**Пример 27.**

$$f_L = x_1x_2x_3x_4 \oplus x_1x_3x_4 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus 1$$
$$f'_L = f_L(x_1, x_2, 0, x_4) = x_1, x_2, x_4 \oplus x_1 \oplus x_2 \oplus x_4 \oplus 1$$
$$X(xy) = f_L(x, x, 0, y) = xy \oplus x \oplus x \oplus y \oplus 1 = xy \oplus y \oplus 1$$
$$a = 0,\ b = 1\ c = 1$$
$$xy = \overline{X(\overline{x}, y)} = \overline{f_l(\overline{x}, \overline{x}, 0, y)}$$

## 2.6. Теорема Поста

**Теорема 22** Поста. *Множество булевых функций полно тогда и только тогда, когда $F$ не содержится ни в одном из классов Поста.*

**Доказательство.**

**Необходимость** Пусть $(\exists C \in \{T_0, T_1, S, M, L\})(F \subseteq C)$. В силу замкнутости классов Поста любая формула из $F$ будет представляться формулой из $C$. Следовательно, ни одна формула, не принадлежащая ни одному из классов Поста, не может быть представлена, что противоречит условию полноты $F$.

**Достаточность** Для того чтобы доказать полноту $F$, удовлетворяющему условию теоремы, достаточно показать каким образом формулы над $F$ могут быть представлены формулами

множества, содержащего функцию, несохраняющую ноль, функцию, не сохраняющую единицу, не самодвойственную функцию, немонотонную функцию, нелинейную функцию.

Рассмотрим функцию, не сохраняющую ноль — $f_0$. $f_0(0) = 1$. $f_0(1)$ может принимать два значения:

1. $f_0(1) = 1$, тогда $f_0(x, x, x, \ldots, x) = 1$.
2. $f_0(1) = 0$, тогда $f_0(x, x, x, \ldots, x) = \overline{x}$.

Рассмотрим функцию, не сохраняющую один — $f_1$. $f_1(1) = 0$. $f_1(0)$ может принимать два значения:

1. $f_1(0) = 0$, тогда $f_1(x, x, x, \ldots, x) = 0$.
2. $f_1(0) = 1$, тогда $f_1(x, x, x, \ldots, x) = \overline{x}$.

Возможны четыре варианта:

1. Мы получили функцию $^-$. По лемме о несамодвойстенной функции (2) получим константы.
2. Мы получили $^-$ и 0. $1 = \overline{0}$.
3. Мы получили $^-$ и 1. $0 = \overline{1}$.
4. Мы получили 1 и 0. Получим отрицание по 2-ой лемме о немонотонной функции (4).

Итак, мы получили обе константы и отрицаний. Воспользовавшись леммой о нелинейной функции (5) получим конъюнкцию.

Таким образом, все функции могут быть представлены через функции полной системы $\{0, 1, ^-, \&\}$, а значит система функций $F$ является полной. □

**Пример 28.**

$F\{|\}, | \notin T_0 \cup T_1 \cup S \cup M \cup L \quad x|y = xy \oplus 1$

$$\overline{x} = x|x \quad x \vee y = \overline{\overline{x} \cdot \overline{y}} = (x|y)|(y|y) \quad xy = (x|y)|(x|y)$$

$$1 = \overline{x}|x = (x|x)|x \quad 0 = 1|1 = ((x|x)|x)((x|x)|x)$$

# 3. Элементы математической логики

## 3.1. Понятие формальной аксиоматической теории

<Пропущенная по болезни лекция>

**Теорема 23.** *Если* $\Gamma \vdash_{\mathcal{T}} \Phi$*, то для любого* $\Gamma' \supset \Gamma \quad \Gamma' \vdash_{\mathcal{T}} \Phi$

**Следствие 5.** *Если* $\vdash_{\mathcal{T}} \Phi$*, то для любого* $\Gamma : \Gamma \vdash_{\mathcal{T}} \Phi$

**Пример 29.**

$$\mathcal{T}_0 = (V_0, \mathcal{F}_0, \mathcal{A}_0, \mathcal{P}_0),$$

где

$$V_0 = Atom \cup Var \cup \{+, *, -\} \cup Aux$$
$$Atom = \{0, 1, \ldots, 9\},\ Var = \{a, b, \ldots, z\}$$

$\mathcal{F}_0$:

1. всякий атом и всякая переменная — формула;
2. если $\Phi$ и $\Psi$ — формула, то $(\Phi + \Psi), (\Phi * \Psi)$ — формулы;
3. если $\Phi$ — формула, то $-\Phi$ — формула;
4. ...

$\mathcal{A}_0 = Atom$

$$\mathcal{P}_0 : (1)\frac{X, Y}{(X + Y)}; (2)\frac{X, Y}{(X * Y)}; (3)\frac{X}{(-X)};$$

Правило вывода получается из схемы правила вывода путём согласованной замены каждой буквы формулой. Согласованной называется такая замена, при которой на место каждого вхождения одной и той же буквы подставляется одна и та формула.

Вывод:

$$\Phi = (-((a + 2) * (b + (-3))))$$
$$a, b, 2, 3, \underbrace{(a + 2)}_{(1)}, \underbrace{(-3)}_{(3)}, \underbrace{(b + (-3))}\, 2_{(1)}, \underbrace{((a + 2) * (b + (-3)))}_{(2)}, \underbrace{(-((a + 2) * (b + (-3))))}_{(3)}$$

Может быть определено:

$$\mathcal{T}_0' : \mathcal{A}_0' = Atom \cup Var$$

## 3.2. Алгебра высказываний

Под высказыванием мы понимаем любое повествовательное предложение, которое может быть истинным (И) или ложным (Л) (третьего не дано).

### Логические связки

1. Дизъюнкция («или», $\vee$)
2. Конъюнкция («и», &)

3. Импликация («если ... то ...», $\rightarrow$). Ложно, только если посылка истинна, а заключение — ложно.
4. Отрицание («не», $\neg$)
5. Эквивалентность («тогда и только тогда», $\equiv$)
6. Строгая дизъюнкция («исключающее или», $\oplus$)

1. $P, Q$ — высказывания $\Rightarrow (P \vee Q), (P \& Q), (P \rightarrow Q), (P \equiv Q), (P \oplus Q)$ — высказывания.
2. $P$ — высказывание $\Rightarrow \neg P$ — высказывание.

**Определение 51** Тавтология. Тождественно равное логическое высказывание называется тавтологией.

**Определение 52** Равносильность высказываний. Выражения логически равносильны, если соответствующие им логические функции равны.

Способы выявления тавтологии:

1. В лоб: через таблицу истинности.
2. Как вывод в некоторой формальной теории.
3. Метод резолюции (попытка найти набор, на котором высказывание будет ложным).

**Пример 30** Доказательство доказательства от противного.

$$\underbrace{(\neg q \to \neg p)}_{\text{И}} \to \underbrace{((\neg q \to p) \to q)}_{\text{Л}}$$

$$\underbrace{(\neg q \to p)}_{\text{И}} \to \underbrace{q}_{\text{Л}} = \text{Л} \Rightarrow q = \text{Л}$$

Пусть $F_1, \ldots, F_n, G$ — высказывания. Говорят, что $G$ является логическим следствием $F_1, \ldots, F_n$, если $F_1 \& \ldots \& F_n \to G$ — тавтология.

**Пример 31** Задача о забастовке. *Условие:* Если конгресс отказывается действовать, то забастовка не будет окончена, если только в течение года и президент не уходит в отставку. Спрашивается, закончится ли забастовка, если конгресс откажется действовать, а забастовка только началась.

*Решение:*

1. $p$ — «конгресс отказывается действовать»
2. $q$ — «забастовка оканчивается»
3. $r$ — «президент фирмы уходит в отставку»
4. $s$ — «забастовка длится более года»

$$\underbrace{(p \to (\neg q \vee (r \& s)))}_{F_1} \& \underbrace{p}_{F_2} \& \underbrace{\neg s}_{F_3} \to \neg q$$

$F_1 \& F_2 \& F_3 = $ И, но $\neg q = $ Л, то есть $q = $ И. $p = $ И, $\neg s = $ И, то есть $s = $ Л.

$\underbrace{\neg q}_{\text{Л}} \vee (r \& s) =$ И, но $r \& s =$ Л, так как $s =$ Л. Противоречие.

## 3.3. Исчисление высказываний

$$L = (V_L, \mathcal{F}_L, \mathcal{A}_L, \mathcal{P}_L)$$
$$V_L = Var \cup \{\text{Л, И}\} \cup \{\neg, \to\} \cup Aux$$

$\mathcal{F}_L$:

1. Каждая переменная есть формула.
2. Если $\Phi$ — формула, $\neg\Phi$ — формула.
3. Если $\Phi$ и $\Psi$ — формула, то $(\Phi \to \Psi)$ — формула.
4. Никаких других формул нет.

1. $A \to (B \to A)$
2. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
3. $(\neg B \to \neg A) \to ((\neg B \to A) \to B)$
4. $\mathcal{P}_L : \frac{A, A \to B}{B}$ — правило отсечения, Modus Ponens (MP).
5. $A \vee B = \neg A \to B$
6. $A \& B = \neg(A \to \neg B)$

**Теорема 24.** $\vdash_L (A \to A)$

**Доказательство.**

1. $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$
   схема (2) при $B := (A \to A)$, $C := A$
2. $A \to ((A \to A) \to A)$ — схема (1) при $B := A \to A$
3. $(A \to (A \to A)) \to (A \to A)$ — MP, (1) и (2)
4. $A \to (A \to A)$ — схема (1) при $B := A$
5. A $\to A$ — MP, (3) и (4)

Итак, $\vdash (A \to A)$ □

## 3.4. Теорема дедукции

**Теорема 25** Гербрамда. *Если из* $\Gamma \cup \{A\} \vdash B$, *то* $\Gamma \vdash (A \to B)$

**Доказательство.** Вместо $\Gamma \cup \{A\} \vdash B$ пишем $\Gamma, A \vdash B$.
Индукция по длине $l$ вывода $B$ из $\Gamma, A$.
*Базис:* $l = 0$, то есть

1. $B = A$. $B = A \Rightarrow A \to B = A \to A$, но $\vdash (A \to A) \Rightarrow \Gamma \vdash (A \to A)$

2. $B \in \Gamma$. $B \in \Gamma \Rightarrow B \to (A \to B)$ по схеме (1) при $A := B, B := A; B \in \Gamma; A \to B$ MP $\Gamma \vdash (A \to B)$
3. $B$ есть аксиома. $B$ — аксиома $\Rightarrow B$ — аксиома; $B \to (A \to B)$ по схеме (1); $A \to B$ MP $\Rightarrow \vdash (A \to B) \Rightarrow \Gamma \vdash (A \to B)$

*Предположение:* пусть $\forall l \leqslant n-1 \quad \Gamma, A \vdash^l B \Rightarrow \Gamma \vdash (A \to B)$.
*Переход:* $l = n (n \geqslant 1)$
Существует в выводе $B$ из $\Gamma, A$ формулы: $\Phi, \Phi \to B$, то есть $\Gamma, A \vdash^m, \Phi \to B$, где $m \leqslant n$ Ⓕ
□

Справедлива обратная теорема.

**Теорема 26.** *Если* $\Gamma \vdash (A \to B)$, *то* $\Gamma, A \vdash B$

**Доказательство.** Пусть $\Gamma \vdash (A \to B)$. Далее вводим новую гипотезу $A$. Тогда $\Gamma, A \vdash B$ (применяем MP к $A \to B$ и $A$). □

В дальнейшем любое утверждение о равносильности будем называть секвенцией. Таким образом, мы доказали: $\Gamma, A \vdash B \Leftrightarrow \Gamma \vdash (A \to B)$.

**Пример 32.** Докажем: $\vdash (\neg B \to \neg A) \to (A \to B)$

1. $\neg B \to \neg A$ гипотеза

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

---

$\neg B \to \neg A, A \vdash B$

---

$\neg B \to \neg A \vdash A \to B$

---

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

**Теорема 27.** *В теории $L$ имеют место следующие секвенции:*

1. $A \to B, B \to C \vdash A \to C$
2. $A \to (B \to C), B \vdash A \to C$
3. $\vdash (\neg\neg A \to A)$
4. $\vdash (A \to \neg\neg A)$
5. $\vdash (A \to (\neg A \to B))$

*6.* $\vdash (\neg B \to \neg A) \to (A \to B)$

*7.* $\vdash (A \to B) \to (\neg B \to \neg A)$

*8.* $\vdash A \to (\neg B \to \neg(A \to B))$

*9.* $\vdash (A \to B) \to ((\neg A \to B) \to B)$

**Доказательство.**
(1) $A \to B, B \to C \vdash A \to C$

| | | |
|---|---|---|
| 1. | $A \to B$ | гипотеза |
| 2. | $B \to C$ | гипотеза |
| 3. | $A$ | гипотеза |
| 4. | $B$ | MP, (1) и (3) |
| 5. | $C$ | MP, (2) и (4) |

---

$A \to B, B \to C, A \vdash C$

---

$A \to B, B \to C \vdash A \to C$

(2) $A \to (B \to C), B \vdash A \to C$

| | | |
|---|---|---|
| 1. | $A \to (B \to C)$ | гипотеза |

| | |
|---|---|
| 2. $B$ | гипотеза |
| 3. $A$ | гипотеза |
| 4. $B \to C$ | MP, (1) и (3) |
| 5. $C$ | MP, (2) и (4) |

$A \to (B \to C), B, A \vdash C$

$A \to (B \to C), B \vdash A \to C$

$(3) \vdash (\neg\neg A \to A)$

| | |
|---|---|
| 1. $\neg\neg A$ | гипотеза |
| 2. $(\neg A \to \neg\neg A) \to ((\neg A \to \neg A) \to A)$ | схема (3) при $A := \neg A, B := A$ |
| 3. $\neg\neg A \to (\neg A \to \neg\neg A)$ | схема (1) при $A := \neg\neg A, B := \neg A$ |
| 4. $\neg A \to \neg\neg A$ | MP, (1) и (3) |
| 5. $(\neg A \to \neg A) \to A$ | MP, (2) и (4) |
| 6. $\neg A \to \neg A$ | теорема |
| 7. $A$ | MP, (5) и (6) |

$\neg\neg A \vdash A$

---

$\vdash (\neg\neg A \to A)$

(4) $\vdash (A \to \neg\neg A)$

| | | |
|---|---|---|
| 1. | $A$ | гипотеза |
| 2. | $(\neg\neg\neg A \to \neg A) \to ((\neg\neg\neg A \to A) \to \neg\neg A)$ | схема (3) при $B := \neg\neg A$ |
| 3. | $\neg\neg\neg A \to \neg A$ | секвенция (3) |
| 4. | $(\neg\neg\neg A \to A) \to \neg\neg A$ | MP, (2) и (3) |
| 5. | $A \to (\neg\neg\neg A \to A)$ | схема (1) при $B := \neg\neg\neg A$ |
| 6. | $\neg\neg\neg A \to A$ | MP, (1) и (5) |
| 7. | $\neg\neg A$ | MP, (4) и (6) |

---

$A \vdash \neg\neg A$

---

$\vdash (A \to \neg\neg A)$

(6) $\vdash (\neg B \to \neg A) \to (A \to B)$ см. пример 32
(7) $\vdash (A \to B) \to (\neg B \to \neg A)$

| | | |
|---|---|---|
| 1. | $A \to B$ | гипотеза |
| 2. | $\neg\neg A \to A$ | секвенция (3) |
| 3. | $\neg\neg A \to B$ | секвенция (1), (2) и (1) |
| 4. | $B \to \neg\neg B$ | секвенция (4) |
| 5. | $\neg\neg A \to \neg\neg B$ | секвенция (1), (3) и (4) |
| 6. | $(\neg\neg A \to \neg\neg B) \to (\neg B \to \neg A)$ | секвенция (6) при $A := \neg B, B := \neg A$ |
| 7. | $\neg B \to \neg A$ | MP, (5) и (6) |

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

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

(8) $\vdash A \to (\neg B \to \neg(A \to B))$ Ⓕ
(9) $\vdash (A \to B) \to ((\neg A \to B) \to B)$

| | | |
|---|---|---|
| 1. | $(A \to B)$ | гипотеза |
| 2. | $(\neg$ | |

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

□

Ⓕ

**Следствие 6** Свойство дизъюнкции.

1. $A \vdash A \vee B; B \vdash A \vee B$
2. $A \wedge B \vdash B \vee A$
3. *Если* $\Phi \vdash \Psi$*, то для любой формулы* $A$ $A \vee \Phi \vdash A \vee \Phi$ *и* $\Phi \vee A \vdash \Psi \vee A$

**Доказательство.** (1)

1. $A$ — гипотеза
2. $A \to (\neg A \to B)$ — секвенция (5)
3. $\neg A \to B$ — MP, (1) и (2)

(2)Ⓕ

1. $\neg A \to B$
2. $(\neg A \to B) \to (\neg B \to \neg\neg A)$
3. $\neg B \to \neg\neg A$
4. $\neg\neg A \to A$
5. $\neg B \to A$

(3) □

**Следствие 7** Свойство конъюнкции.

1. $A\&B \vdash A, B$
2. $A, B \vdash A\&B$
3. $A\&B \vdash B\&A$

**Доказательство.** $A\&B = \neg(A \to \neg B)$

(1) Нужно доказать $A\&B \vdash A$, что равносильно: $\neg A \vdash \neg(A\&B) = \neg\neg(A \to \neg B)$

1. $\neg A$ — гипотеза
2. $\neg A \to (A \to \neg B)$ — секвенция (5)
3. $A \to \neg B$ — MP, (1) и (3)
4. $(A \to \neg B) \to \neg\neg(A \to \neg B)$ — секвенция (4)
5. $\neg\neg(A \to \neg B)$ — MP, (3) и (4)

(2)

1. $A$ — гипотеза
2. $B$ — гипотеза
3. $A \to (\neg\neg B \to \neg(A \to \neg B))$ — секвенция (8) при $B := \neg B$
4. $\neg\neg B \to \neg(A \to \neg B)$ — MP, (1) и (3)

5. $B \to \neg\neg B$ — секвенция (4)
6. $B \to \neg(A \to \neg B)$ — секвенция (1); (5) и (4)
7. $\neg(A \to \neg B) = A\&B$ — MP, (2) и (6)

□

## 3.5. Непротиворечивость, полнота и разрешимость теории L

**Теорема 28.** *Если* $\vdash_L \Phi$, *то* $\Phi$ — *тавтология.*

**Доказательство.** Достаточно показать, что применение правила MP по 2 тавтологиям даёт тавтологию.

Пусть $\Phi$ — тавтология, и $\Phi \to \Psi$ — тавтология.

$$\Phi = \Phi(x_1, \ldots, x_n), \Psi = \Psi(x_1, \ldots, x_n)$$

Допустим, что при этих условиях формула $\Psi$ не является тавтологией. Тогда

$$(\exists\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n))(\Psi(\tilde{\alpha} = Л$$

Следовательно: $(\Phi \to \Psi)(\tilde{\alpha}) = \Phi(\tilde{\alpha}) \to \Psi(\tilde{\alpha}) = И \to Л = Л$
Противоречие, так как $\Phi \to \Psi$ — тавтология. □

**Следствие 8.** *Теория L — непротиворечива, то есть в ней ни одна формула не может быть доказана вместе с её отрицанием.*

Ⓕ

$$\Phi^{\tilde{\alpha}} \rightleftharpoons \begin{cases} \Phi, & \text{если} \\ \dots \end{cases}$$

**Лемма 6.** *Если* $\Phi = \Phi(), \tilde{\alpha} = (\alpha_1, \dots, \alpha_)$, *mo* $x_1^{\alpha_1}, \dots, x_n^{\alpha_n} \vdash \Phi^{\tilde{\alpha}}$

**Доказательство.** Индукция по числу $l(\Phi)$ логических связок в формуле $\Phi$.
*Базис:* $l(\Phi) = 0$, то есть $\Phi = x_i$. Тогда $x_i^{\alpha_i} \vdash x_i^{\alpha_i}$, что очевидно (так как $\vdash_L (A \to A)$)
*Предположение:* пусть лемма доказана $\forall l(\Phi) \leqslant m - 1$ $(m \geqslant 1)$
*Переход:* положим $l(\Phi) = m$
*1 случай.* $\Phi = \neg\Psi$, где $l(\Psi) = m - 1$
Ⓕ □

**Теорема 29.** *Теория $L$ полна, то есть любая тавтология доказуема в $L$.*

**Доказательство.** Пусть формула $\Phi = \Phi(x_1, \ldots, x_n)$ — тавтология. Тогда для $(\forall \tilde{\alpha} = (\alpha_1, \ldots, \alpha_n))(\Phi(\tilde{\alpha}) = \text{И})$, и по лемме $x_1^{\alpha_1}, \ldots, x_n^{\alpha_n} \vdash \Phi$. От $\tilde{\alpha}$ переходим к $(\alpha_1, \ldots, \alpha_{n-1}, \neg\alpha_n)$, но тогда $x_1^{\alpha_1}, \ldots, x_{n-1}^{\alpha_{n-1}}, \neg x_n^{\alpha_n} \vdash \Phi$. Откуда, по (1) (сл. 1) из теоремы (27): $x_1^{\alpha_1}, \ldots, x_{n-1}^{\alpha_{n-1}} \vdash \Phi$. Действуя таким образом, получим $x_1^{\alpha_1} \vdash \Phi$ и $\neg x_1^{\alpha_1} \vdash \Phi$, откуда $\vdash \Phi$. □

**Теорема 30.** *Теория $L$ разрешима, то есть существует алгоритм, который для любой наперёд заданной формулы определяет, является ли она тавтологией или нет.*

## 3.6. Эквивалентные формулы

**Определение 53.** Формулы $\Phi, \Psi$ называются эквивалентными, если они выводимы друг из друга:

$$\Phi \equiv \Psi \rightleftharpoons (\Phi \vdash \Psi) \wedge (\Psi \vdash \Psi)$$

То есть:

$$\Phi \equiv \Psi \rightleftharpoons \vdash (\Phi \to \Psi)\&(\Psi \to \Phi)$$

Также: $\Phi \equiv \Psi \Leftrightarrow \neg\Phi \equiv \neg\Psi$

Известные эквивалентности: $\neg\neg A = A$; $(A \to B) \equiv (\neg B \to \neg A)$; $(A \to (B \to C)) \equiv (A\&B \to C)$.

**Формулы де-Моргана** $\neg(A \vee B) \equiv \neg A \& \neg B;\ \neg(A \& B) \equiv \neg A \vee \neg B$

**Доказательство.**

$$\neg(A \vee B) = \neg(\neg A \to B) \equiv \neg(\neg A \to \neg\neg B) \Leftrightarrow \neg A \to B = \neg A \to \neg\neg B$$

1. $\neg A \to B \equiv \neg A \to \neg\neg B$
   (а) $\neg A \to B$ — гипотеза
   (б) $B \to \neg\neg B$ — секвенция (4)
   (в) $\neg A \to \neg\neg B$ — секвенция (1); (1а) и (1б)
2. $\neg A \to \neg\neg B \vdash \neg A \to B$
   (а) $\neg A \to \neg\neg B$ — гипотеза
   (б) $\neg\neg B \to B$ — секвенция (3)
   (в) $\neg A \to B$ — секвенция (1); (2а) и (2б)

□

**Определение 54** подформула. Подформулой называется часть формулы, которая сама является формулой.

Пусть $\Phi[\Theta], \Theta' \equiv \Theta$. Тогда $\Phi[\Theta'/\Theta]$ — формула $\Phi$, в которой заменены все вхождения подформулы $\Theta'$ на $\Theta$.

**Теорема 31.** *Если* $\vdash_L \Phi[\Theta]$, *то* $\vdash_L \Phi[\Theta'/\Theta]$, *где* $\Theta' \equiv \Theta$.

**Следствие 9.** *Если* $\Gamma \vdash_l \Phi[\Theta]$, *и* $\Theta' \equiv \Theta$, *то* $\Gamma \vdash_L \Phi[\Theta'/\Theta]$

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

1. $(\neg B \to \neg A) \to ((\neg B \to A) \to B)$ — схема (3)
2. $(A \to B) \to ((\neg B \to A) \to B)$ — эквивалентность $(A \to B) \equiv (\neg B \to \neg A)$
3. $(A \to B) \to ((\neg A \to \neg\neg B) \to B)$ — эквивалентность $(\neg B \to A) \equiv (\neg A \to \neg\neg B)$
4. $((A \to B) \to ((\neg A \to B) \to B))$ — эквивалентность $\neg\neg B \equiv B$

## 3.7. Понятие алгебраической системы

**Пример 33.** $x_1 + x_2 \geqslant x_1 \cdot x_2 \quad (x_1, x_2 \in \mathbb{R})$ — истинно или ложно, в зависимости от $x_1, x_2$.

**Определение 55** предикат арности $n$. $p : A^n \to \{\text{И}, \text{Л}\}$, $n \geqslant 1$

**Определение 56** алгебраическая система. $\mathcal{A} = (A, \Omega, \Pi)$, где $\Pi$ — множество предикатов.

Пусть $p \in \Pi^{(n)}, \Pi = \Pi^{(1)} \cup \Pi^{(2)} \cup \ldots \Pi^{(n)} \cup \ldots$.
$\rho_p \rightleftharpoons \{(x_1, \ldots, x_n) : p(x_1, \ldots, x_n) = \text{И}\} \quad p(x_1, x_2) = \text{И} \rightleftharpoons x_1 \leqslant x_2 \ (x_1, x_2 \in \mathbb{R}$
$\rho_p \subseteq A^n$

Мы можем трактовать алгебраическую систему как некоторое множество предикатов как отношения. Если множество предикатов пусто, то получим алгебру. Если $\Omega$ пусто, то получим модель.

Пусть $\omega : A^n \to A \mapsto p_\omega \rightleftharpoons \{(x_1, \dots x_n, x_{n+1}) : x_{n+1} = \omega(x_1, \dots, x_n)\} \subseteq A^{n+1},\ n \geqslant 0$

Так можно определить: $+ : \mathbb{R}^2 \to \mathcal{R} \mapsto p_+ \rightleftharpoons \{(x_1, x_2, x_3) : x_3 = x_1 + x_2\}$

Примеры алгебраических систем:

1. $(\mathbb{R}, +, \cdot, 0, 1; \leqslant)$
2. $(Z, +, \cdot, 0, 1; \leqslant, |) \quad m|n \rightleftharpoons n = km, k \in Z$
3. $\mathcal{G} = (V, \rho), \rho \in V^2$

## 3.8. Исчисление предикатов первого порядка: алфавит, понятие формулы

### 3.8.1. Алфавит

1. $X$ — множество предметных переменных.
2. $C$ — множество предметных констант.
3. $\mathcal{F}$ — множество функциональных символов. $\mathcal{F} = \mathcal{F}^{(0)} \cup \mathcal{F}^{(1)} \cup \ldots \cup \mathcal{F}^{(0)} \cup \ldots$, причём $\mathcal{F}^{(0)} = C$
4. $\mathcal{P}$ — множество предикатных символов. $\mathcal{P} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)} \cup \ldots \cup \mathcal{P}^{(0)} \cup \ldots$
5. Логические символы: $\neg, \to, \forall$.

6. $Aux$ — множество вспомогательных символов.

### 3.8.2. Понятие формулы

**Определение 57** терм.

1. Всякая переменная из $X$ и константа из $C = \mathcal{F}^{(0)}$ есть терм.
2. Если $t_1, \ldots, t_n$ — терм, и $f^{(n)} \in \mathcal{F}^n$, то $f^{(n)}(t_1, \ldots, t_n)$ есть терм.
3. Других термов нет.

**Определение 58** атомарная формула. $p^{(n)}(t_1, \ldots, t_n)$, где $p^{(n)} \in \mathcal{P}$, $t_1, \ldots, t_n$ — термы.
Пример: $\geqslant (+(x_1, x_2), \cdot(x_1, x_2))$

1. Атомарная формула есть формула.
2. Если $\Phi$ и $\Psi$ — формулы, то $\Phi \to \Psi$ — формула.
3. Если $\Phi$ — формула, то $\neg\Phi$ — формула.
4. Если $\Phi$ — формула, $x_i \in X$, то $(\forall x_i)\Phi$ — формула.
5. Других формул нет.

## 3.9. Метод резолюций

### 3.9.1. Пронесение кванторов

Формула для пронесения квантора всеобщности через дизъюнкцию:

$$(\forall x X(x) \vee Y) \sim \forall x (X(x) \vee Y)$$

$$\neg \forall x X \sim \exists x \neg X$$

$$X \to Y \sim \neg X \vee Y$$

$$(\forall x X(x) \to Y \sim \exists x (X(x) \to Y)$$

$$Y \to \forall x X(x)) \sim \forall x (Y \to X(x)$$

Ⓕ

если конъюнкция ... является противоречием.

Суть метода резолюций: если мы имеем множество дизъюнктов, то определённым методом мы можем проверить, является ли данная формула противоречием.

### 3.9.2. Элиминация квантора существования

Мы предполагаем, что у нас есть любое множество констант.

$$\exists x X$$

ⓕ

### 3.9.3. Метод резолюций

Пусть есть два дизъюнкта $D_1, D_2$, в каждом из которых каждая переменная встречается только один раз.

Контрарная пара:

$$D_1 = x_1 \vee \tilde{D}_1$$
$$D_2 = \neg x_1 \vee \tilde{D}_2$$

Резольвента: $\tilde{D}_1 \vee \tilde{D}_2$.

**Теорема 32.** *Резольвента двух дизъюнктов является их логическим следствием, то есть:* $D_1 \wedge D_2 \to \tilde{D}_1 \vee \tilde{D}_2$ *— тавтология.*

**Доказательство.** Надо доказать, что $D_1 \wedge D_2 \wedge \ldots \wedge D_k \to R$. Для этого докажем более сильный результат: $D_i \wedge D_j \to R$.

$$D_i = x \vee \tilde{D}_i$$
$$D_j = \neg x \vee \tilde{D}_j$$

$$D_i \wedge D_j \to \tilde{D}_i \wedge \tilde{D}_j$$

Пусть $\tilde{D}_i \wedge \tilde{D}_j$ ложно. Значит и $\tilde{D}_i$ и $\tilde{D}_j$ ложно. □

Если $D_{k+1}$ есть логическое следствие $D_1, \ldots, D_k$, то $D_1, \ldots, D_k$ противоречиво $\Leftrightarrow$ $D_1, \ldots, D_k, D_{k+1}$ противоречиво.

Множество противоречиво тогда и только тогда, когда можно получить пустой дизъюнктор.

**Определение 59** результивный вывод. Последовательность дизъюнктов, в которой каждый дизъюнкт есть либо элемент исходного множества, либо получен из предыдущих элементов последовательности как резольвента, называется результивным выводом.

$D_1, \ldots, D_k; U_1, U_2, \ldots, U_k$

1. $U_i \in D_1, \ldots, D_k$
2. $U_i = \tilde{U}_{j_1} \vee \tilde{U}_{j_2}$, где $U_{j_1} = x \vee \tilde{U}_{j_1},\ U_{j_2} = x \vee \tilde{U}_{j2},\ j_1, j_2 < i$

Конечный элемент последовательности результивного вывода называется .

Множество дизъюнктов $D_1, \ldots, D_k$ противоречиво тогда и только тогда, когда существует вывод из этого множества пустого дизъюнкта.

Атомарная формула:

$$(\forall x_i)\Phi < x_i >$$

$$(\forall x_1)(x_1 \geqslant x_2) \vee (\forall x_2)(x_2 \geqslant x_1)$$

Первое вхождение $x_1$ — связаное, второе — свободное.

**Определение 60.** Пусть $t$ — терм, $\Phi < x_i >$ — формула, содержащая вхождение $x_i$. Тогда терм $t$ называют свободным для переменной $x_i$ в формуле $\Phi$, если ни одно свободное вхождение $x_i$ в $\Phi$ не лежит в области действия квантора по переменной терма.

$t = x_1 + x_2$. Терм не свободен для обоих переменных.
Терм, являющийся константой всегда свободен для любой переменной в любой формуле.

**Вспомогательные связки**

$$A \vee B = \neg A \to B,\ A\&B = \neg(A \to \neg B),\ (\exists x_i)\Phi = \neg(\forall x_i)(\neg\Phi)$$

## 3.10. Понятие интерпретации. Выполнимость, истинность, логическая общезначность

**Определение 61** интерпретация.

$$I = (\mathcal{A} = (A, \Omega, \Pi), i_F, i_P$$

$A$ — область интерпретации. $i_F : \mathcal{F} \to \Omega$, то есть $(\forall n \geqslant 0)(i_F(f^{(n)} \in \Omega^{(n)}.$ $i_P : \mathcal{P} \to \Pi, (\forall n \geqslant 1)(i_P(P^{(n)} \in \Pi^{(n)}.$

**Определение 62** состояние.

$$\sigma : X \to A$$

**Пример 34.** Пусть есть атомарная формула:

$$x_1 + x_2 \geqslant x_1 * x_2$$

Задаём интерпретацию: $A = \mathbb{Z}; +, *$ — арифметические операции. Эта формула выполнима в заданной интерпретации, но не истинна (не для любого $x_1, x_2$ она выполняется).

$$x_1 * x_2 = x_2 * x_1$$

Истинна в числовой интерпретации, но не в матричной.

В дальнейшем будет рассматриваться интерпретация $I$ и её состояния.

**Определение 63** значение терма. Значение $t^\sigma$ терма $t$ в состоянии $\sigma$:

1. Если $t = x_i \in X$, то $t^\sigma \rightleftharpoons \sigma(x_i)$
2. Если $t = c \in C = \mathcal{F}^{(0)}$, то $t^\sigma \rightleftharpoons i_F(c)$
3. Если $t = f^{(n)}(s_1, \ldots, s_n)$, где $f^{(n)} \in \mathcal{F}^{(n)},\ s_1, \ldots, s_n$ — термы, то $t^\sigma \rightleftharpoons i_F(f^{(n)})(s_1^\sigma, \ldots, s_n^\sigma)$

**Пример 35.** $t = (x_1 + x_2) * ((-x_3) + x_1)$
Когда положим: $A = \mathbb{Z},\ \sigma = (1/x_1, -2/x_2, 3/x_3, \ldots)$ сможем вычислить: 2.

**Определение 64** истинностное значение формулы в состоянии $\sigma$.

1. Если $\Phi = P^{(n)}(t_1, \ldots, t_n)$, где $P^{(n)} \in \mathcal{P}^{(n)},\ t_1, \ldots, t_n$ — термы, то $\Phi^\sigma \rightleftharpoons i_P(P^{(m)}(t_1^\sigma, \ldots, t_n^\sigma)$
2. Если $\Phi = \Psi \to \Theta$, то $\Phi^\sigma \to \Theta^\sigma$; если $\Phi = \neg\Psi$, то $\Phi^\sigma = \neg\Psi^\sigma$
3. $[(\forall x_i)\Phi]^\sigma$ = И $\rightleftharpoons$ Для любого состояния $\sigma' \overline{=}_i \sigma$ $\Phi^{\sigma'}$ = И, где $\sigma' \overline{=}_i \sigma \rightleftharpoons$ Для любой $x_j \neq x_i; \sigma'(x_j) = \sigma(x_j)$

$[(\exists x_i)\Phi]^\sigma$ = И $\Leftrightarrow$ $[\neg(\forall x_i)\neg\Phi]^\sigma$ = И $\Leftrightarrow$ $[(\forall x_i)\neg\Phi]^\sigma$ = Л $\Leftrightarrow$ Существует $\sigma' \overline{=}_i \sigma$ $\neg\Phi^{\sigma'}$ = Л $\Leftrightarrow$ Существует состояние $\sigma' \overline{=}_i \sigma$ $\Phi^{\sigma'}$ = И

**Степени истинности**

1. Выполнимость: [U+22A7]$_I\Phi \rightleftharpoons$ Для некоторого состояния $\Phi^\sigma$ = И
2. Истинность: $\vdash_I \Phi \rightleftharpoons$ Для любого состояния $\Phi^\sigma$ = И
3. Логическая общезначимость: формула истинна в любой интерпретации

## 3.11. Исчисление предикатов первого порядка: аксиомы и правила вывода

Аксиомы ИП1 (схемы)

1. $A \to (B \to A)$
2. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
3. $(\neg B \to \neg A) \to ((\neg B \to A) \to B)$
4. $(\forall x_i)A < x_i > \to A < t|x_i >$, при условии, что терм $t$ свободен для $x_i$ в формуле $A$
5. $(\forall x_i)(A \to B) \to (A \to (\forall x_i)B)$, при условии, что формула $A$ не содержит свободного вхождения $x_i$

**Правила вывода**

1. Modus-Ponus: $\frac{A, A \to B}{B}$
2. Jen: $\frac{A}{(\forall x_i)A}$

Контрпример к (5):

$$A = B = p^{(1)}(x_1),\ x_i = x_1$$

$$\underbrace{(\forall x_1)(p^{(1)}(x_1) \to p^{(1)}(x_1))}_{\text{И}} \to \underbrace{(p^{(1)}(x_1) \to (\forall x_1)p^{(1)}(x_1))}_{\text{Л}}$$

**Пример 36.**

$$A, (\forall x_1)A \to B \vdash (\forall x_1)B$$

1. $A$ — гипотеза
2. $(\forall x_1)A \to B$ — гипотеза
3. $(\forall x_1)A$ — Jen (1)
4. $B$ — MP, (2) & (3)
5. $(\forall x_1)B$ — Jen (4)

**Теорема 33.**

1. *Исчисление предикатов первого порядка непротиворечиво, т.е. в этой теории нельзя доказать некоторую формулу и её отрицание.*
2. *Всякая теорема исчисления предикатов первого порядка является логически общезначимой формулой.*
3. *Всякая логически общезначимая формула доказуема в исчислении предикатов первого порядка.*

**Следствие 10** теорема Гёделя о полноте. *Формула логически общезначима тогда и только тогда, когда она является теоремой исчисления предикатов первого порядка.*

**Теорема 34** свойство неразрешимости. *Исчисление предикатов первого порядка является неразрешимой теорией, то есть множество её теорий алгоритмически неразрешимо.*

**Определение 65** зависимость формул в выводе. Пусть есть вывод: $\Phi_1, \Phi_2, \ldots, \Phi_n, \ldots,$ — вывод в ИП1. Формула $\Phi_i$ этого вывода зависит от формулы $A$, если $\Phi_i = A$ или $\Phi_i$ получена применение правила (MP) или (Jen), к формуле, зависящей в выводе от $A$.

**Теорема 35** о дедукции для ИП1. *Если* $\Gamma, A \vdash B$, *то если ни при каком применении правила обобщения правила (Jen) к формуле, зависящей в этом выводе от* $A$, *не связывает свободные вхождения переменной в* $A$, *то* $\Gamma \vdash (A \to B)$

**Следствие 11.** *Если в выводе* $\Gamma, A \vdash B$ *не применяется правило (Jen), то* $\Gamma \to (\mathrm{A} \to B)$.

**Следствие 12.** *Если в выводе* $\Gamma, A \vdash B$ *применение правила (Jen) не связывает свободного вхождения переменного ни в одну из гипотез, то* $\Gamma \to (\mathrm{A} \to B)$.

**Доказательство.**

**Утверждение 1** Если $\Phi$ — тавтология, то $\vdash_{\text{ИП1}} \Phi$ и $\Phi$ может быть доказана при использовании только (MP) и схем (1) – (3)

**Утверждение 2** Если $\Gamma, A \vdash B$, причём B не зависит в выводе от A, то $\Gamma \vdash B$.

Если формула $B$ получена без использования (Jen), то мы оказываемся в условиях теоремы о дедукции в исчислении высказываний.

Базис индукции дословно повторяет соответствующее место из доказательства теоремы о дедукции в исчислении высказываний.

Сделаем индукционный переход при условии, что $B$ получена из некоторой формулы $\Phi$ применением (Jen), т.е. $B = (\forall x_i)\Phi$. $\Phi$ справедливо $\square$

$$(x = y) \to ((y = z) \to (x = z)) \equiv (x = y)\&(y = z) \to (x = z)$$

1. $x = y$ — гипотеза
2. $(x = y) \to (y = x)$ — п. (2)
3. $y = x$ — MP, (1) & (2)
4. $(y = x) \to ((y = z) \to (x = z))$ — (НЛ2) при $x := y, \quad y := x, \quad A < y, y >:= (y = z), \quad A < y, x >:= (x = z)$
5. $(y = z) \to (x = z)$ — MP, (3) & (4)

**Теорема 36.**

1. *Для любых термов* $s$ *и* $t$*:* $\vdash (s = t) \to (t = s)$
2. *Для любых термов* $s, t, u$*:* $\vdash (s = t) \to ((t = u) \to (s = u))$

**Доказательство.** $x, y \notin Var(s) \cup Var(t)$

1. $(x = y) \to (y = x)$ — теорема
2. $(\forall y)((x = y) \to (y = x))$ — Jen, (1)
3. $(\forall x)(\forall y)((x = y) \to (y = x))$ — Jen, (2)
4. $(\forall x)(\forall y)((x = y) \to (y = x)) \to (\forall y)((s = y) \to (y = s))$ — схема (4)
5. $(\forall y)((s = y) \to (y = s))$ — MP, (3) & (4)

6. $(\forall y)((s = y) \to (y = s)) \to ((s = t) \to (t = s))$ — схема (4)
7. $(s = t) \to (t = s)$ — MP, (5) & (6)

□

**Пример 37.** Доказательство теоремы о единственности нейтрального элемента в группе. Формулировка:

$$(\exists \varepsilon_1)(\forall x)(\varepsilon_1 \times x = x \times \varepsilon_1 = x) \to ((\exists \varepsilon_2)(\exists x)(\varepsilon_2 \times x = x \times \varepsilon = x) \to (\varepsilon_1 = \varepsilon_2), \text{ где}$$

$s = t = u \rightleftharpoons (s = t)\&(t = u) \equiv (s = u)\&(t = u)$

**Доказательство.** $(\forall x)(\varepsilon_1 \times x = x \times \varepsilon_1 = x), (\forall x)(\varepsilon_2 \times x = x \times \varepsilon_2 = x) \vdash (\varepsilon_1 = \varepsilon_2)$

1. $(\forall x)(\varepsilon_1 \times x = x \times \varepsilon_1 = x)$ — гипотеза
2. $(\forall x)(\varepsilon_2 \times x = x \times \varepsilon_2 = x)$ — гипотеза
3. $(\forall x)(\varepsilon_1 \times x = x \times \varepsilon_1 = x) \to (\varepsilon_1 \times \varepsilon_2 = \varepsilon_2 \times \varepsilon_1 = \varepsilon_2)$ — схема (4)
4. $(\varepsilon_1 \times \varepsilon_2 = \varepsilon_2 \times \varepsilon_1 = \varepsilon_2)$ — MP, (1) & (3)
5. $(\forall x)(\varepsilon_2 \times x = x \times \varepsilon_2 = x) \to (\varepsilon_2 \times \varepsilon_1 = \varepsilon_1 \times \varepsilon_2 = \varepsilon_1)$ — схема (4), $t = \varepsilon_1$
6. $(\varepsilon_2 \times \varepsilon_1 = \varepsilon_1 \times \varepsilon_2 = \varepsilon_1$ — MP, (2) & (5)
7. $\varepsilon_1 \times \varepsilon_2 = \varepsilon_2$ — свойство конъюнкции, (4)
8. $\varepsilon_1 \times \varepsilon_2 = \varepsilon_1$ — свойство конъюнкции, (6)

9. $(\varepsilon_1 \times \varepsilon_2 = \varepsilon_1) \to (\varepsilon_1 = \varepsilon_1 \times \varepsilon_2)$ — теорема
10. $\varepsilon_1 = \varepsilon_1 \times \varepsilon_2$ — MP, (8) & (9)
11. $(\varepsilon_1 = \varepsilon_1 \times \varepsilon_2) \to ((\varepsilon_1 \times \varepsilon_2) = \varepsilon_2) \to (\varepsilon_1 = \varepsilon_2)$ — теорема
12. $((\varepsilon_1 \times \varepsilon_2) = \varepsilon_2) \to (\varepsilon_1 = \varepsilon_2)$ — MP, (10) & (11)
13. $\varepsilon_1 = \varepsilon_2$ — MP, (7) & (12)

□

**Теорема 37.** *Все кошки одного цвета.*

**Доказательство.** *База*: Одна кошка одного цвета.

*Предположение индукции*: $n - 1$ кошка одного цвета.

*Шаг*: Вытащим кошку, получим $n-2$ кошек одного цвета. Добавим ещё одну кошку, получим $n - 1$ кошек одного цвета. Вернём ту кошку. Так как она такого же цвета, как и $n - 2$ кошек, все $n$ кошек будут одного цвета. □