

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. Инфлянская
Иллюстрации: А. С. Никичкин

26 апреля 2013 г.



Данный конспект распространяется по лицензии «Attribution-NonCommercial-ShareAlike» («С указанием авторства — Некоммерческая — С сохранением условий») 3.0. Для получения информации о данной лицензии посетите официальный сайт организации Creative Commons.

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

Данный конспект лекций предоставлен держателями авторских прав и/или другими сторонами «как есть» без какого-либо вида гарантий, выраженных явно или подразумеваемых, включая, но не ограничиваясь ими, подразумеваемые гарантии коммерческой ценности и пригодности для конкретной цели. Ни в коем случае, если не требуется соответствующим законом или не установлено в устной форме, ни один держатель авторских прав и ни одно другое лицо, которое может изменять и/или повторно распространять конспект, как было разрешено выше, не ответственны перед вами за убытки, включая любые общие, случайные, специальные или последовавшие убытки, проистекающие из использования или невозможности использования конспекта (включая, но не ограничиваясь получением вами не удовлетворяющей вас оценки по данному предмету), даже если такой держатель или другое лицо были извещены о возможности таких убытков.

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	5
1.1. Понятие алгоритма в интуитивном смысле слова	5
1.2. Машина Тьюринга	6
1.3. Нормальные алгорифмы Маркова	11
1.4. Эквивалентность нормальных алгорифмов. Теорема о переводе	13
1.4.1. Естественное и формальное распространение нормального алгорифма на более широкий алгорифм	14
1.5. Способы сочетаний нормальных алгорифмов	15
1.5.1. Композиция	15
1.5.2. Объединение	16
1.5.3. Разветвление	17
1.5.4. Повторение	17
1.6. Универсальный нормальный алгорифм	18
1.7. Разрешимые и перечислимые множества (языки)	19
1.7.1. Конструктивные числа	20
1.8. Проблема применимости для нормальных алгорифмов	21
1.9. Рекурсивные функции	23
1.10. λ -исчисление	25
1.10.1. β -редукция	26
1.10.2. Комбинаторы	27
2. Булевы функции	29
2.1. Булевы алгебры	29
2.2. Булевы функции. Основные понятия, таблица булевой функции	30
2.3. Равенство булевых функций. Фиктивные переменные	32
2.4. Суперпозиции и формулы	32
2.5. Дизъюнктивные и конъюнктивные нормальные формы	34
3. Элементы математической логики	35
3.1. Понятие формальной аксиоматической теории	35
3.2. Алгебра высказываний	36
3.3. Исчисление высказываний	37
3.4. Теорема дедукции	38
3.5. Непротиворечивость, полнота и разрешимость теории L	41
3.6. Эквивалентные формулы	42
3.7. Понятие алгебраической системы	43

1. Теория алгоритмов

Предтечи

Парадокс Рассела Пусть множество Y определяется следующим образом:

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

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:

$$Z = \{X : X \notin X\}$$

$$Z \in Z \Rightarrow Z \notin Z$$

$$Z \notin Z \Rightarrow Z \in Z$$

$$Z \notin Z \Leftrightarrow Z \in Z$$

$$Z \notin Z \Rightarrow Z \neq Z, \text{ то есть } Z \in Z \Rightarrow Z \notin Z \Rightarrow (Z \in Z) \& (Z \notin Z) - \text{противоречие!}$$

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.

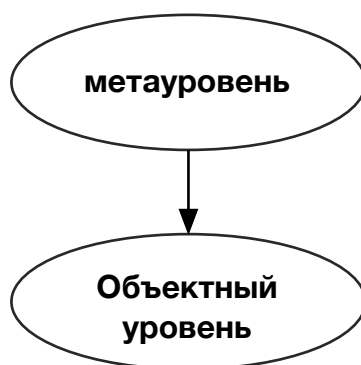


Рис. 1. Выход за пределы математики

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.

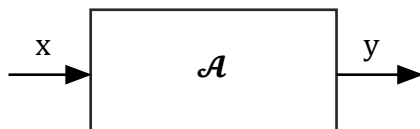


Рис. 2. Схема работы алгоритма

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

A — алгоритм типа XY : $A : X \rightarrow Y$

A — частичный алгоритм типа XY : $A : X \rightarrowtail Y$.

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: $V, W \quad f : V^* \rightarrowtail W^*$

Пример: $V = W \quad f(x) \Leftrightarrow xx = x^2 \quad f : V^* \rightarrow V^*$

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 $A : V^* \rightarrowtail W^*$. 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 $A_f : V^* \rightarrow W^*$, 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.

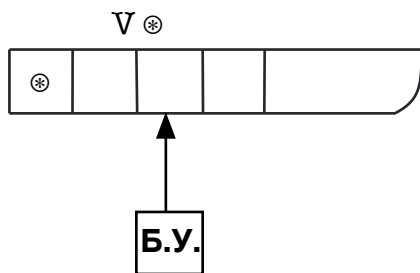


Рис. 3. Машина Тьюринга

Машина Тьюринга — полубесконечная лента, разделённая на буквы.

\odot — маркер начала ленты.

\square — символ пробела.

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

Запись команды

$$qa \rightarrow rb, \begin{cases} S \\ L \\ R \end{cases} \quad q, r \in Q, a, b \in V \cup \{\odot, \square\}$$

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).

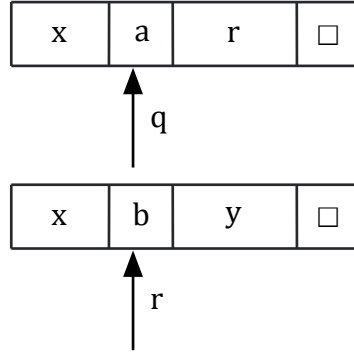


Рис. 4. Запись команды

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

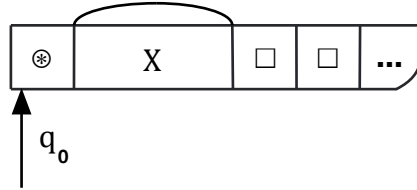


Рис. 5. Машина Тьюринга со входным словом

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

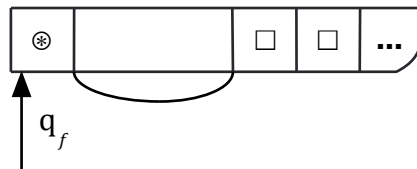


Рис. 6. Окончание работы машины Тьюринга

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?

$$\begin{array}{ll}
q_0(\odot) \rightarrow q_0(\odot), R & q_2b \rightarrow q_2b, L \\
q_0a \rightarrow q_0a, R & q_2(\odot) \rightarrow q_f(\odot), L \\
q_0b \rightarrow q_0b, R & q_1\Box \rightarrow q_3\Box, L \\
q_0c \rightarrow q_1c, R & q_3a \rightarrow q_3\Box, L \\
q_1a \rightarrow q_1a, R & q_3b \rightarrow q_3\Box, L \\
q_1b \rightarrow q_1b, R & q_3c \rightarrow q_3\Box, L \\
q_1c \rightarrow q_1c, R & q_3(\odot) \rightarrow q_3\Box, L \\
q_0\Box \rightarrow q_2\Box, L & q_3(\odot) \rightarrow q_f(\odot), S \\
q_2a \rightarrow q_2a, L &
\end{array}$$

Ответ: стирает слова, которые содержат букву c .

Формально машина Тьюринга определяется как следующий кортеж:

$$T = (V, Q, q_0, q_f, \odot, \Box, S, L, R, \delta),$$

где δ — система команд:

$$\delta : Q \times V' \rightarrow 2^{Q \times V' \times \{S, L, R\}}, \text{ где } V' = V \cup \{\odot, \Box\}$$

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

В детерминированной машине Тьюринга $\delta : Q \times V' \rightarrow Q \times V' \times \{S, L, R\}$ 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:

$$C_0 = (q_0, \lambda, \odot x \Box)$$

и конечная конфигурация:

$$C_f = (q_f, \lambda, \odot y \Box)$$

Определение 3 Отношение непосредственной выводимости.

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

Определение 4 Выводимость на множестве конечных конфигураций машины Тьюринга.

$$C_1, C_2, \dots, C_n, \dots \quad n \geq 1$$

$$(\forall i \geq 1)(C_i \vdash_{\mathcal{T}} C_{i+1}), \text{ если } C_{i+1} \text{ определена в последовательности}$$

$$C_1 \vdash_{\mathcal{T}} C_2 \vdash_{\mathcal{T}} \dots \vdash_{\mathcal{T}} C_n \text{ — длина} = n - 1 (n \geq 1)$$

$$C \vdash_{\mathcal{T}}^* C' \Leftrightarrow \text{существует вывод } C_1 \vdash_{\mathcal{T}} C_2 \vdash_{\mathcal{T}} \dots \vdash_{\mathcal{T}} C_n = C', n \geq 1)$$

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

$$!\mathcal{T}(x) \Rightarrow (q_0, \lambda, (*x\Box) \vdash_{\mathcal{T}}^* (q_f, \lambda, (*y\Box) \ y \Rightarrow \mathcal{T}(x) \oplus \neg !\mathcal{T}(x) \Rightarrow [(q_0, \lambda, \dots$$

Определение 5. Вербальная функция

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

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, вычисляемая в интуитивном смысле слова вычислима по Тьюрингу.*

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

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

$$\begin{aligned} q_0(*) &\rightarrow q_0(*), R \\ q_0a &\rightarrow q_1a_1, R \\ q_0b &\rightarrow q_0b_1, R \\ q_1a &\rightarrow q_2a_1, R \\ q_1b &\rightarrow q_0b, R \\ q_2a &\rightarrow q_2a, R \\ q_2b &\rightarrow q_3b, R \\ q_3\alpha &\rightarrow q_3\alpha, R, \alpha \in \{a, b\} \\ q_3\Box &\rightarrow q_4\Box, L \\ q_4\alpha &\rightarrow q_4\Box, L, \alpha \in \{a, b\} \\ q_4(*) &\rightarrow q_f(*), S \\ q_0\Box &\rightarrow q_5\Box, L \quad q_5 \text{ — движение в случае неуспешного поиска} \\ q_1\Box &\rightarrow q_5\Box, L \\ q_2\Box &\rightarrow q_5\Box, L \\ q_5\alpha &\rightarrow q_5\alpha, L, \alpha \in \{a, b\} \\ q_5(*) &\rightarrow q_5(*), S \end{aligned}$$

Прогонка:

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

Пример 3.

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

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

Прогонка:

$$\begin{aligned}
V_0 = \{a, b\} \quad &(q_0, \lambda, \circledast ab\Box) \vdash (q_0, \circledast, ab\Box) \vdash (q_a, \circledast\#, b\Box) \vdash (q_b, \circledast\#a, \Box) \vdash (q_1, \circledast\#, ab\Box) \vdash \\
&\vdash (q_1, \circledast, \#ab\Box) \vdash (q_1, \lambda, \circledast\#ab\Box) \vdash (q_f, \lambda, \circledast\#ab\Box)
\end{aligned}$$

Пример 4.

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

$$\begin{aligned}
q_0(\circledast) &\rightarrow q_0(\circledast), R \\
q_0\# &\rightarrow q_\#\#, R \\
q_\#\alpha &\rightarrow q_\alpha\#, L \\
q_\alpha\# &\rightarrow q_0\alpha, R \\
q_\#\Box &\rightarrow q_1\Box, L \\
q_1\alpha &\rightarrow q_1\alpha, L \quad \alpha \in V_0 \\
q_1(\circledast) &\rightarrow q_f(\circledast), S \\
q_1\# &\rightarrow q_1\Box, L
\end{aligned}$$

Прогонка:

$$\begin{aligned}
&(q_0, \lambda, \circledast\#abc\Box) \vdash (q_0, \circledast, \#abc\Box) \vdash (q_\#, \circledast, \#, abc\Box) \vdash (q_a, \circledast, \#\#bc\Box) \vdash (q_o, \circledast a, \#bc\Box) \vdash \\
&\vdash (q_\#, \circledast a\#, bc\Box) \vdash (q_b, \circledast a, \#\#c\Box) \vdash (q_0, \circledast ab, \#c\Box) \vdash (q_\#, \circledast ab\#, c\Box) \vdash (q_c, \circledast ab, \#\#\Box) \vdash \\
&\vdash (q_0, \circledast abc, \#\Box) \vdash (q_\#, \circledast abc\#, \Box) \vdash (q_1, \circledast abc, \#\Box) \vdash (q_1, \circledast ab, c\Box\Box) \vdash^3 (q_1, \lambda, \circledast abc\Box) \vdash \\
&\vdash (q_f, \lambda, \circledast abc\Box)
\end{aligned}$$

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

Вместо:

$$\begin{aligned}
q_1\alpha &\rightarrow q_1\alpha, L \quad \alpha \in V_0 \\
q_1(\circledast) &\rightarrow q_f(\circledast), S \\
q_1\# &\rightarrow q_1\Box, L
\end{aligned}$$

из предыдущего примера вставим:

$$\begin{aligned}
q_1\#\Box &\rightarrow q_2\Box, L \\
q_2\alpha &\rightarrow \alpha, L \quad (\alpha \in V_0) \\
q_2\# &\rightarrow q_\#\#, R \\
q_\#\# &\rightarrow q_\#\#, R \\
q_2(\circledast) &\rightarrow q_f(\circledast), S
\end{aligned}$$

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.

$$\begin{aligned}
 &V, x, y \in V^* \quad x \sqsubseteq y \Leftrightarrow (\exists y_1, y_2)(y = \underbrace{y_1}_{\text{левое крыло}} \underbrace{x}_{\text{основа}} \underbrace{y_2}_{\text{правое крыло}}) \\
 &(\forall x)(\lambda \sqsubseteq x) \& (x \sqsubseteq x) \\
 &x \sqsubseteq y, y \sqsubseteq z \Rightarrow x \sqsubseteq z \\
 &(y_1, x, y_2), \text{ где } y = y_1 x y_2 \quad y_1 \star x \star y_2, \star \notin V
 \end{aligned}$$

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.

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

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

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

$$y \equiv x_1 v x_2 \equiv \omega x \text{ —}$$

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

$$x = \begin{array}{|c|c|c|} \hline x_1 & u & x_2 \\ \hline \end{array}$$

$$y = \begin{array}{|c|c|c|} \hline x_1 & v & x_2 \\ \hline \end{array}$$

Рис. 7. Формула подстановки

Например, $x =$ входит, $\omega : \text{вход} \rightarrow \text{уход}$. $\omega x =$ 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):

$$\left\{ \begin{array}{l} u_1 \rightarrow [\cdot] v_1 \\ u_2 \rightarrow [\cdot] v_2 \\ \vdots \\ u_n \rightarrow [\cdot] v_n \end{array} \right.$$

Пример 6 Добавление aba в конец слова.

$$\mathcal{A}_0 : \left\{ \begin{array}{l} \#a \rightarrow a\# \\ \#b \rightarrow b\# \\ \# \rightarrow \cdot aba \\ \rightarrow \# \end{array} \right. \quad V = \{a, b, \#\}$$

Прогонка:

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

В общем случае:

$$\begin{aligned} \mathcal{A}_0 : x = x(1)x(2) \dots x(k) \vdash \#x(1)x(2) \dots x(k) \vdash x(1)\#x(2) \dots x(k) \vdash \\ \vdash x(1)x(2)\# \dots x(k) \vdash \dots \vdash x(1)x(2) \dots x(k)\# \vdash \cdot x(1)x(2) \dots x(k)aba \quad (k \geq 1) \end{aligned}$$

Пример 7 Добавление aba в начало слова.

$$\begin{aligned} \mathcal{A}_1 : \left\{ \begin{array}{l} \rightarrow \cdot aba \end{array} \right. \\ (\forall x \in \{a, b\}^*)(\mathcal{A}_1 : x \vdash \cdot abax) \end{aligned}$$

Пусть есть нормальный алгорифм:

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

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

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

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

$$\begin{aligned} x = x_0, x_1, x_2, \dots, x_n = y, \text{ где } (\forall i = 0, n-1)(\mathcal{A} : x_i \vdash x_{i+1}) \\ \mathcal{A} : x \models y \Leftrightarrow (\mathcal{A} : \vdash \cdot y) \Leftrightarrow (\mathcal{A} : x \vdash \cdot y) \vee (\exists z)(\mathcal{A} : x \models z \vdash \cdot y) \end{aligned}$$

$!\mathcal{A}(x)$ — 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:

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

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. $\mathcal{A} : x_{n-2} \vdash \cdot x_{n-1}, n \geq 2$

Пусть $x_n = x_0, x_1, \dots, x_n$ — 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

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

называется вычислимой по Маркову, если может быть построен нормальный алгоритм \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, вычисляемая в интуитивном смысле слова, вычислима по Маркову.*

Пример 8 Правое присоединение.

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

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

Пример 9 Удвоение слова.

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

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

Прогонка:

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

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

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

Таким образом действие вышеописанной модели Маркова:

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

1.4. Эквивалентность нормальных алгоритмов. Теорема о переводе

Определение 12 Условное равенство.

$$\mathcal{A}, \mathcal{B} : V^* \rightarrow V^* \quad (\forall x \in V^*)(! \mathcal{A}(x) \Leftrightarrow ! \mathcal{B} \& (\mathcal{A}(x) = \mathcal{B}(x)) \Rightarrow \mathcal{A}(x) \cong \mathcal{B}(x))$$

Замыкание схемы нормального алгоритма Исходная схема:

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

Новая схема:

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

называется замыканием нормального алгорифма \mathcal{A} .

Утверждение 1.

$$(\forall x \in V^*)(A(x) \cong A'(x))$$

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

1. $A : x \models y, A : \neg y$ (есть обрыв) или
2. $\mathcal{A} : x \models y$
1. $\mathcal{A} : x \models y \vdash y$, то есть $\mathcal{A} : x \models y$;
2. $\mathcal{A} : x \models y$. Если $!A(x)$, то $!\mathcal{A}(x)$, причём $\mathcal{A} : x \models \mathcal{A}(x) = \mathcal{A}'(x)$. Если же $\neg !A(x)$, то $\neg !\mathcal{A}(x)$, то есть $!\mathcal{A}'(x) \Rightarrow !A(x)$.

Итак, $A(x) \cong \mathcal{A}'(x)$ □

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

$$\begin{aligned} A &= (V, S, P), V' \supset V \\ A' &= (V', S, P) \text{ — естественное распространение} \\ (\forall x \in V^*)(A(x) &\cong A'(x)) \\ A^f &= (V', S^f, P) \\ S^f &= \begin{cases} \xi \rightarrow \xi & (\xi \in V' \setminus V) \\ S \end{cases} \\ (\forall x \in V^*)(A^f(x) &\cong A(x)), \text{ но } (\forall x \notin V^*)(\neg !A^f(x)) \end{aligned}$$

Пусть есть алфавиты V, V_0 :

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

Можно закодировать буквы и слова алфавита V буквами алфавита V_0 .

$$\begin{aligned} [a_i \Rightarrow 0 \underbrace{11 \dots 1}_i 0] \quad x \in V^*[\lambda = \lambda, [x(1)x(2) \dots x(k) \Rightarrow [x(1) \dots [x(k) \\ V = \{a, b, c\}[abca = 010011001110010 \end{aligned}$$

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

1.5. Способы сочетаний нормальных алгоритмов

1.5.1. Композиция

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

Доказательство. Определим $\bar{V} = \{\bar{a}_1, \dots, \bar{a}_n\}$, где $V = \{a_1, \dots, a_n\}$, и $\bar{V} \cap V = \emptyset$ $\alpha, \beta \notin V \cup \bar{V}$

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

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

\mathbb{A}^\cdot	\mathbb{A}^α	\mathbb{B}^\cdot	$\bar{\mathbb{B}}_\alpha^\beta$
$u \rightarrow v$	$u \rightarrow v$	$\rightarrow v$	$\alpha \rightarrow \alpha\bar{v}$
$u \rightarrow \cdot v$	$u \rightarrow \alpha v$	$u \rightarrow v$ ($u \neq \lambda$)	$\bar{u} \rightarrow \bar{v}$
		$u \rightarrow \cdot v$ ($u \neq \lambda$)	$\bar{u} \rightarrow \beta\bar{v}$
		$\rightarrow \cdot v$	$\alpha \rightarrow \alpha\beta\bar{v}$

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

$$C : x \models y_1\alpha y_2, \text{ где } y_1y_2 = A^\cdot(x) \quad (I)$$

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

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

$$\alpha \bar{y} \models_{(8)} \alpha \bar{z}_1 \beta \bar{z}_2, \text{ где } z_1 z_2 \rightleftharpoons z = B^\cdot(y) = B^\cdot(A^\cdot(x)) \quad (IV)$$

□

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

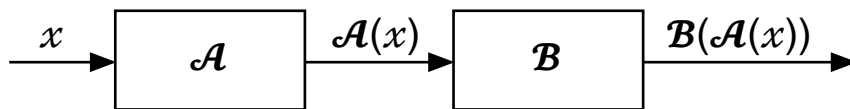


Рис. 8. Композиция

Композиция может обозначаться: $C = B \circ A$.

Степень:

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

Пример 10.

$$\begin{aligned} A : & \begin{cases} \#a \rightarrow a\# \\ \#b \rightarrow b\# \\ \# \rightarrow \cdot aba \\ \rightarrow \# \\ \rightarrow \cdot \end{cases} \\ B : & \begin{cases} \rightarrow \cdot baba \\ \rightarrow \cdot \end{cases} \\ C : & \begin{cases} \xi\alpha \rightarrow \alpha\xi \\ \alpha\xi \rightarrow \alpha\bar{\xi} \\ \bar{\xi}\eta \rightarrow \xi\bar{\eta} \\ \bar{\xi}\beta \rightarrow \beta\bar{\xi} \\ \beta\bar{\xi} \rightarrow \beta\xi \\ \xi\bar{\eta} \rightarrow \xi\eta \\ \alpha\beta \rightarrow \cdot \\ \alpha \rightarrow \alpha\beta\bar{b}\bar{a}\bar{b}\bar{a} \\ \alpha \rightarrow \alpha\beta \\ \#a \rightarrow a\# \\ \#b \rightarrow b\# \\ \# \rightarrow \alpha aba \\ \rightarrow \# \\ \rightarrow \alpha \end{cases} \end{aligned}$$

Работа алгорифма:

$$\begin{aligned} C : x = baa \vdash \#baa \models baa\# \vdash baa\alpha aba \models \alpha baa aba \vdash \alpha\bar{b}\bar{a} aba \models \alpha\bar{b}\bar{a}\bar{a}\bar{b}\bar{a} \vdash \alpha\beta\bar{b}\bar{a}\bar{b}\bar{a}\bar{a}\bar{a}\bar{b}\bar{a} \vdash \\ \vdash \alpha\beta\bar{b}\bar{a}\bar{b}\bar{a}\bar{a}\bar{a}\bar{b}\bar{a} \vdash \alpha\beta\bar{b}\bar{a}\bar{b}\bar{a}\bar{a}\bar{a}\bar{b}\bar{a} \models \alpha\beta bababaaaba \vdash \cdot bababaaaba \end{aligned}$$

1.5.2. Объединение

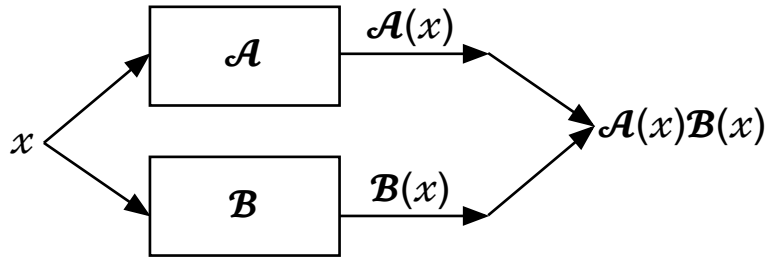


Рис. 9. Объединение

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*

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

$$\mathcal{C}(x\$y) \cong \mathcal{A}(x)\$ \mathcal{B}(y)$$

$$x, y \in V^*, \$ \notin V$$

1.5.3. Разветвление

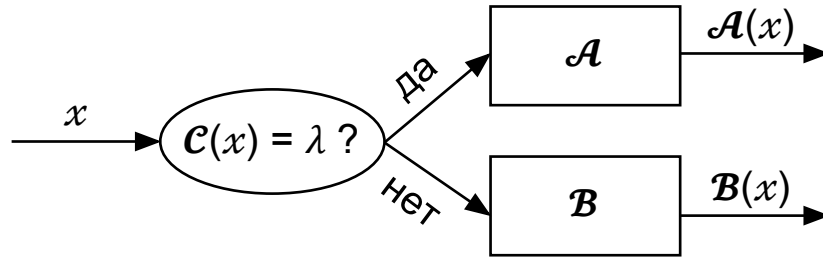


Рис. 10. Разветвление

```

1  if  $C(x) = \lambda$  then
2     $y \leftarrow A(x)$ 
3  else
4     $y \leftarrow B(x)$ 

```

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

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

1.5.4. Повторение

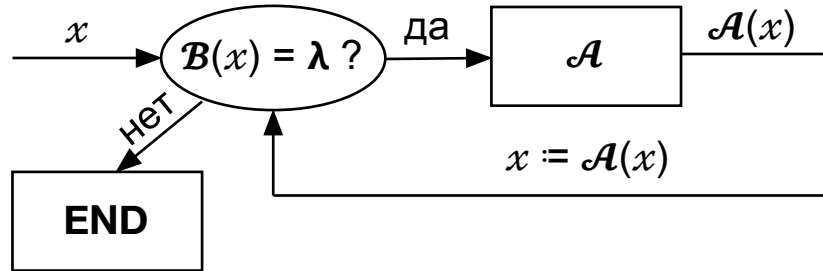


Рис. 11. Повторение алгоритма \mathcal{A} , управляемого алгоритмом \mathcal{B}

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

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

Обозначение: $\mathcal{C} =_B \{A\}$

```

1  while  $(B(x) = \lambda)$  do
2     $x \leftarrow A(x)$ 
3  end

```

Другой вид повторения:

Обозначение: $\mathcal{C} =_B < A >$

```

1  while ( $B(x) \neq \lambda$ ) do
2     $x \leftarrow A(x)$ 
3  end

```

Определение 13. Векторное слово в алфавите V :

$$x_1 \$ x_2 \$ \dots \$ x_n, \quad n \geq 1, \quad \$ \notin V \text{ n-ка слов}$$

Пример 11 Проектирующие алгоритмы. V — алфавит.

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

$$\mathcal{P}_1 = \begin{cases} \$ \eta \rightarrow \$ (\eta \in V) \\ \$ \rightarrow \\ \rightarrow \end{cases} \quad \mathcal{P}_1(x_1 \$ x_2 \$ \dots \$ x_n) = x_1$$

$$\mathcal{P}_2 = \begin{cases} \eta \# \rightarrow \# (\eta \in V, \# \notin V, \eta \neq \#) \\ \# \rightarrow \\ \$ \rightarrow \# \end{cases} \quad \mathcal{P}_2(x_1 \$ x_2 \$ \dots \$ x_n) = x_2 \$ \dots \$ x_n$$

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

Пример 12 Распознавание равенства слов.

$$EQ(x \$ y) = \lambda \Leftrightarrow x = y, \text{ где } x, y \in V^*, \$ \notin V$$

$$Inv(y) = y^R$$

$$EQ(x \$ y) \cong Comp(Id(x) \$ Inv(y))$$

$$Comp : \begin{cases} \eta \$ \eta \rightarrow \$ (\eta \in V) \\ \$ \rightarrow \end{cases}$$

1.6. Универсальный нормальный алгоритм

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

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

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

α — стрелки, β — подточки, γ — разделитель между формулами.

Пример 13.

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

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

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

Обозначение: $\llbracket \mathcal{A} \rrbracket$

$$\llbracket \mathcal{A}_0 \rrbracket = \langle a-1, b-2, \#-3, \alpha-4, \beta-5, \gamma-6 \rangle = \underbrace{01110}_{\#} \underbrace{010}_a \underbrace{011110}_{\alpha} \underbrace{010}_a \underbrace{01110}_{\#} \underbrace{01111110}_{\gamma} \dots$$

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

$$U(\llbracket \mathcal{A} \rrbracket \$x) \cong \mathcal{A}(x)$$

1.7. Разрешимые и перечислимые множества (языки)

Определение 15. Язык L в алфавите V^* называется алгоритмически разрешимым, если может быть построен нормальный алгоритм \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 алгоритм.

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

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

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

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

□

Пример 14 Язык двойных слов.

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

Докажем, что язык является разрешимым.

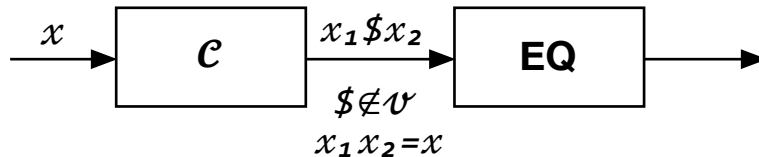


Рис. 12. Язык двойных слов

$$x_1 x_2 = x$$

$$|x_1| = |x_2|, \text{ если } |x| = 2k$$

$$||x_1| - |x_2|| = 1 \text{ иначе}$$

$$EQ \circ C(x_1 = \lambda \Leftrightarrow x = ww \text{ для некоторого } w \in V^+)$$

Пример 15.

$$\begin{aligned}
R &: \begin{cases} \gamma\xi \rightarrow \xi\gamma, \xi \in V, \gamma \notin V, \beta \notin V \\ \xi\gamma \rightarrow \cdot\beta\xi \\ \xi\beta \rightarrow \cdot\beta\xi \\ \rightarrow \gamma \end{cases} \\
L &: \begin{cases} \alpha\beta \rightarrow \cdot\alpha\beta (\alpha \notin V) \\ \alpha\xi \rightarrow \cdot\xi\alpha \\ \rightarrow \alpha \end{cases} \\
A &: \begin{cases} \xi\alpha\beta \rightarrow \alpha\beta \\ \alpha\beta\xi \rightarrow \alpha\beta \\ \alpha\beta \rightarrow \cdot \end{cases} \\
\mathcal{B} &: \{\alpha\beta \rightarrow \cdot \$ \\
\mathcal{C} &= \mathcal{B} \circ_A < L \circ R >
\end{aligned}$$

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 $(\forall n \in \text{конструктивное нормальное число})(\mathcal{N}_L(n) \& \mathcal{N}_L(x) \in L)$ и $\forall x \in L$ осуществимо конструктивное натуральное число n такое, что $\mathcal{N}_L(n) = x$.

Нумерация:

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

Рис. 13. Нумерация рациональных чисел

Пример 16.

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

$$\begin{aligned}
\mathcal{N}_L &= {}_c(\mathcal{A} \vee \mathcal{B}) \\
\mathcal{C} &: \begin{cases} 011 \rightarrow 0 \\ 0 \rightarrow \cdot \end{cases} \\
\mathcal{C}(n) &= \lambda \Leftrightarrow n = 2k \text{ (} k \text{ — конструктивное натуральное число)} \\
\mathcal{A} &\begin{cases} \alpha 11 \rightarrow 1\alpha \\ \alpha \rightarrow \cdot \\ 0 \rightarrow -0\alpha \end{cases} \\
\mathcal{A}(n) &= -\frac{n}{2} \\
\mathcal{B} &: \begin{cases} \alpha 11 \rightarrow 1\alpha \\ \alpha \rightarrow \cdot \\ 0 \rightarrow 0\alpha 1 \end{cases} \\
\mathcal{B}(n) &= \frac{n+1}{2}
\end{aligned}$$

Определение 21 Область применимости нормального алгорифма.

$$\mathcal{A}: V^* \rightarrow V^* \text{ — нормальный алгорифм над } V$$

Область применимости:

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

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 $x \in V^*$

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

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

$$!\mathcal{B}(\llbracket \mathcal{A} \rrbracket) \& \mathcal{B}(\llbracket \mathcal{A} \rrbracket) = \lambda \Leftrightarrow \neg !\mathcal{A}(\llbracket \mathcal{A} \rrbracket)?$$

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.

$$\begin{aligned} \mathcal{A}_0 : \begin{cases} \#a \rightarrow a\# \\ \#b \rightarrow b\# \\ \# \rightarrow \cdot aba \\ \rightarrow \# \\ \rightarrow \cdot \end{cases} \\ V = \{a, b, \#\} \\ \mathcal{A}_0 : \llbracket \mathcal{A}_0 \rrbracket \vdash \# \llbracket \mathcal{A}_0 \rrbracket \vdash \cdot aba \llbracket \mathcal{A}_0 \rrbracket \end{aligned}$$

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:

$$\mathcal{B}(\llbracket \mathcal{A} \rrbracket) \Leftrightarrow !\mathcal{A}(\llbracket \mathcal{A} \rrbracket)$$

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

$$!\mathcal{B}(\llbracket \mathcal{B} \rrbracket) \Leftrightarrow \neg !\mathcal{B}(\llbracket \mathcal{B} \rrbracket)$$

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

□

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

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

$$!\mathcal{B}(\llbracket \mathcal{A} \rrbracket) \Leftrightarrow \neg !\mathcal{A}(\llbracket \mathcal{A} \rrbracket)$$

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

По теореме о переводе может быть построен нормальный алгорифм \mathcal{B}_1 в алфавите $V_2 = V_0 \cup \{\alpha, \beta\}$ ($\alpha, \beta \notin V_0$) 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}_∞ в V_1 такой, что для любого нормального алгорифма \mathcal{A} в V_1 имеет место

$$!\mathcal{B}'_1(\llbracket \mathcal{A} \rrbracket) \Leftrightarrow \neg !\mathcal{A}(\llbracket \mathcal{A} \rrbracket)$$

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

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

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

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 алгоритме построим нормальный алгоритм \mathcal{U} так, что $(\forall y \in V_2^*)$ и любого нормального алгоритма \mathcal{C} в алгоритме V_2 имеет место $\mathcal{U}([C]\$y) \cong \mathcal{C}(y)(\$ \notin V_2)$. Строим нормальный алгоритм \mathcal{U}_1 так, что $(\forall y \in V_2^*)(\mathcal{U}_1(y) \cong \mathcal{U}(y\$y))$. Можно определить $\mathcal{U}_1 = \mathcal{U} \circ \text{Double}^{\$}(\text{Double}^{\$}(y) = y\$y)$ $\mathcal{U}, \mathcal{U}_1$ — алгоритмы 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 алгоритм над V_2 может быть заменён вполне эквивалентным ему относительно алфавита V_0 нормальным алгоритмом в алфавите V_2 . Может быть построен нормальный алгоритм \mathcal{A} в V_2 так, что $(\forall x \in V_0^*)(\mathcal{A}(x) \cong \mathcal{U}_1(x))$. Утверждается, что нормальный алгоритм \mathcal{A} и есть искомым алгоритм.

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

$$!\mathcal{B}(\llbracket C \rrbracket) \Leftrightarrow \neg !\mathcal{A}(\llbracket C \rrbracket) \Leftrightarrow \neg !\mathcal{U}_1(\llbracket C \rrbracket) \Leftrightarrow \neg !\mathcal{U}(\llbracket C \rrbracket \$ \llbracket C \rrbracket) \Leftrightarrow \neg !\mathcal{C}(\llbracket C \rrbracket)$$

Поскольку алгоритм \mathcal{B} может быть заменён вполне эквивалентным ему относительно V_0 нормальным алгоритмом, то алгоритм \mathcal{B} решает проблему самоприменимости в алфавите V_2 , что невозможно. \square

Следствие 2. Проблема применимости для нормальных алгоритмов алгоритмически неразрешима.

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

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

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

Проблема соответствий Поста ставится так: для любого заданного наперёд отношения ρ выяснить, является ли пустым язык L_ρ .

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

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 алгоритмов:

$$V, \llbracket \mathcal{A} \rrbracket, \quad \varphi_{\llbracket \mathcal{A} \rrbracket} — \text{функция, которая вычисляет нормальный алгоритм } \mathcal{A} \quad L = \{\llbracket \mathcal{A} \rrbracket : \varphi_{\llbracket \mathcal{A} \rrbracket} \in \mathcal{F}\}$$

Теорема 14 Райса. Язык L алгоритмически неразрешим.

1.9. Рекурсивные функции

Базовые функции:

1. $(\forall x \in \mathbb{N}) \mathbb{O}(x) = 0$
2. $(x \in \mathbb{N}) +\mathbb{K}(x) = x + 1$
3. $\prod_i(x_1, \dots, x_i, \dots, x_n) = x_i, i = \overline{1, n}$

Правила:

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

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

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

При $n = 0$:

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

Сложение

$$x + y = ?$$

$$1. \quad x + 0 \Rightarrow x = h(x, 0)$$

$$2. \quad x + (y + 1) \Rightarrow (x + y) + 1$$

Усечённое вычитание

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

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

Модуль разности

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

Умножение

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

Факториал

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

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

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

Определение 23. Рекурсивной называется функция типа

$$f : \mathbb{N}^p \rightarrow \mathbb{N} \quad (p \geq 1)$$

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

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

$$0 : \begin{cases} 01 \rightarrow 0 \\ 0 \rightarrow \cdot 0 \end{cases}$$

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

$$+ 1 : \begin{cases} 0 \rightarrow \cdot 01 \end{cases}$$

$$\prod_i (x_1 \$ x_2 \$ \dots \$ x_n) = x_i$$

$$(\forall i = \overline{1, n})(x_i - \text{КНЧ})$$

Композиция (подстановка)

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

Если предположить что

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

то

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

Рекурсия

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

$$h(\tilde{x}, 0) \Rightarrow A_f(\tilde{x}\$0) \cong A_n(\tilde{x}\$0)$$

$$h(\tilde{x}, y + 1) \Rightarrow A_g(\tilde{x}\$y\$A_h(\tilde{x}\$y))$$

Минимизация

<div style="display: flex; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 20px; text-align: right; padding-right: 5px;">1</div> <div><u>while</u> ($A_f(\tilde{x}\\$y) \neq 0$) <u>do</u></div> </div> <div style="display: flex; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 20px; text-align: right; padding-right: 5px;">2</div> <div>$y \leftarrow y + 1$</div> </div> <div style="display: flex;"> <div style="width: 20px; text-align: right; padding-right: 5px;">3</div> <div><u>end</u></div> </div>
--

Теорема 15. *Всякая рекурсивная функция нормально вычислима.*

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

Следствие 4.

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

где \mathcal{N} — класс нормально вычисляемых функций, \mathcal{R} — класс рекурсивных функций.

1.10. λ -исчисление

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

Примеры

$$\lambda x \cdot (x^2 + 3)a \triangleright_\beta a^2 + 3; \quad \lambda x \cdot (x^2 + 3) \triangleright_\beta 12$$

$$\lambda y \cdot (x^2 + y^2 + 3)a \triangleright_\beta x^2 + a^2 + 3$$

$$\lambda xy \cdot f(x, y) \equiv \lambda x \cdot (\lambda y \cdot f(x, y))$$

λ -терм X — множество переменных. $X = \{x_1, \dots, x_n, \dots\}$.

Определение 24 λ -терм.

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

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

Пример:

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

Рис. 14

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

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

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

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

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

$$[N|x]M \text{ или } [x := N]M$$

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

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

Правила:

1. $[N|x]x \equiv N$
2. $[N|x]y \equiv (y \neq x)$
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.

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

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

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

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

Определение 26 Понятие λ -конвертируемости. Терм M α 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. β -редукция

Определение 27 β -редекс. Применить функцию M к терму N .

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

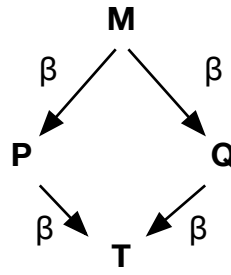


Рис. 15. Теорема Чёрча-Росса

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

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

Пример 19.

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

Пример 20 Неприводимый к нормальной форме терм.

$$(\lambda x \cdot xx)(\lambda x \cdot xx) \triangleright_\beta (\lambda x \cdot xx)(\lambda x \cdot xx) \triangleright_\beta \dots$$

Пример 21.

$$\begin{aligned} (\lambda xy \cdot M)XY &\equiv ((\lambda x \cdot (\lambda y \cdot M))X)Y \triangleright_\beta ([X|x]\lambda y \cdot M)Y \equiv (\lambda y \cdot [X|x]M)Y \triangleright_\beta \\ &\triangleright_\beta [Y|y][X|x]M \end{aligned}$$

1.10.2. Комбинаторы

Определение 29 Комбинатор. λ -терм, не содержащий свободных переменных, называется комбинатором (замкнутый терм).

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

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

Комбинатор истинности: $K \equiv T$

2. Ложь: $F \equiv \lambda xy \cdot y$, $FXY \triangleright_\beta Y$

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

3. $F \equiv \bar{0}$

4. Тожественная функция: $\mathbb{I} \equiv \lambda x \cdot x$; $\mathbb{I}X \equiv (\lambda x \cdot x)X \triangleright_\beta X$

5. $\bar{n} \equiv \lambda xy \cdot x^n y$, где $\underbrace{((\dots (x) \dots) x)}_n = x^n$; $\bar{n}XY \equiv (\lambda xy \cdot x^n y)XY \triangleright_\beta X^n Y$

6. Прибавление единицы: \oplus

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

7. Упорядоченная пара

$$\langle M, N \rangle \equiv \lambda z \cdot (zMN)$$

$$\begin{aligned} < M, N >_0 \equiv (\lambda z \cdot (zMN))\mathbb{T} \triangleright_\beta \mathbb{T}MN \triangleright_\beta M \\ < M, N >_1 \equiv (\lambda z \cdot (zMN))\mathbb{F} \triangleright_\beta \mathbb{F}MN \triangleright_\beta N \end{aligned}$$

8. КORTEЖ \oplus

$$\begin{aligned} < M_0, M_1, \dots, M_n > \equiv \lambda z \cdot (zM_0M_1 \dots M_n) \\ P_i^n < M_0, M_1, \dots, \end{aligned}$$

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

Пример 22.

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

Определение 30 λ -определимая функция. $f : \mathbb{N}^p \rightarrow \mathbb{N}$ называется λ -определимой, если может быть построен λ -терм M такой, что для любых $n_1, \dots, n_p \in \mathbb{N}$, $\bar{n}_1\bar{n}_2 \dots \bar{n}_p \triangleright_\beta \bar{f}(n_1, n_2, \dots, n_p)$, если $f(n_1, n_2, \dots, n_p)$ определено; и терм $M\bar{n}_1\bar{n}_2 \dots \bar{n}_p$ не имеет β -нормальной формы, если $f(n_1, n_2, \dots, n_p)$ не определена.

Теорема 18 Основная. *Функция λ -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:

$$\text{Int}(P, x) \cong P(x)$$

\oplus Если данные неизвестны, то получаем объектный код:

$$\lambda x. \text{Int}(P, x) = \lambda x. \text{Int}_P(x) \quad \text{Int}^* : P \mapsto \text{Int}_P$$

Получение транслятора:

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

$$\lambda \text{Int}Px. \text{mix}(\text{mix}, (\text{Int}, (P, x))) = \lambda \text{Int}Px. \text{mix}_{\text{mix}}(\text{Int}, (P, x)) \quad \text{mix}^* : \text{mix} \mapsto \text{mix}_{\text{mix}}$$

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 двойственности. *Каждое тождество симметричного полукольца остаётся справедливым, 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. $\mathcal{S}_M = (2^M, \cup, \cap, \emptyset, M)$; $\mathcal{S}_M^* = (2^M, \cap, \cup, M \setminus \emptyset)$
 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.

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

□

$$a \leq b \Leftrightarrow ab = a$$

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

$$(\forall a)(a \leq 1)$$

Доказательство. $a + 1 = 1 \Rightarrow a \leq 1$ □

Определение 31 Дополнение.

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

Не у всех элементов симметричного полукольца есть дополнения.

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

Доказательство.

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

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

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

□

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: $+$ $\rightarrow \vee, \cdot \rightarrow \wedge$,

$$B = (B, \vee, \wedge, , 0, 1)$$

1. $(B, \vee, \wedge, 0, 1)$ — симметричное полукольцо.

2. $a \vee \bar{a} = 1, a \wedge \bar{a} = 0$.

$$\Phi B = (\{0, 1\}, \vee, \wedge, 0, 1,) \quad \bar{0} = 1, \bar{1} = 0$$

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

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

$$\bar{\tilde{\alpha}} = ()$$

Определение 33 Булев куб. $B^n = (\{0, 1\}^n, \vee, \wedge, \tilde{0}, \tilde{1})$

$$\tilde{\alpha} \leq \tilde{\beta} \Leftrightarrow \tilde{\alpha} \vee \tilde{\beta} = \tilde{\beta}$$

$$\tilde{\alpha} = (\alpha_1, \dots, \alpha_n) \leq \tilde{\beta} = (\beta_1, \dots, \beta_n) \Leftrightarrow (\forall i = \overline{1, n})(\alpha_i \leq \beta_i), \text{ где } 0 < 1$$

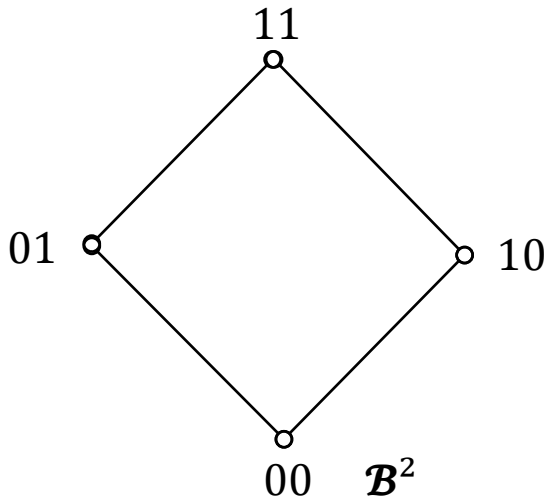
$$B = (B, \vee, \wedge, , 0, 1)$$

$$f : X \rightarrow B$$

$$(f \vee g)(x) \Leftrightarrow f(x) \vee g(x), \quad (f \wedge g)(x) = f(x) \wedge g(x)$$

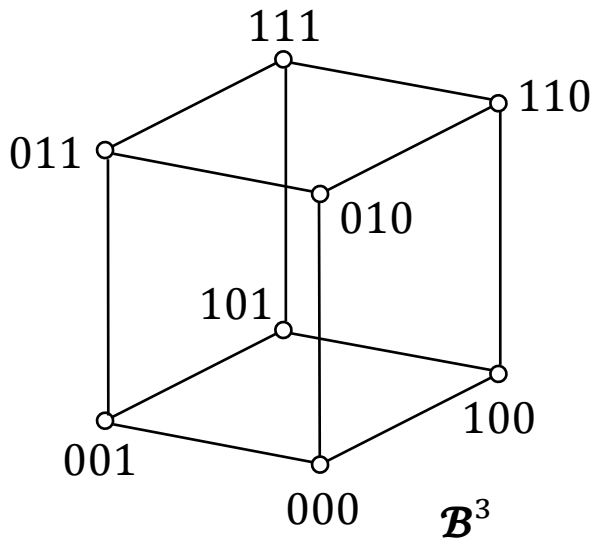
2.2. Булевы функции. Основные понятия, таблица булевой функции

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



10-1-square.: 0x0 pixel, 0dpi, 0.00x0.00 cm, bb=

Рис. 16. Булев квадрат



10-1-cube.: 0x0 pixel, 0dpi, 0.00x0.00 cm, bb=

Рис. 17. Булев квадрат

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

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

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: $|\mathcal{P}^{(n)}| = 2^{2^n}$.

$n = 0 : 0, 1$

	f_1	f_2	f_3	f_4
$n = 1 :$	0	1	0	1
	1	0	0	1

	x_1	x_2	\vee	\wedge	\rightarrow	\sim	\oplus	$ $	\downarrow
$n = 2 :$	0	0	0	0	1	1	0	1	1
	1	0	1	0	1	0	1	1	0
	2	1	0	1	0	0	1	1	0
	3	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:

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

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

2.3. Равенство булевых функций. Фиктивные переменные

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

Определение 36 Равенство функций.

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

1. $x_1 \rightarrow x_2 = \overline{x_1} \vee x_2$
2. $x_1 \sim x_2 = (x_1 \rightarrow x_2) \cdot (x_2 \rightarrow x_1) = \overline{x_1 \oplus x_2}$
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 $(\forall \tilde{\alpha}, \tilde{\beta})(\tilde{\alpha} = (\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n), \tilde{\beta} = (\beta_1, \dots, \beta_{i-1}, 0, \beta_{i+1}, \dots, \beta_n))(f(\tilde{\alpha}) = f(\tilde{\beta}))$

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

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

Определение 40 Равенство булевых функций. 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.

$$pr_i(x_1, \dots, x_n) \Leftrightarrow x_i, \quad 1 \leq i \leq n$$

Ввод фиктивных переменных для уравнивания количества переменных у двух функций:

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

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

Рис. 18. Суперпозиция

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

Определение 42.

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

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

Пример 23.

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

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

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

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

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

Пример 25 Базис Жегалкина.

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

Определение 43 Формула над базисом F .

1. Всякая переменная из X есть формула.
2. Если Φ_1, \dots, Φ_n — формулы, а $f^{(n)} \in \mathcal{F}^{(n)}$, то $f^{(n)}(\Phi_1, \dots, \Phi_n)$ — формула.
3. Других формул нет.

Пример 26.

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

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 Φ_1 представляет функцию g_1, \dots , формула Φ_n представляет функцию g_n , а $f^{(n)} \in \mathcal{F}^{(n)}$, то $f^{(n)}(\Phi_1, \dots, \Phi_n)$ представляет $\Phi \cap f^{(n)}(g_1, \dots, g_n)$

□

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).

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

ДНФ

Определение 47 Элементарная конъюнкция. Элементарная конъюнкция — конъюнкция литералов $\tilde{x}_{i_1}, \tilde{x}_{i_2}, \dots, \tilde{x}_{i_k}, i_1, \dots, i_k \subseteq \{1, 2, \dots, n\}$

Определение 48 ДНФ. $K_1 \vee K_2 \vee \dots \vee K_m, m \geq 1$

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 $\tilde{x}_{i_1}, \tilde{x}_{i_2}, \dots, \tilde{x}_{i_k}, i_1, \dots, i_k \subseteq \{1, 2, \dots, n\}$

Определение 50 КНФ. $D_1 \cdot D_2 \cdot \dots \cdot D_m, m \geq 1$

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 $y = f(x_1, \dots, x_n) \neq 0$.

$$\begin{aligned} C_f^{(1)} &\Rightarrow \{\tilde{\alpha} : f(\tilde{\alpha}) = 1\} \neq \emptyset \\ K_{\tilde{\alpha}} &= x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}, \text{ где } \tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n) \\ K_{\tilde{\alpha}}(\tilde{\beta}) &= 1 \Leftrightarrow \tilde{\beta} = \tilde{\alpha} \\ f(\tilde{\alpha}) &= 1 \Leftrightarrow \tilde{\alpha} \in C_f^{(1)} \end{aligned}$$

Таким образом:

⊕

□

см. учебник

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

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

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

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

$$\begin{aligned} h(x) &\Rightarrow f_S(x^{\alpha_1}, \dots, x^{\alpha_n}); \\ h(0) &= f_S(0^{\alpha_1}, \dots, 0^{\alpha_n}) = f_S(\overline{\tilde{\alpha}}), \text{ так как } 0^1 = 0, 0^0 = 1 \\ h(1) &= f_S(1^{\alpha_1}, \dots, 1^{\alpha_n}) = f_S(\tilde{\alpha}) \Rightarrow h(0) = h(1) = \text{const} \in \{0, 1\} \end{aligned}$$

□

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

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

и

$$\tilde{\beta} = (\beta_1, \dots, \beta_{i-1}, 0, \beta_{i+1}, \dots, \beta_n)$$

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 $f(\tilde{\gamma}) = 1, f(\tilde{\delta}) = 0$.

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

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

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

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

Где-то должен произойти скачок с 1 до 0.

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

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.

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

где

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

□

Лемма 5 о нелинейной функции. Если $f_L \notin L$, то конъюнкцию можно представить формулой над $\{f_L, 0, 1\}$

Доказательство.

□

3. Элементы математической логики

3.1. Понятие формальной аксиоматической теории

<Пропущенная по болезни лекции>

Теорема 22. Если $\Gamma \vdash_{\mathcal{T}} \Phi$, то для любого $\Gamma' \supset \Gamma$ $\Gamma' \vdash_{\mathcal{T}} \Phi$

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

Пример 27.

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

где

$$V_0 = Atom \cup Var \cup \{+, *, -\} \cup Aux$$

$$Atom = \{0, 1, \dots, 9\}, \quad Var = \{a, b, \dots, z\}$$

\mathcal{F}_0 :

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

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. Тавтологически 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 28 Доказательство доказательства от противного.

$$\begin{array}{c} \underbrace{(\neg q \rightarrow \neg p)}_{\text{И}} \rightarrow \underbrace{((\neg q \rightarrow p) \rightarrow q)}_{\text{Л}} \\ \underbrace{(\neg q \rightarrow p)}_{\text{И}} \rightarrow \underbrace{q}_{\text{Л}} = \text{Л} \Rightarrow q = \text{Л} \end{array}$$

Пусть F_1, \dots, F_n, G — высказывания. Говорят, что G является логическим следствием F_1, \dots, F_n , если

$$F_1 \& \dots \& F_n \rightarrow G \text{ — тавтология}$$

Пример 29 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»

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

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

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

3.3. Исчисление высказываний

$$\begin{array}{l} L = (V_L, \mathcal{F}_L, \mathcal{A}_L, \mathcal{P}_L \\ V_L = \text{Var} \cup \{\text{Л}, \text{И}\} \cup \{\neg, \rightarrow\} \cup \text{Aux} \end{array}$$

\mathcal{F}_L :

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

1. $A \rightarrow (B \rightarrow A)$
2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

3. $(\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B)$
4. $\mathcal{P}_L : \frac{A, A \rightarrow B}{B}$ — правило отсечения.
5. $A \vee B = \neg A \rightarrow B$
6. $A \& B = \neg(A \rightarrow \neg B)$

Теорема 23. $\vdash_L (A \rightarrow A)$

Доказательство.

1. $(A \rightarrow ((A \rightarrow A) \rightarrow A)) \rightarrow ((A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A))$ по схеме (2) при $B := (A \rightarrow A), C := A$
2. $A \rightarrow ((A \rightarrow A) \rightarrow A)$ по схеме (1) при $B := A \rightarrow A$
3. $(A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)$ МР (1) и (2)
4. $A \rightarrow (A \rightarrow A)$ по схеме (1) при $B := A$
5. $A \rightarrow A$ МР (3) и (4)

Итак, $\vdash (A \rightarrow A)$ □

3.4. Теорема дедукции

Теорема 24 Гербрамда. Если из $\Gamma \cup \{A\} \vdash B$, то $\Gamma \vdash (A \rightarrow B)$

Доказательство. Вместо $\Gamma \cup \{A\} \vdash B$ пишем $\Gamma, A \vdash B$.

Индукция по длине l вывода B из Γ, A .

Базис: $l = 0$, то есть

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

Предположение: пусть $\forall l \leq n-1 \quad \Gamma, A \vdash^l B \Rightarrow \Gamma \vdash (A \rightarrow B)$.

Переход: $l = n (n \geq 1)$

Существует в выводе B из Γ, A формулы: $\Phi, \Phi \rightarrow B$, то есть $\Gamma, A \vdash^m \Phi \rightarrow B$, где $m \leq n$ □

Справедлива обратная теорема.

Теорема 25. Если $\Gamma \vdash (A \rightarrow B)$, то $\Gamma, A \vdash B$

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

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

Пример 30. Докажем: $\vdash (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$

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

Итак, $\neg B \rightarrow \neg A, A \vdash B \quad \vdash (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B) \quad \neg B \rightarrow \neg A \vdash A \rightarrow B$

Теорема 26. В теории L имеют место следующие секвенции:

1. $A \rightarrow B, B \rightarrow C \vdash A \rightarrow C$
2. $A \rightarrow (B \rightarrow C), B \vdash A \rightarrow C$
3. $\vdash (\neg\neg A \rightarrow A)$
4. $\vdash (A \rightarrow \neg\neg A)$
5. $\vdash (A \rightarrow (\neg A \rightarrow B))$
6. $\vdash (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$
7. $\vdash (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$
8. $\vdash A \rightarrow (\neg B \rightarrow \neg(A \rightarrow B))$
9. $\vdash (A \rightarrow B) \rightarrow ((\neg A \rightarrow B) \rightarrow B)$

Доказательство. (1)

1. $A \rightarrow B$ — гипотеза
2. $B \rightarrow C$ — гипотеза
3. A — гипотеза
4. B — МР, (1) и (3)
5. C — МР, (2) и (4)

$A \rightarrow B, B \rightarrow C, A \vdash C$

(2)

1. $A \rightarrow (B \rightarrow C)$ — гипотеза
2. B — гипотеза
3. A — гипотеза
4. $B \rightarrow C$ — МР, (1) и (3)
5. C — МР, (2) и (4)

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

(3)

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

5. $(\neg A \rightarrow \neg A) \rightarrow A$ — МР, (2) и (4)

6. $\neg A \rightarrow \neg A$ — теорема

7. A — МР, (5) и (6)

$\neg\neg A \vdash A \Rightarrow \vdash (\neg\neg A \rightarrow A)$
(4)

1. A — гипотеза

2. $(\neg\neg\neg A \rightarrow \neg A) \rightarrow ((\neg\neg\neg A \rightarrow A) \rightarrow \neg\neg A)$ — схема (3) при $B := \neg\neg A$

3. $\neg\neg\neg A \rightarrow \neg A$ — секвенция (3)

4. $(\neg\neg\neg A \rightarrow A) \rightarrow \neg\neg A$ — МР, (2) и (3)

5. $A \rightarrow (\neg\neg\neg A \rightarrow A)$ — схема (1) при $B := \neg\neg\neg A$

6. $\neg\neg\neg A \rightarrow A$ — МР, (1) и (5)

7. $\neg\neg A$ — МР, (4) и (6)

$A \vdash \neg\neg A \Rightarrow \vdash (A \rightarrow \neg\neg A)$
(5) \oplus
(6) — см. 30
(7)

1. $A \rightarrow B$ — гипотеза

2. $\neg\neg A \rightarrow A$ — секвенция (3)

3. $\neg\neg A \rightarrow B$ — секвенция (1); (2) и (1)

4. $B \rightarrow \neg\neg B$ — секвенция (4)

5. $\neg\neg A \rightarrow \neg\neg B$ — секвенция (1), (3) и (4)

6. $(\neg\neg A \rightarrow \neg\neg B) \rightarrow (\neg B \rightarrow \neg A)$ — секвенция (6) при $A := \neg B, B := \neg A$

7. $\neg B \rightarrow \neg A$ — МР, (5) и (6)

(8) \oplus

(9)

1. $(A \rightarrow B)$ — гипотеза

2. $(\neg$

□

\oplus

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 $A \vee \Phi \vdash A \vee \Psi$ и $\Phi \vee A \vdash \Psi \vee A$

Доказательство. (1)

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

(2)⊕

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

(3)

□

Следствие 7 Свойство конъюнкции.

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

Доказательство. $A \& B = \neg(A \rightarrow \neg B)$

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

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

(2)

1. A — гипотеза
2. B — гипотеза
3. $A \rightarrow (\neg\neg B \rightarrow \neg(A \rightarrow \neg B))$ — секвенция (8) при $B := \neg B$
4. $\neg\neg B \rightarrow \neg(A \rightarrow \neg B)$ — МР, (1) и (3)
5. $B \rightarrow \neg\neg B$ — секвенция (4)
6. $B \rightarrow \neg(A \rightarrow \neg B)$ — секвенция (1); (5) и (4)
7. $\neg(A \rightarrow \neg B) = A \& B$ — МР, (2) и (6)

□

3.5. Непротиворечивость, полнота и разрешимость теории L

Теорема 27. Если $\vdash_L \Phi$, то Φ — тавтология.

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

Пусть Φ — тавтология, и $\Phi \rightarrow \Psi$ — тавтология.

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

Допустим, что при этих условиях формула Ψ не является тавтологией. Тогда

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

Следовательно: $(\Phi \rightarrow \Psi)(\tilde{\alpha}) = \Phi(\tilde{\alpha}) \rightarrow \Psi(\tilde{\alpha}) = \text{И} \rightarrow \text{Л} = \text{Л}$

Противоречие, так как $\Phi \rightarrow \Psi$ — тавтология. \square

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

⊕

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

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

Доказательство. Индукция по числу $l(\Phi)$ логических связок в формуле Φ .

Базис: $l(\Phi) = 0$, то есть $\Phi = x_i$. Тогда $x_i^{\alpha_i} \vdash x_i^{\alpha_i}$, что очевидно (так как $\vdash_L (A \rightarrow A)$)

Предположение: пусть лемма доказана $\forall l() \leq m - 1$ ($m \geq 1$)

Переход: положим $l() = m$

1 случай. $\Phi = \neg\Psi$, где $l(\Psi) = m - 1$

⊕

\square

Теорема 28. Теория L полна, то есть любая тавтология доказуема в L .

Доказательство. Пусть формула $\Phi = \Phi(x_1, \dots, x_n)$ — тавтология. Тогда для $(\forall \tilde{\alpha} = (\alpha_1, \dots, \alpha_n))(\Phi(\tilde{\alpha}) = \text{И})$, и по лемме $x_1^{\alpha_1}, \dots, x_n^{\alpha_n} \vdash \Phi$. От $\tilde{\alpha}$ переходим к $(\alpha_1, \dots, \alpha_{n-1}, \neg\alpha_n)$, но тогда $x_1^{\alpha_1}, \dots, x_{n-1}^{\alpha_{n-1}}, \neg x_n^{\alpha_n} \vdash \Phi$. Откуда, по (1) (сл. 1) из теоремы (26): $x_1^{\alpha_1}, \dots, x_{n-1}^{\alpha_{n-1}} \vdash \Phi$. Действуя таким образом, получим $x_1^{\alpha_1} \vdash \Phi$ и $\neg x_1^{\alpha_1} \vdash \Phi$, откуда $\vdash \Phi$. \square

Теорема 29. 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 Φ, Ψ называются эквивалентными, если они выводимы друг из друга:

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

То есть:

$$\Phi \equiv \Psi \Leftrightarrow (\Phi \rightarrow \Psi) \& (\Psi \rightarrow \Phi)$$

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

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

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

Доказательство.

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

$$1. \neg A \rightarrow B \equiv \neg A \rightarrow \neg\neg B$$

(а) $\neg A \rightarrow B$ — гипотеза

(б) $B \rightarrow \neg\neg B$ — секвенция (4)

(в) $\neg A \rightarrow \neg\neg B$ — секвенция (1); (1а) и (1б)

$$2. \neg A \rightarrow \neg\neg B \vdash \neg A \rightarrow B$$

(а) $\neg A \rightarrow \neg\neg B$ — гипотеза

(б) $\neg\neg B \rightarrow B$ — секвенция (3)

(в) $\neg A \rightarrow B$ — 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 Φ , в которой заменены все вхождения подформулы Θ' на Θ .

Теорема 30. 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 $\Gamma \vdash_L \Phi[\Theta'/\Theta]$

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

$$1. (\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B) \text{ — схема (3)}$$

$$2. (A \rightarrow B) \rightarrow ((\neg B \rightarrow A) \rightarrow B) \text{ — эквивалентность } (A \rightarrow B) \equiv (\neg B \rightarrow \neg A)$$

$$3. (A \rightarrow B) \rightarrow ((\neg A \rightarrow \neg\neg B) \rightarrow B) \text{ — эквивалентность } (\neg B \rightarrow A) \equiv (\neg A \rightarrow \neg\neg B)$$

$$4. ((A \rightarrow B) \rightarrow ((\neg A \rightarrow B) \rightarrow B)) \text{ — эквивалентность } \neg\neg B \equiv B$$

3.7. Понятие алгебраической системы

Пример 31. $x_1 + x_2 \geq x_1 \cdot x_2$ ($x_1, x_2 \in \mathbb{R}$) — истинно или ложно, в зависимости от x_1, x_2 .

Определение 55 предикат арности n . $p : A^n \rightarrow \{И, Л\}$, $n \geq 1$

Унарные предикаты ($p^{(1)}$) есть высказывания.

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

Пусть $p \in \Pi^{(n)}$, $\Pi = \Pi^{(1)} \cup \Pi^{(2)} \cup \dots \Pi^{(n)} \cup \dots$

$$\rho_p \Rightarrow \{(x_1, \dots, x_n) : p(x_1, \dots, x_n) = И\} \quad p(x_1, x_2) = И \Leftrightarrow x_1 \leq x_2 \quad (x_1, x_2 \in \mathbb{R})$$

$$\rho_p \subseteq A^n$$

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

$$\text{Пусть } \omega : A^n \rightarrow A \mapsto p_\omega \Rightarrow \{(x_1, \dots, x_n, x_{n+1}) : x_{n+1} = \omega(x_1, \dots, x_n)\} \subseteq A^{n+1}, \quad n \geq 0$$

$$\text{Так можно определить: } + : \mathbb{R}^2 \rightarrow \mathbb{R} \mapsto p_+ \Rightarrow \{(x_1, x_2, x_3) : x_3 = x_1 + x_2\}$$

Примеры алгебраических систем:

$$1. (\mathbb{R}, +, \cdot, 0, 1; \leq)$$

$$2. (Z, +, \cdot, 0, 1; \leq, | \quad m|n \Leftrightarrow n = km, k \in Z)$$

$$3. \mathcal{G} = (V, \rho), \rho \in V^2$$