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

лекции

Логика и теория алгоритмов

Алексей Иванович Белоусов

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

27 марта 2013 г.

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.

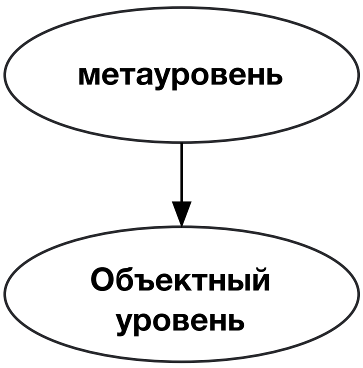

Рис. 1. Выход за пределы математики

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.

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 : $\quad A : X \to Y$

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

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: $V, W \quad f : V^* \underset{\cdot}{\to} W^*$

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 $A : V^* \underset{\cdot}{\to} W^*$. 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.

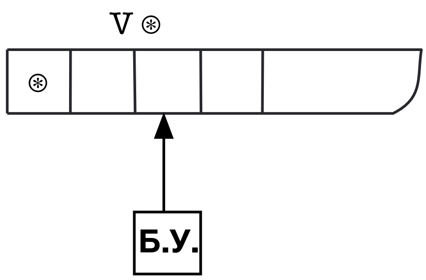

Рис. 3. Машина Тьюринга

Машина Тьюринга — полубесконечная лента, разделённая на буквы.

$\circledast$ — маркер начала ленты.

$\square$ — символ пробела.

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

Запись команды

$$qa \to rb, \begin{Bmatrix} S \\ L \\ R \end{Bmatrix} \quad q, r \in Q,\ a, b \in V \cup \{\circledast, \square\}$$

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).

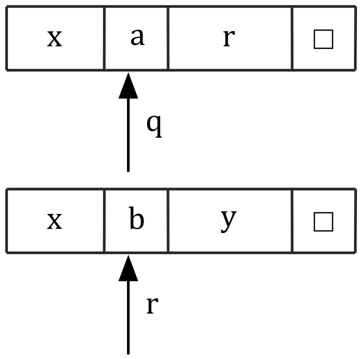

Рис. 4. Запись команды

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

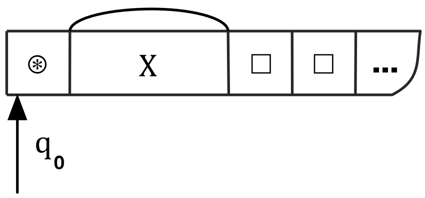

Рис. 5. Машина Тьюринга со входным словом

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

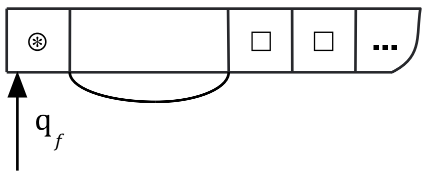

Рис. 6. Окончание работы машины Тьюринга

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?

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

Ответ: стирает слова, которые содержат букву c.

Формально машина Тьюринга определяется как следующий кортеж:

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

где δ — система команд:

$$
\begin{aligned}
&\delta : Q \times V' \to 2^{Q\times V'\times\{S,L,R\}}, \text{ где } V' = V \cup \{\circledast, \Box\} \\
&\delta : Q \times V' \underset{\cdot}{\to} Q \times V' \times \{S, L, R\}
\end{aligned}
$$

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

Определение 2 Конфигурация.

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

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:

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

и конечная конфигурация:

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

Определение 3 Отношение непосредственной выводимости.

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

Определение 4 Выводимость на множестве конечных конфигураций машины Тьюринга.

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

В детерминированной машине Тьюринга из каждой конфигурации
Пусть дана машина Тьюринга $\mathcal{T}$ и слово $x \in V^*$

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

Определение 5. Вербальная функция

$$f : V^* \mathrel{\underset{\cdot}{\to}} V^*$$

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.

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

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

Прогонка:

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

Пример 3.

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

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

Прогонка:

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

Пример 4.

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

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

Прогонка:

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

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

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

из предыдущего примера вставим:

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

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.

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

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 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.

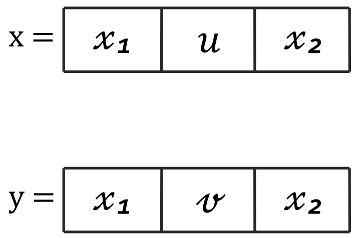

Рис. 7. Формула подстановки

Например, $x =$ входит, ω : вход $\to$ 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:

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

Пример 7 Добавление aba в начало слова.

$$\mathcal{A}_1 : \left\{ \ \to\cdot\, aba \right.$$
$$(\forall x \in \{a, b\}^*)(\mathcal{A}_1 : x \vdash\cdot\, abax)$$

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 ω — 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 ω — 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

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

$!\mathcal{A}(x)$ — 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:

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

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 $x_n = x_0, x_1, \dots, x_n$ — 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

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

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.

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

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

Пример 9 Удвоение слова.

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

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

Прогонка:

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

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

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

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

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

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.

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

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:

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

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

Утверждение 1.

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

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

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

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.

$$[a_i \rightleftharpoons 0\underbrace{11\ldots1}_{i}0 \quad x \in V^*[\lambda = \lambda, [x(1)x(2)\ldots x(k) \rightleftharpoons [x(1)\ldots[x(k)$$
$$V = \{a, b, c\}[abca = 010011001110010$$

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 $\overline{V} = \{\overline{a_1}, \dots, \overline{a_n}\}$, где $V = \{a_1, \dots, a_n\}$, и $\overline{V} \cap V = \emptyset \quad \alpha, \beta \notin V \cup \overline{V}$

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

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

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

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

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

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

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

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

$\alpha\bar{y} \models_{(8)} \alpha\bar{z}_1\beta\bar{z}_2$, 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.

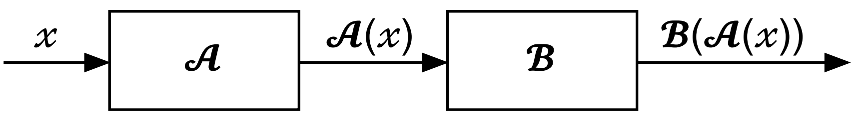

Рис. 8. Композиция

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:

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

1.5.2. Объединение

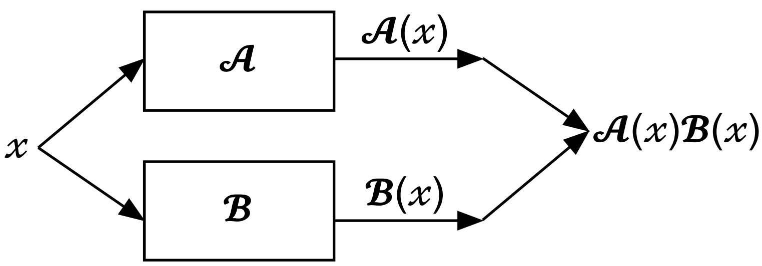

Рис. 9. Объединение

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

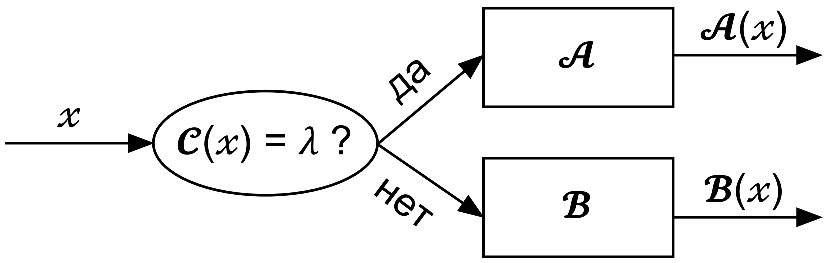

Рис. 10. Разветвление

```
if C(x) = λ then
   y ← A(x)
else
   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*

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

1.5.4. Повторение

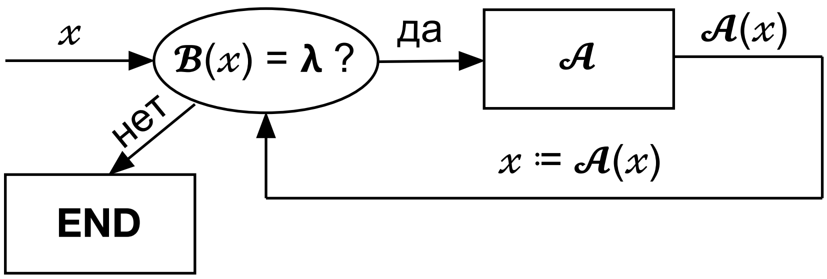

Рис. 11. Повторение алгорифма A, управляемого алгорифмом B

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*

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

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

```
while (B(x) = λ) do
   x ← A(x)
end
```

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

```
while (B(x) ≠ λ) do
   x ← A(x)
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.

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

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

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

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

Пример 12 Распознавание равенства слов.

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

1.6. Универсальный нормальный алгорифм

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

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

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

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

1.7. Разрешимые и перечислимые множества (языки)

Определение 14. 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 15 Полуразрешающий алгорифм.

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

Теорема 8. *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.

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

□

Пример 13 Язык двойных слов.

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

Докажем, что язык является разрешимым.

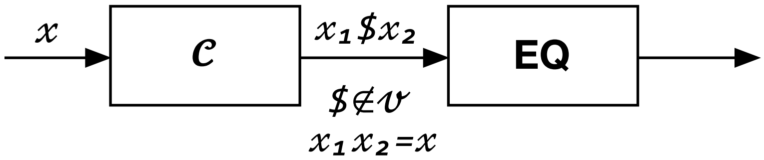

Рис. 12. Язык двойных слов

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

Пример 14.

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

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

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

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

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

1.7.1. Конструктивные числа

Определение 16 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 17 Конструктивное целое число. Это слово вида $[-]n$, где n — конструктивное натуральное число, то есть слово в алфавите $V_0 \cup \{-\}$

Определение 18 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 19 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 15.

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

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

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

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

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

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

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

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

Определение 20 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 9. *Всякий алгорифмически разрешимый язык является алгорифмически перечислимым (но обратное — неверно).*

Теорема 10 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

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

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

$$!\mathcal{B}([\![\mathcal{A}]\!]) \text{ и } \mathcal{B}([\![\mathcal{A}]\!])?$$

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 16 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 11. *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 $(\forall x \in V_0^*)(\mathcal{B}_1(x) \approxeq \mathcal{B}(x))$

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 12. *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 $\mathcal{U}([\mathcal{C}]\$y) \approxeq \mathcal{C}(y)(\$ \notin V_2))$. Строим нормальный алгорифм $\mathcal{U}_1$ так, что $(\forall y \in V_2^*)(\mathcal{U}_1(y) \approxeq \mathcal{U}(y\$y))$. 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 $\varrho \subseteq V^+ \times V^+$. $\$, \# \notin V$.

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

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

$V = \{a, b\}, \quad \varrho = \{(aba, ab), (b, ab)\}$. 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 13 Райса. *Язык L алгоритмически неразрешим.*

1.9. Рекурсивные функции

Базовые функции:

1. $(\forall x \in \mathbb{N})\ \mathbb{O}(x) = 0)$
2. $(x \in \mathbb{N})\ \ +\!\not\Vdash(x) \rightleftharpoons x + 1$
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 21. Рекурсивной называется функция типа

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

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 обожествляется с частичной рекурсивностью.

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

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

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

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

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

Композиция (подстановка)

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

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

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

$$h(\tilde{x}, 0) \rightleftharpoons A_f(\tilde{x}\$0) \approx A_n(\tilde{x}\$0)$$

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

Минимизация

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

Теорема 14. *Всякая рекурсивная функция нормально вычислима.*

Теорема 15. *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. $\boldsymbol{\lambda}$-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

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

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

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

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

Определение 22 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 $\underbrace{\lambda x}_{\lambda\text{-астракция}} \cdot \underbrace{M}_{\text{область действия}}$ — λ-терм (абстракция).
4. Других λ-термов не существует.

Определение 23 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:

$$[N|x]M\text{или}[x := N]M$$

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

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 17.

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

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

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

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

Определение 24 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 25 β-редекс. Применить функцию M к терму N.

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

Теорема 16 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 $Q \rhd \beta T$.*

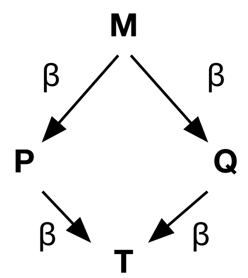

Рис. 15. Теорема Чёрча-Росса

Определение 26. Терм, не содержащий редексов называется приведённым к нормальной форме.

Пример 18.

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

Пример 19 Неприводимый к нормальной форме терм.

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

Пример 20.

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

1.10.2. Комбинаторы

Определение 27 Комбинатор. λ-терм, не содержащий свободных переменных, называется комбинатором (замкнутый терм).

1. $K \equiv \lambda xy \cdot x$
$$KXY \equiv (\lambda xy \cdot x)XY \rhd_\beta [Y|y][X|x]x \rhd_\beta X$$
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 $\underbrace{((\dots(x)\dots)x)x}_{n} = x^n$; $\overline{n}XY \equiv (\lambda xy \cdot x^n y)XY \rhd_\beta \mathrm{X}^n Y$
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 Ⓕ
$$< M_0, M_1, \dots, M_n > \equiv \lambda z \cdot (zM_0 M_1 \dots M_n)$$
$$P_i^n < M_0, M_1, \dots,$$
9. $\mathbb{B} \equiv \lambda xyz \cdot x(yz)$Ⓕ

Пример 21.

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

Определение 28 λ-определимая функция. $f : \mathbb{N}^p \dot{\to} \mathbb{N}$ называется λ-определимой, если может быть построен λ-терм M такой, что для любых $n_1, \dots, n_p \in \mathbb{N}, \quad \overline{n}_1\overline{n}_2 \dots \overline{n}_p \rhd_\beta \overline{f}(n_1, n_2, \dots, n_p)$, если $f(n_1, n_2, \dots, n_p)$ определено; и терм $M\overline{n}_1\overline{n}_2 \dots \overline{n}_p$ не имеет β-нормальной формы, если $f(n_1, n_2, \dots, n_p)$ не определена.

Теорема 17 Основная. *Функция λ-определима, если она рекурсивна.*

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

$P^* : x$ [U+21A6] P_x

Условие корректности смешанного вычисления:

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

Режим работы чистого интерпретатора:

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

ⓕ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:

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

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