

Элементы Теории Алгоритмов

1.1 Понятие алгоритма в интуитивном смысле слова

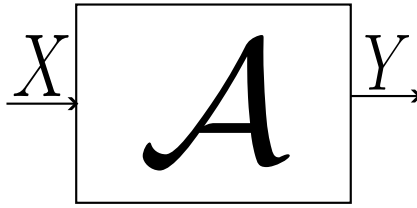


Рис. 1.1: Команда

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

Признаки алгоритма:

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

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

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

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

Определение 3. Конструктивный объект - слово в конечном алфавите.

Определение 4. Вербальная, или словарная, функция - это

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

Вербальная функция (V, W) .

Определение 5. Алгоритм можно записать так:

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

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

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

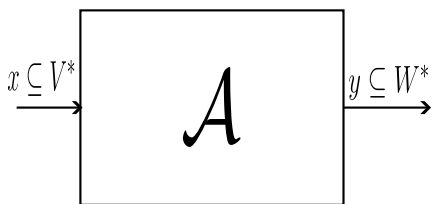


Рис. 1.2: Автомат

1.2 Машина Тьюринга.

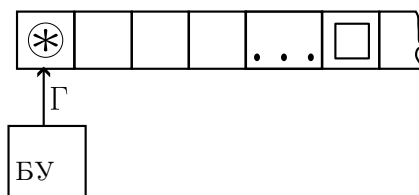


Рис. 1.3: Машина Тьюринга

Команды следующего формата:

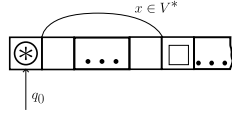
$$qa \rightarrow rb, \begin{cases} S \\ L \\ R \end{cases}; q, r \in Q; a, b \in V \cup \{\otimes, \square\}$$

$$\begin{array}{|c|c|c|c|} \hline \otimes & \dots & a & \dots \\ \hline \end{array} \vdash \begin{cases} \begin{array}{|c|c|c|c|} \hline \otimes & \dots & b & \dots \\ \hline \end{array} \begin{array}{c} \uparrow_r \\ \text{если } S \end{array} \\ \begin{array}{|c|c|c|c|} \hline \otimes & \dots & c & b \dots \\ \hline \end{array} \begin{array}{c} \uparrow_r \leftarrow \\ \text{если } L \end{array} \\ \begin{array}{|c|c|c|c|} \hline \otimes & \dots & b & c \dots \\ \hline \end{array} \begin{array}{c} \rightarrow \uparrow_r \\ \text{если } R \end{array} \end{cases}$$

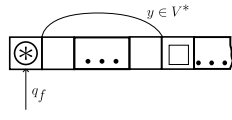
Рис. 1.4: Что к чему

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

Начальная конфигурация:



Заключительная конфигурация:



Пример программы:

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

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

Определение 7. Машина Тьюринга (МТ):

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

Конфигурация МТ:

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

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

Мы полагаем, что

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

Определение 8. Вывод на множестве конфигураций:

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

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

Дано:

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

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

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

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

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

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

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

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

Пример.

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

λ - Пустое слово.

Тогда программа записывается так:

$$\begin{aligned}
q_0\otimes &\rightarrow q_0\otimes, R \\
q_0\Box &\rightarrow q_f\#, L \\
q_0a &\rightarrow q'_0a, R \\
q_0b &\rightarrow q'_0b, R \\
q_0c &\rightarrow q_1c, R \\
q'_0a &\rightarrow q'_0a, R \\
q'_0b &\rightarrow q'_0b, R \\
q'_0c &\rightarrow q_1c, R \\
q_1a &\rightarrow q_2a, R \\
q_1b &\rightarrow q'_0b, R \\
q_1c &\rightarrow q_1c, R \\
q_2a &\rightarrow a'_0a, R // caa \\
q_2b &\rightarrow q_3b, R // cab \\
q_2c &\rightarrow q_1c, R // cac \\
q_3\alpha &\rightarrow q_3\alpha, R // \alpha \in \{a, b, c\} \\
q_3\Box &\rightarrow q_4\Box, L \\
q_4\alpha &\rightarrow q_4\Box, L \\
q_4\otimes &\rightarrow q_f\otimes, S \\
r\Box &\rightarrow q_5\Box, L // r \in \{q'_0, q_1, q_2\} \\
q_5\alpha &\rightarrow q_5\alpha, L \\
q_5\otimes &\rightarrow q_f\otimes, S
\end{aligned}$$

Для ошибочного решения (q'_0 не вводится):

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

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

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

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

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

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

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

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

Общие разделы:

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

1.3 Нормальные алгоритмы Маркова

Предположим, что есть

$$V; x, y \in V^*; x \sqsubseteq y \Leftrightarrow (\exists y_1, y_2)(y = y_1xy_2)$$

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

Некоторые свойства:

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

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

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

Еще один: $\underbrace{\text{абрака}}_x \underbrace{\text{дабра}}_x$

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

Определение 13. Подстановка:

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

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

$$\omega : u \rightarrow v$$

Тогда вхождение:

$$x = x_1ux_2; x_1 * u * x_2 \text{ - 1-е вхождение } u \text{ в } x$$

Отсюда

$$y \Leftarrow \omega x \Leftarrow x_1vx_2$$

Это можно представить так:

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

Пример. Пусть дана замена:

$$\omega : B \rightarrow y$$

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

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

Пример.

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

Рассматриваем систему сверху вниз и ищем первую подходящую формулу. Пусть

$$x = bbab$$

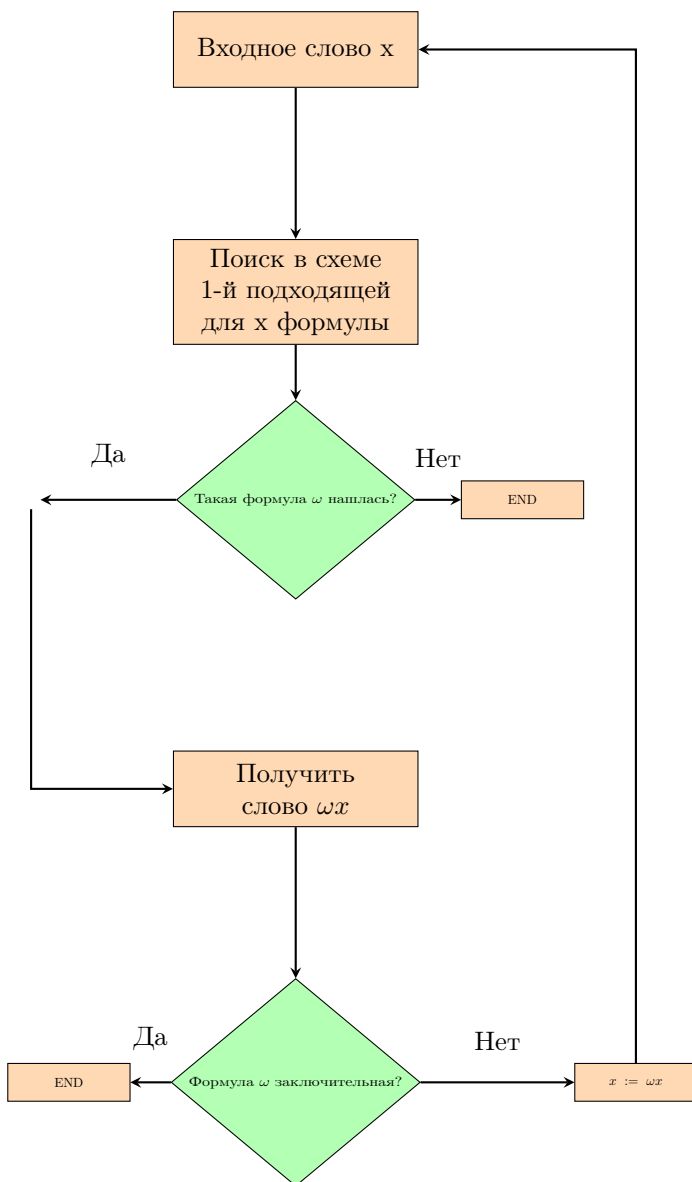
Отсюда получаем:

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

Общий вид:

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

Можно записать это в виде блок-схемы неформально:



Теперь формально опишем его. Распишем 5 разных ситуаций.

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

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

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

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

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

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

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

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

Примеры НА. Первый пример.

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

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

Второй пример.

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

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

Третий пример.

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

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

Четвертый пример.

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

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

Пятый пример.

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

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

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

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

Третий тест:

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

Можно строго доказать, что

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

1.4 Эквивалентность нормальных алгоритмов. Теорема о переводе.

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

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

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

Это называется условным равенством:

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

Рассмотрим такую конструкцию, называемую замыканием НА.

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

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

То есть

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

Рассмотрим преобразования:

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

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

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

Если $!A(x)$, то $x \models \bullet A(x)$ (система A замкнутая)

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

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

Формальное распространение НА на более широкий алфавит. $A = (V, S, P)$ в алфавите V .

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

Получаем:

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

Нам нужно расширить алфавит. Как это делается?

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

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

Пример.

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

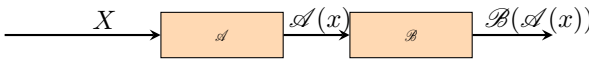
$$V_\alpha = \{\alpha, \beta\}$$

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

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

1.5 Теорема сочетания

1.5.1 Композиция



Теорема 1.3. (О композиции). Каковы бы ни были НА A, B в алфавите V может быть построен НА алгорифм C над алфавитом V такой, что

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

Доказательство. Вводится алфавит двойников.

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

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

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

A^\bullet	A^α
$u \rightarrow v$	$u \rightarrow v$
$u \rightarrow \bullet v$	$u \rightarrow \alpha v$

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

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

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

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

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

где $y_1 y_2 = y$. Далее получаем

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

Следующий, третий шаг

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

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

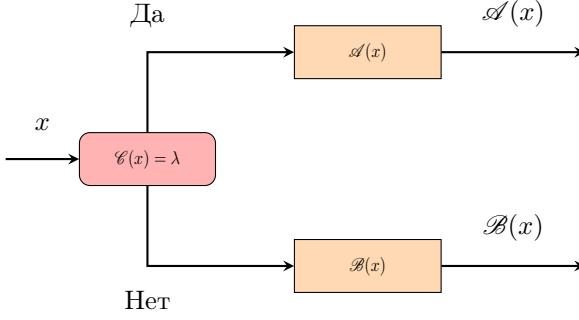
$$\alpha \bar{z}_1 \beta \bar{z}_2 \models_{(4)} \alpha \beta \overline{z_1 z_2} = \alpha \beta \bar{z} \models_{(5),(6)} \alpha \beta z \vdash \bullet z = B^\bullet(y) = B^\bullet(A^\bullet(x)) = B(A(x))$$

□

Пример.

$$A^\bullet : \begin{cases} \# \alpha \rightarrow \alpha \# \\ \# \beta \rightarrow \beta \# \\ \# \rightarrow \bullet aba \\ \rightarrow \# \\ \rightarrow \bullet \end{cases}$$

1.5.3 Разветвление



Записать в виде псевдокода можно так:

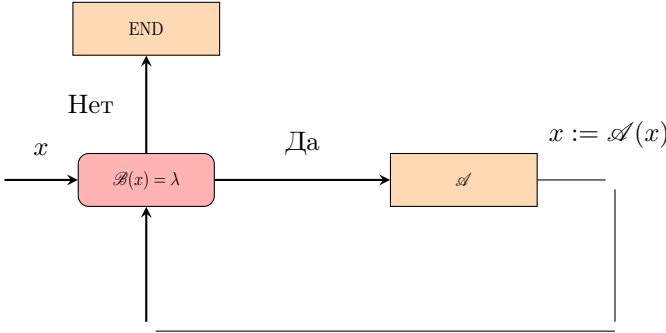
$if(C(x) = \lambda) \text{ then } y := A(x) \text{ else } y := B(x);$

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

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

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

1.5.4 Повторение



В виде псевдокода:

- Для цикла с условием, пока правда:

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

- Для цикла с условием, пока неправда:

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

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