Элементы Теории Алгоритмов

1.1 Понятие алгоритма в интуитивном смысле слова

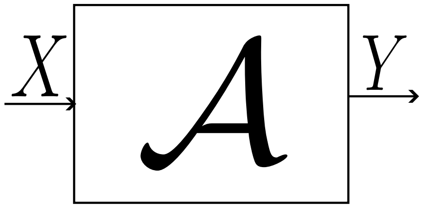


Рис. 1.1: Команда

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

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

$$f : V^* \underset{\bullet}{\to} W^*$$

Вербальная функция (V, W).

Определение 5. Алгоритм можно записать так:

$$\mathcal{A} : V^* \underset{\bullet}{\to} W^*$$

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

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

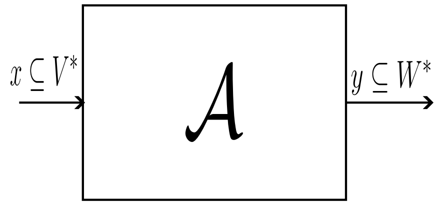


Рис. 1.2: Автомат

1.2 Машина Тьюринга.

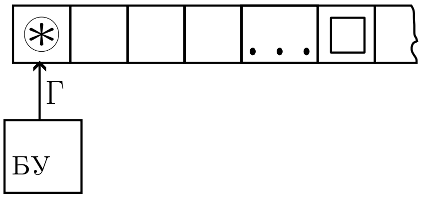


Рис. 1.3: Машина Тьюринга

Команды следующего формата:

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

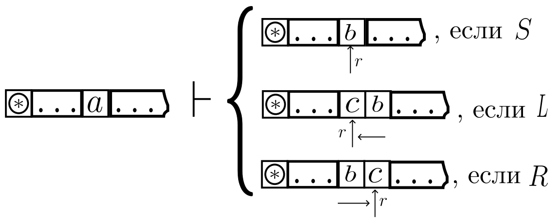
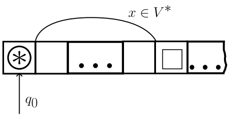


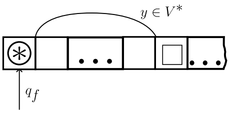
Рис. 1.4: Что к чему

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

Начальная конфигурация:



Заключительная конфигурация:



Пример программы:

$$q_0\circledast \to q_0\circledast, R$$
$$q_0 a \to q_0 a, R$$
$$q_0 b \to q_0 b, R$$
$$q_0 c \to q_1 c, R$$
$$q_1 a \to q_2 a, R$$
$$q_1 b \to q_0 b, R$$
$$q_1 c \to q_1 c, R$$
$$q_2 a \to q_0 a, R$$
$$q_2 b \to q_3 b, R$$
$$q_2 c \to q_1 c, R$$
$$q_3\alpha \to q_3\alpha, R \; //\alpha \in \{a, b, c\}$$
$$q_3\Box \to q_4\Box, R$$
$$q_i\Box \to q_5\Box, L \; //i = 0, 1, 2$$
$$q_4\circledast \to q_f 1, L$$
$$q_5\alpha \to q_5\Box, L$$
$$q_5\circledast \to q_5 *, R$$
$$q_5\Box \to q_f 0, L$$

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

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

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

Определение 8. Вывод на множестве конфигураций:

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

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

Дано:
Начальная конфигурация $C_0 = (q_0, \lambda, \circledast x \square)$, где $x \in V^*$
Конечная конфигурация $C_f = (q_f, \lambda, \circledast y \square)$, где $y \in V^*$

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

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

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

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:

$$
\begin{aligned}
&q_0\circledast \to q_0\circledast, R\\
&q_0\Box \to q_f\#, L\\
&q_0a \to q_0'a, R\\
&q_0b \to q_0'b, R\\
&q_0c \to q_1c, R\\
&q_0'a \to q_0'a, R\\
&q_0'b \to q_0'b, R\\
&q_0'c \to q_1c, R\\
&q_1a \to q_2a, R\\
&q_1b \to q_0'b, R\\
&q_1c \to q_1c, R\\
&q_2a \to a_0'a, R\ //caa\\
&q_2b \to q_3b, R\ //cab\\
&q_2c \to q_1c, R\ //cac\\
&q_3\alpha \to q_3\alpha, R\ //\alpha \in \{a, b, c\}\\
&q_3\Box \to q_4\Box, L\\
&q_4\alpha \to q_4\Box, L\\
&q_4\circledast \to q_f\circledast, S\\
&r\Box \to q_5\Box, L\ //r \in \{q_0', q_1, q_2\}\\
&q_5\alpha \to q_5\alpha, L\\
&q_5\circledast \to q_f\circledast, S
\end{aligned}
$$

Для ошибочного решения (q_0' не вводится):

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

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 коммaнд 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 $f : V^* \underset{\bullet}{\to} V^*$ 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 функциию. Основная гипотеза.
3. Эквивалентный алгоритм.
4. Теорема сочетания.
5. Универсальный алгоритм.
6. Разрешимые перечислимые множества (языки).
7. Анализ алгоритмически неразрешимых задач.

1.3 Нормальные алгорифмы Маркова

Предположим, что есть

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

причем тройка слов $(y1, x, y2)$ - 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: $y = \underbrace{\text{вход}}_{x}\text{ит}$; $*\text{вход}*\text{ит}$ - корень

Еще один: $\underbrace{\text{абра}}_{x}\text{када}\underbrace{\text{абра}}_{x}$

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

Определение 13. Подстановка:

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

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

$$\omega : u \to v$$

Тогда вхождение:

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

Отсюда

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

Это можно представить так:

$$\begin{array}{ll} x & = \boxed{\text{x}_1}\boxed{\text{u}}\boxed{\text{x}_2} \\ & \qquad\ \ \downarrow \\ y = \omega x & = \boxed{\text{x}_1}\boxed{\text{v}}\boxed{\text{x}_2} \end{array}$$

Пример. Пусть дана замена:

$$\omega : B \to y$$

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

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

Пример.

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

Рассматриваем систему сверху вниз и ищем первую подходящую формулу. Пусть

$$x = bbab$$

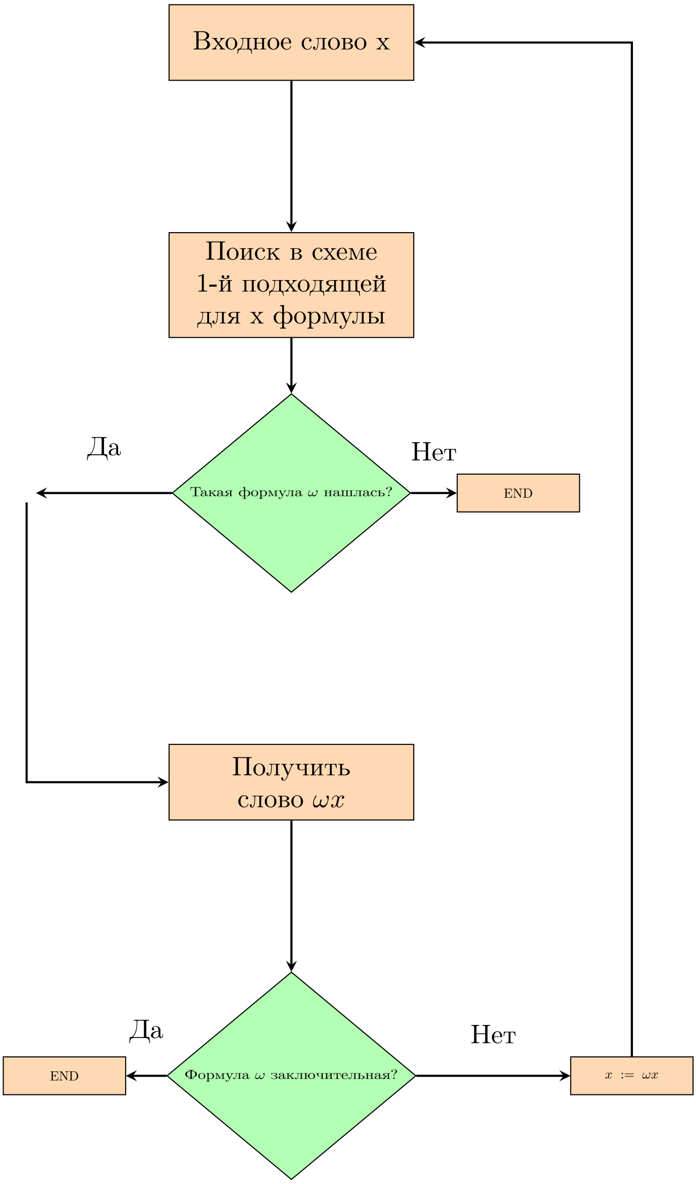
Отсюда получаем:

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

Общий вид:

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

Можно записать это в виде блок-схемы неформально:



Теперь формально опишем его. Распишем 5 разных ситуаций.

1) $\mathcal{A} : x \vdash y \leftrightharpoons$ непосредственно просто переводит слово x в слово y $\leftrightharpoons y = \omega x$, где ω - 1-я в схеме $\mathcal{A}$ формула, которая оказывается простой

2) $\mathcal{A} \vdash \cdot y \leftrightharpoons$ Алгорифм A непосредственно заключительно переводит слово x в слово y $\leftrightharpoons y = \omega x$, где ω - 1-я в схеме $\mathcal{A}$, которая оказывается заключительной

3) $\mathcal{A}x \models y \leftrightharpoons$ Алгорифм A переводит слово x в слово y, когда существует последовательность $x = x_0, x_1, \ldots, x_n = y$, где $(\forall i = \overline{0, n-1})(\mathcal{A} : x_i \vdash x_{i+1})$

4) $\mathcal{A} : x \models \cdot y \leftrightharpoons$ Алгорифм A заключительно переводит слово x в слово y $\leftrightharpoons \mathcal{A} : x \vdash \cdot y \vee (\exists z)(\mathcal{A} : x \models z \vdash \cdot y)$

5) $\sim \mathcal{A}(x) \leftrightharpoons$ в схеме A нет ни одной подходящей формулы для x.

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

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

Если процесс работы НА $\mathcal{A}$ со словом конечный, то есть $x = x_0, x_1, \ldots, x_n, n \geq 0$, то $!\mathcal{A}(x)$ и $x_n \leftrightharpoons \mathcal{A}(x)$. 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 алгорифм $\mathcal{A} = (V, S, P)$. Тогда мы говорим, что это НА в алфавите V. Пусть $\mathcal{A}_1 = (V_1 \subset V, S_1, P_1)$ - нормальный алгорифм над алфавитом V.

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

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

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

Примеры НА. Первый пример.

$$\mathcal{J}\alpha : \Big\{ \to \bullet$$

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

Второй пример.

$$Null : \Big\{ \to$$

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

Третий пример.

$$Lc : \Big\{ \to \bullet x_0, \text{ где } x_0 \in V^* \text{ - } \underline{\text{фиксированное}} \text{ слово}$$

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

Четвертый пример.

$$Rc : \begin{cases} \#\xi \to \xi\# \\ \# \to \bullet x_0 (x_0 \in V^* \text{ - фиксированное слово }) \\ \to \# \end{cases}$$

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

Пятый пример.

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

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

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

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

Третий тест:

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

Можно строго доказать, что

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

1.4 Эквивалентность нормальных алгоритмов. Теорема о переводе.

Пусть даны $\mathcal{A}, \mathcal{B} : V^* \underset{\bullet}{\to} V^*$ 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.

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

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

То есть

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

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

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

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

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

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

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

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

Получаем:

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

Нам нужно расширить алфавит. Как это делается?
Рассмотрим алфавиты $V = \{a_1, a_2, \ldots, a_n\}, V_\alpha = \{\alpha, \beta\}$ и $V \cap V_\alpha = \varnothing$
Тогда считается

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

Пример.

$$[\underbrace{abca}_{V_0} = \underbrace{010}_{a}\underbrace{0110}_{b}\underbrace{0111}_{c}\underbrace{010}_{a}$$

Примеры использования теоремы сочетания.

1) Проекцирующие НА

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

   Нужен алгоритм, который вычисляет его x_i

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

$$\mathscr{P}_1 : \begin{cases} \$\eta \to \ //\eta \in V \\ \$ \to \\ \to \bullet \end{cases}$$

Результат работы $\mathscr{P}_1(x_1\$x_2\$\ldots\$x_n) = x_1$

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

То есть $\mathscr{P}_2(x_1\$x_2\$\ldots\$x_n) = x_2\$\ldots\$x_n$

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

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

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

2) НА распознавания равенства слов

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

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

$\mathcal{I}nv(y) = y^R$

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

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

3) НА определения центра слова

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

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

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

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

$$\mathscr{A}: \begin{cases} \alpha\beta\xi \to \alpha\beta \\ \xi\alpha\beta \to \alpha\beta \\ \alpha\beta \to \bullet \\ \to \bullet \end{cases}$$

$$\mathscr{B}: \begin{cases} \alpha\beta \to \bullet\$ \\ \to \bullet\$ \end{cases}$$

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

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

Пример 2. $x = a \in V$

$\mathscr{A}(a) = a \ne \lambda$

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

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

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

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

Пример 3. $x = ab$

$\mathscr{A}(ab) = ab \ne \lambda$

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

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

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

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

Пример 4. $x = abcde$

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

1 Итерация:

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

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

2 Итерация:

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

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

3 Итерация:

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

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

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

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

1.5 Универсальный нормальный алгорифм.

Пусть дан НА:

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

$$A^{\text{И}} \leftrightharpoons u_1\alpha[\beta]v_1\gamma u_2\alpha[\beta]v_2\gamma \ldots \gamma u_n\alpha[\beta]v_n, \text{ где } \alpha, \beta, \gamma \notin V$$

Пусть

$$\mathscr{A}_0: \begin{cases} \#a \to a\# \\ \#b \to b\# \\ \# \to \bullet aba \to \# \end{cases}$$

Отсюда

$$A_0^{\text{И}} = \#a\alpha a\#\gamma\#b\alpha b\#\gamma\#\alpha\beta aba\gamma\alpha\#$$

$$\text{Ɛ}A_0\text{З} = \underbrace{01110}_{\#}\underbrace{010}_{a}\underbrace{011110}_{\alpha}\underbrace{010}_{a}\underbrace{01110}_{\#}\underbrace{01111110}_{\gamma}$$

a	b	$\#$	α	β	γ
1	2	3	4	5	6

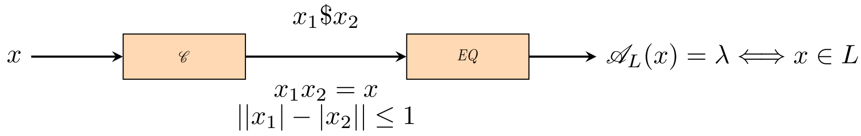
Теорема 1.2. *(Об универсальном НА). Пусть V - произвольный алфавит. Может быть построен НА U над алфавитом $V \cup V_0$ такой, что для любых НА $\mathcal{A}$ в алфавите V и слова $x \in V^*$ имеет место $U(\text{Ɛ}\mathcal{A}\text{З}\$x) \simeq \mathcal{A}(x)$, где $\$ \notin V \cup V_0$*

1.6 Разрешимые и перечислимые языки.

Определение 18. Язык $L \subseteq V^*$ называется алгоритмически разрешимым, если может быть построен НА $\mathscr{A}_L$ над алфавитом V такой, что

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

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



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

$$\underbrace{\mathscr{C} \to EQ}_{\mathscr{A}_L}$$

Определение 19. НА $\widetilde{\mathscr{A}_L}$ называется полуразрешимым для языка $L \subseteq V^*$, если

$$!\widetilde{\mathscr{A}_L}(x) \Longleftrightarrow x \in L$$

Теорема 1.3. *Если для языка L невозможен полуразрешающий НА, то невозможен и разрешающий.*

Доказательство. От противного. Предполагаем, что для языка L невозможен полуразрешающий, то возможен разрешающий НА.

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

По теореме о разветвлении строим

$$\mathscr{B}_L = {}_{\mathscr{A}_L}(\mathscr{A}_L \vee Null),$$

где

$$Null : \begin{cases} \to \end{cases}$$

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

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

Итак, $!\mathscr{B}_L(x) \Longleftrightarrow x \in L$, то есть $\mathscr{B}_L$ - полуразрешающий НА для L вопреки условию теоремы. □

Теорема 1.4. *Если язык L разрешим, то и разрешимо его дополнение.*

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

Для универсального языка:

$$L = V^* \quad \mathscr{A}_{V^*} : \begin{cases} \xi \to \quad //\xi \in V \\ \to \cdot \end{cases}$$

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

Определение 20. Конструктивное натуральное число (КНЧ) - это слово вида $0\underbrace{11\dots1}_{n\geq0}$. Ноль кодирует ноль, 01 кодирует 1 и так далее. КНЧ $x \in V_0^*$
$$0 \to 0; \quad 01 \to 1; \quad 011 \to 2; \quad \dots$$

Определение 21. Конструктивное целое число (КЦЧ) - это слово вида $[-]n$, где n - КНЧ.

Определение 22. Конструктивное рациональное число (КРЧ): m/n, где m, n - КЦЧ, то есть слово в $\{0, 1, -, /\}$ и $n \neq 0$

Определение 23. Язык $L \subseteq V^*$ называется алгорифмически перечислимым, если может быть построен НА N_L такой, что для любого КНЧ n $!N_L(n)$ и $N_L(n) \in L$, и $(\forall x \in L)$ осуществимо КНЧ n такое, что $x = N_L(n)$

Определение 24. $A, \quad \nu : \mathbb{N}_0 \to A$ сюръективно, то есть $(\forall x \in A)(\exists n \in \mathbb{N}_0)(x = \nu(n))$. Это называется нумерацией множества A.

Далее будем предполагать, что отображение ν будет биективной.

Проведем нумерацию целых чисел
рис1
Можно записать в виде формулы:
$$\gamma(n) = \begin{cases} -\frac{n}{2}, \text{ если } n \text{ четное} \\ \frac{n+1}{2}, \text{ если } n \text{ нечетное} \end{cases}$$

Сначала сделаем 3 алгорифма, нужных для следующей задачи (?)
$$\mathscr{C} : \begin{cases} 11 \to \\ 0 \to \cdot \end{cases}$$

Можем заметить, что $\mathscr{C}(n) = \lambda \iff n$ четное
$$N_L = {}_{\mathscr{C}}(\mathscr{A} \vee \mathscr{B})$$

Схема $\mathscr{A}$:
$$\mathscr{A} : \begin{cases} \alpha 11 \to 1\alpha \\ \alpha \to \cdot \\ 01 \to -0\alpha 1 \\ 0 \to \cdot 0 \end{cases}$$

Причем $\alpha \notin V_0$
Схема $\mathscr{B}$
$$\mathscr{B} : \begin{cases} \alpha 11 \to 1\alpha \\ \alpha \to \cdot \\ 01 \to 0\alpha 11 \\ \to \cdot \end{cases}$$

Нужно пронумеровать рациональные числа. Это по факту пары двух целых. Значит, учимся упорядочивать пары.
рис2

Определение 25. Область применимости НА $\mathscr{A}$ относительно алфавита V: пусть $\mathscr{A} = (V' \supset V, S, P)$ - НА над V; Тогда область применимости НА относительно алфавита V есть множество $\mathscr{M}_{\mathscr{A}}^V \leftrightharpoons \{x : x \in V^* \text{ и } !\mathscr{A}(x)\}$, причем $\mathscr{A} : V^* \underset{\bullet}{\to} V^*$. $\mathscr{M}_{\mathscr{A}}^V$ и есть область применимости.

Теорема 1.5. *Язык $L \subseteq V^*$ перечислим тогда и только тогда, когда он является областью применимости относительно алфавита V некоторого НА.*

Следствие. Всякий разрешимый язык перечислим.

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

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

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

Отсюда получаем

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

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

Однако обратное неверно! □

1.7 Проблема применимости нормальных алгорифмов Маркова

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

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

$$!\mathscr{B}(\text{Ɛ}\mathscr{A}\text{З}\$x) \text{ и } \mathscr{B}(\text{Ɛ}\mathscr{A}\text{З}\$x) = \lambda \Longleftrightarrow \neg!\mathscr{A}(x)$$

1.7.1 Проблема самоприменимости.

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

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

$$V \to V \cup V_0$$

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

$$\mathscr{A} : (V \cup V_0)^* \subset \bullet\!\!\to (V \cup V_0)^*$$

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

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

$$!\mathscr{B}(\text{Ɛ}\mathscr{A}\text{З}) \text{ и } \mathscr{B}(\text{Ɛ}\mathscr{A}\text{З}) = \lambda \Longleftrightarrow \neg!\mathscr{A}(\text{Ɛ}\mathscr{A}\text{З})$$

Примеры. Построим как самоприменимые, так и несамоприменимые НА.

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

Дадим ему на вход свою же запись:

$$\mathscr{A}_0 : \text{Ɛ}\mathscr{A}_0\text{З} \vdash \#\text{Ɛ}\mathscr{A}_0\text{З} \vdash \cdot aba\text{Ɛ}\mathscr{A}_0\text{З}$$

Причем $V_0 \cap \{\#, a, b\} = \varnothing$. Этот алгорифм самоприменим.

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

Дадим ему на вход свою же запись:

$$\mathscr{A}_0^f : \text{Ɛ}\mathscr{A}_0^f\text{З} \vdash \text{Ɛ}\mathscr{A}_0^f\text{З} \vdash \dots$$

То есть $\neg !\mathscr{A}_0^f(\text{Ɛ}\mathscr{A}_0^f\text{З})$

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

$$!\mathscr{B}(\text{Ɛ}\mathscr{A}\text{З}) \iff \neg !\mathscr{A}(\text{Ɛ}\mathscr{A}\text{З})$$

Доказательство. Пусть алгорифм $\mathscr{B}$ построен. Тогда при $\mathscr{A} = \mathscr{B}$ имеем:

$$!\mathscr{B}(\text{Ɛ}\mathscr{B}\text{З}) \iff \neg !\mathscr{B}(\text{Ɛ}\mathscr{B}\text{З})$$

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

Теорема 1.6. *Невозможен НА $\mathscr{B}$ над алфавитом V_0 так, что для любого НА $\mathscr{A}$ в алфавите V_1 имело бы место*

$$!\mathscr{B}(\text{Ɛ}\mathscr{A}\text{З}) \iff \neg !\mathscr{A}(\text{Ɛ}\mathscr{A}\text{З})$$

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

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

Пусть

$$!\mathscr{B}(\text{Ɛ}\mathscr{A}\text{З}) \iff \neg !\mathscr{A}(\text{Ɛ}\mathscr{A}\text{З}),$$

но тогда $!\mathscr{B}(\text{Ɛ}\mathscr{A}\text{З}) \iff !\mathscr{B}_1(\text{Ɛ}\mathscr{A}\text{З}) \iff !\mathscr{B}_2(\text{Ɛ}\mathscr{A}\text{З}) \iff \neg !\mathscr{A}(\text{Ɛ}\mathscr{A}\text{З})$, что невозможно в силу самой леммы. □

Итак, мы доказали невозможность полуразрешимость самоприменимости.

Проблема самоприменимости для алгорифмов алгорифмически неразрешима.