Дискретка, Второй семестр.

19 августа 2014 г.

1 Конечные автоматы.

Будем рассматриватьмоднли систем обрабатывающие данные из внешней среды. Эти системы имеют устройства для ввода и вывода, механизм обратотки.

Вход $\longrightarrow$ [Ы] $\longrightarrow$ Выход

Будем рассматриватьмодели, работающие в дескретном времени, тоесть работа выполняется по шагам и они пронумерованы по времени.

t_0- начальный момент времени.

$t_0, t_0 + 1, t_0 + 2 \ldots$

Будем считать что в каждый момент времени устройство находится в одниом из своих внутренних состояний. Функционирование устройства однозначно определяется входными данными и текущим состоянием.

Результатом работы устройства является однозначное определяемое новое состояние (В следующий момент времени) и значением, поступающим на выход устройства.

Определение. Формальное определение автоны функции $g(x_1 \ldots x_n)$ и $h(x_1 \ldots x_n, x_{n+1}, x_{n+2})$

Определение. Функция $f(x_1, \ldots x_n, x_{n+1})$ получается из g и h с помошью операция примитивной рекурсии, если

$f(x_1 \ldots x_n, 0) = g(x_1 \ldots x_n)$

$f(x_1 \ldots x_n, y+1) = h(x_1 \ldots x_n, y, f(x_1 \ldots x_n, y))$

В f :$x_1 \ldots x_n$- параметры, которые в схеме не меняются, x_{n+1} - переменная рекурсии, x_{n+2} - процедура рекурсивногго обращения.

примитивная рекурсия мложелирует оператор арифмитического цикла.

Класс всех числ

Определение. Формальное определение автомата. Конечным автоматом называется пятёрка Ы $= (A, B, Q, \varphi, \psi)$, где A, B, Q - конечные множества, а φ, ψ- отображения: $\varphi : A \times Q \longrightarrow Q$;$\psi : A \times Q \longrightarrow B$ называются функциями перехода и выхода автомата.

A - Входной алфавит.

B - Выходной алфавит.

Q - Множество состояний.

φ- Функция изменения состояния. $(a \in A; q, q' \in Q; \varphi(a, q) = q')$

ψ- Функция вывода.

1.1 Представление автоматов

1.1.1 Табличное задание.

Ы $= (A, B, Q, \varphi, \psi)$

$A = \{a_1 \ldots a_n\}$

$B = \{b_1 \ldots b_m\}$

$Q = \{q_1 \ldots q_r\}$

$$\begin{pmatrix} \varphi & q_1 & \ldots & q_i & \ldots & q_r \\ a_1 & & & \downarrow & & \\ \ldots & & & \downarrow & & \\ a_j & \longrightarrow & \longrightarrow & \sigma_{i,j} & & \\ \ldots & & & & & \\ a_n & & & & & \end{pmatrix}$$

$\sigma_{i,j} = \varphi(a_i, q_j)$

$$\begin{pmatrix} \psi & q_1 & \dots & q_i & \dots & q_r \\ a_1 & & & \downarrow & & \\ \dots & & & \downarrow & & \\ a_j & \longrightarrow & \longrightarrow & \delta_{i,j} & & \\ \dots & & & & & \\ a_n & & & & & \end{pmatrix}$$

$\delta_{i,j} = \psi(a_i, q_j)$

1.1.2 Диаграммы перехода.

Пусть Ы $= (A, B, Q, \varphi, \psi)$

$A = \{a_1 \dots a_n\}$

$B = \{b_1 \dots b_m\}$

$Q = \{q_1 \dots q_r\}$

Будем изображатьсостояние автоматакругами, которым приписаны состояния.Из кружков выходят дуги, где дуга - алфавит $\varphi(a_i, q_j)$. Дополнительно все дуги размечены значения $\psi(a_i, q_j)$, т.е. выходные параметры.

Пример. Газетный автомат.

$3, 5$ сантимов - это монеты. В автомат помещаются монеты,и он способен выдать газету, если в него помещено не менее 10 монет.

$A = \{3, 5, \Lambda, K\}$(Λ - ничего, K - кнопка)

$B = \{\Lambda, \varrho\}$(Λ - ничего, ϱ - газета)

$Q = (0, 3, 6, 8, 9, 10)$

и т.д.

Замечание. Это неполный пример, я пока не знаю, как можно его дополнить.

1.1.3 Каконические уравнения.

Пусть Ы $= (A, B, Q, \varphi, \psi)$; $A = \{a_1 \dots a_n\}$; $B = \{b_1 \dots b_m\}$; $Q = \{q_1 \dots q_r\}$.

t_0 - время состояния $q_0 \in Q$ в этот момент тогда работа Ы во времени представляется следующими отношениями:

$q(t_0) = t_0$

$q(t+1) = \varphi(x(t), q(t))$

$x(t)$- входной символ.

$q(t)$- состояние автомата в момент t.

$y(t) = \psi(x(t), q(t))$

$y(t)$- символ выхода.

1.2 Функции конечных автоматов.

Пусть $A = (a_1 \dots a_n)$, где элементы - символы. Слово в A - всякая конечная последовательность символов. Пустое слово - Λ. Множество слов в A - это A^*.

$\alpha, \beta \in A^*$. $\alpha\beta$ - сцепление(конкютинация) - это слово, получаемое выписыванием сначала α, потом β.

Ы $= (A, B, Q, \varphi, \psi)$ - автомат.

$t_0; q_0 \in Q$

$\alpha = \sigma_1\sigma_2\sigma_3 \dots \sigma_s$ - поступает слово.

$$\left| \begin{array}{cccccccc} & t_0 & t_0+1 & t_0+2 & t_0+3 & \dots & t_0+s-1 & \\ \alpha & \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \dots & \sigma_s & \\ q_i & q_0 & q' & q'' & q''' & \dots & q^{s-1} & q^s s \\ \beta & \delta_1 & \delta_2 & \delta_3 & \delta_4 & \dots & \delta_s & \end{array} \right|$$

$\psi\left(\sigma_i, q^{i-1}\right) = \delta_i$

$\varphi\left(\sigma_j, q^{j-1}\right) = q^j$

Если на вход Ы поступает $\alpha = \sigma_1 \dots \sigma_s$(входное слово), а на выходе выходит $\beta = \delta_1 \dots \delta_s$(выходное слово), то будем говорить, что автомат из q_0 перерабатывает слово α в слово β.

Определение. $f : A^* \longrightarrow B^*$, вычисляемая автоматом Ы из $q_0 \Longleftrightarrow (def)\ \forall \alpha \in A^* \left(f(\alpha) = \beta \longleftrightarrow\right.$ Ыиз q_0перерабатывает αi

1.2.1 Числовые функции, вычисляемые автоматами.

$f : N^k \longrightarrow N$

$N = \{1, 2, 3, 4, 5, 6, 7, 8 \ldots\}$

$f(x_1 \ldots x_n)$

$x_1 \ldots x_n$ - исходные данные ($x_i \in \{0,1\}$), добавлены нули слева для выравнивания их по длинне. Рассматривам $x_1 \ldots x_n$двоичными наборами с любым количеством незначимых нулей. ($\overline{x_1 \ldots x_n}$)- слово, симвалами которого различные наборы длинны n. Соответстыуют 1,2,3.. разрядов $x_1, x_2 \ldots x_n$.

Определение. $f : N^k \longrightarrow N$ вычисляется Ы $= \left(E_2^2, E_2, Q, \varphi, \psi\right)$ из $q_0 \Longleftrightarrow (def)\ \forall x_1 \ldots x_n \in N\,(f(x_1 \ldots x_n) = y \Longleftrightarrow$ Ыиз q_0

Пример. $f(x, y) = x + y$

$$A = \begin{Bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{Bmatrix}$$

$B = \{0, 1\}$

......

Замечание. С примерами у меня как то не пошло, хз как его тут делать.

Определение. Если Ы $= (A, B, Q, \varphi, \psi)$,то словяная функция, вычасляющаяся Ы из q_0 обозначаается как $f_{\text{ы}}^{q_0}$.

1.2.2 Неформулируемость функции умножения в автоматах.

Теорема. *Числовая функция $f(x, y) = x \cdot y$ не вычислима конечными автоматами.*

Доказательство. Предположим обратное. Ы $= \left(E_2^2, E_2, Q, \varphi, \psi\right)$. Из начального состояния $q_0 \in Q$ вычисляет функцию $f(x, y) = x \cdot y$, тоесть $f_{\text{ы}}^{q_0} = x \cdot y$. Пусть $|Q| = r$. Рассмотрим процесс преработки автоматом Ы входных слов, представляющих числа, двоичная запись которых длинны n. Длинна произведения таких чисел может равняться $2 \cdot n$ это так, поскольку самое большое число - $(2 \cdot n - 1)$, а их произведение - $(2^n - 1)^2 = 2^{2n} - 2 \cdot 2^{2n} + 1$. Рассмотрим процесс переработки таких чисел после подачи на вход автомата их старших разрядов. После переработки x и y он находится в первом из своих состояний и продолжает работу, пока на вход не поступит пара нулей. Тоесть заполнение автоматом старших n разрялов произведения зависит лишь в каком он состоянии он начинает формировать значения этих разрядов. $\Longrightarrow$ Разных способов заполненгия старших n разряддов 2^nразрядных произведений по n - разрядным числам неболее чем r и значения старших n разрядов произведения зависит только от того, в каком состоянии автомат начинает формировать значения этих разрядов. Значения старших n разрядов произведения n разрядных чисел представляет собой любые двоичные последовательности длинны n, поэтому их не меньще чем 2^n.$\Longrightarrow$Должно выполняться $r \geq 2^n$, не верно так как r - фиксировано, а n - произвольное.$\Longrightarrow$ Противоречие.$\Longrightarrow$Предположеие не верно. *Чтд.* ☐

1.3 Переработка автоматами неограниченых сверхслов.

Пусть A - конечный алфовит. A^∞ - множество сверхслов в алфовите A (∞ последовательность символов из A).

Определение. Сверхслово α - переодическое $\Longleftrightarrow (def)\ \alpha = \beta(\gamma)^\infty$, где $\beta, \gamma \in A^*$ и $|\gamma| > 0$.

Теорема. *Пусть Ы $= \left(E_2^2, E_2, Q, \varphi, \psi\right)$ - конечный автомат, $q' \in Q$ (q' - начальное состояние),$\alpha \in A^\infty$(α - переодическое сверхслово)$\Longrightarrow$ Ы перерабатывает α в бесконечное переодическое сверхслово.*

Доказательство. $\alpha = \beta(\gamma)^\infty$; $|\gamma| > 0$. Рассмотрим процесс переработки α.Рассмотрим последовательность $q_1, q_2, q_3 \ldots q_i \ldots (1^{**})$. Это состояния автомата в момент переработки символов последовательно идущих слов γ в α . $\left|\begin{matrix} \beta & \gamma & \gamma & \gamma & \ldots & \gamma & \ldots \\ & q_1 & q_2 & q_3 & & q_i & \end{matrix}\right|$ Так как множество состояний Ы конечное, то в послеповательности (1^{**}) состояния повторяются. Пусть $q_s = q_p$ при $s < p$. Это означает что переработка последовательно идущих слов $(\gamma)^{p-s}$ начиная с вхождения γ в α с номера s выполняется одинаково. $\left|\begin{matrix} & 1 & \ldots & s & \ldots & s+p & \ldots \\ \beta & \gamma & \ldots & \gamma & \ldots & \gamma & \ldots \end{matrix}\right|$
Тоесть :$f_{\text{ы}}^{q'}(\alpha)$=$f_{\text{ы}}^{q'}(\beta\gamma^\infty)$=$f_{\text{ы}}^{q'}\left(\beta\gamma^{s-1}\left(\gamma^{p-s}\right)^\infty\right)$ =$f_{\text{ы}}^{q'}\left(\beta\gamma^{s-1}\right)\left(f_{\text{ы}}^{qs}\left(\gamma^{p-s}\right)\right)$ *Чтд.* ☐

1.4 Отличимость состояний конечного автомата.

$\varepsilon = \{(q', q'')\,|q'$и q'' - неотличимы $\}$

Определение. Состояния $q', q'' \in Q; Ы = (A, B, Q, \varphi, \psi)$ - неотличимы $\Longleftrightarrow$ (def) $f_Ы^{q'} \equiv f_Ы^{q''}$ или $\forall\alpha \in A^* \left(f_Ы^{q'}(\alpha) = f_Ы^{q''}(\alpha)\right)$.

Определение. Состояния $q', q'' \in Q; Ы = (A, B, Q, \varphi, \psi)$ - отличимы $\Longleftrightarrow$ (def) $f_Ы^{q'} \neq f_Ы^{q''}$ или $\exists\alpha \in A^* \left(f_Ы^{q'}(\alpha) \neq f_Ы^{q''}(\alpha)\right)$.

Распознавания отличимости или неотличимости состояния автомана по определению не возможно.
Рассмотрим задачу распознавания отличимых и неотличимых состояний автомата.

Определение. Состояния $q', q'' \in Q; Ы = (A, B, Q, \varphi, \psi)$ называются K - неотличимыми если $\forall\alpha \in A^* \Big(|\alpha| \leq K \to f_Ы^{q'}(\alpha) =$

На множестве Q определим отногшение K - неотличимости.
$q_1, q_2, q_3 \dots q_i$, где $q_i = \{(q', q'')\,|q'$ и $q'' - i -$ неотличимы$\}$
Добавим к этим отношениям отношение неотличимости.
$\varepsilon = \{(q', q'')\,|q'$ и $q'' -$ неотличимы$\}$

Теорема. *Каждое отношение $q_1, q_2 \dots \varepsilon$ - отношения эквивалентности на Q.*

Доказательство. Проверим рефлексивность, симмертичность и транзитивность.
1) *Рефлексивность.*
$\forall q \in Q\,(q\rho q)$
$|\alpha| \leq i \longrightarrow f_Ы^q = f_Ы^q$
Рефлексивность + .
2) *Симметричнось.*
$\forall q', q''\,(q'\rho q'' \longrightarrow q''\rho q')$
$\forall\alpha \in A^* \left(|\alpha| < i \longrightarrow f_Ы^{q'}(\alpha) = f_Ы^{q''}(\alpha)\right) \Longrightarrow \forall\alpha \in A^* \left(|\alpha| < i \longrightarrow f_Ы^{q''}(\alpha) = f_Ы^{q'}(\alpha)\right)$
Симметричность допишу мб, а пока я хз ($\pm$)
3) *Транзитивность.*
$\forall q', q'', q'''\,(q'\rho q''\&q''\rho q''' \longrightarrow q'\rho q''')$
.....
В общем чтд. ☐

Лемма. $q_1 \supseteq q_2 \supseteq q_3 \supseteq \dots \supseteq \varepsilon$

Доказательство. Разделим доказательство на два пункта.
1) *Докажем что* $q_i \supseteq q_{i+1}$ *при* $i = 1, 2, 3 \dots n$
Пусть $(q', q'') \in \rho_{i+1} \Longrightarrow \forall\alpha \in A^* \left(|\alpha| \leq i + 1 \longrightarrow f_Ы^{q'}(\alpha) = f_Ы^{q''}(\alpha)\right) \Longrightarrow \forall\alpha \in A^* \left(|\alpha| \leq i \longrightarrow f_Ы^{q'}(\alpha) = f_Ы^{q''}(\alpha)\right) \Longrightarrow (q', q'') \in \rho_i$
2) *Покажем что*$\rho_i \supseteq \varepsilon$.
Пусть $q', q" \in \varepsilon \Longrightarrow \forall\alpha \in A^* \left(f_Ы^{q'}(\alpha) = f_Ы^{q''}(\alpha)\right) \Longrightarrow \forall\alpha \in A^* \left(|\alpha| \leq i \longrightarrow f_Ы^{q'}(\alpha) = f_Ы^{q''}(\alpha)\right) \Longrightarrow (q', q'') \in \rho_i$
Чтд. ☐

Лемма. *Если* $\rho_i = \rho_{i+1} \Longrightarrow$
1)$\forall j\,(\rho_i = \rho_{i+j})$
2)$\rho_i = \varepsilon$

Доказательство. 1)$\rho_i = \rho_{i+1} \Rightarrow \rho_i = \rho_{i+2}$
$\rho_i \supseteq \rho_{i+2}$ и $\rho_{i+2} \supseteq \rho_i$
$(q', q'') \in \rho_i \Rightarrow (q', q'') \in \rho_{i+2}$
$\alpha \in A^*$; $|\alpha| = i + 2$; $\alpha = a \cdot \beta$; $|\beta| = i + 1$
$f_Ы^{q'}(\alpha) = f_Ы^{q'}(a\beta) = f_Ы^{q'}(a) * f_Ы^{\varphi(a,q')}(\beta)$
$f_Ы^{q''}(\alpha) = f_Ы^{q''}(a\beta) = f_Ы^{q''}(a) * f_Ы^{\varphi(a,q'')}(\beta)$
$f_Ы^{q''}(\alpha) = f_Ы^{q'}(\alpha) \Rightarrow f_Ы^{\varphi(a,q')}(\beta) = f_Ы^{\varphi(a,q'')}(\beta)$
$(\varphi(a, q'), \varphi(a, q'')) \in \rho_i \Longrightarrow (\varphi(a, q'), \varphi(a, q'')) \in \rho_{i+1}$

2)
$\rho_i = \rho_{i+k}$; $k > 0$
$\rho_i = \rho_{i+1} = \rho_{i+2} \Longrightarrow \rho_{i+1} = \rho_{i+3} = \ldots$
Покажем что $\rho_i = \varepsilon$
Пусть $\alpha \in A^*$, если$|\alpha| \leq i \Longrightarrow f^{q'}(\alpha) = f^{q''}(\alpha)$
$|\alpha| > i \Longrightarrow f^{q'}(\alpha) = f^{q''}(\alpha)$так как $(q', q'') \in \rho_{i+k} \Longrightarrow \forall\alpha \in A* \left(f^{q'}(\alpha) = f^{q''}(\alpha)\right) \Longrightarrow (q', q'') \in \varepsilon$ □

Лемма. *(3) Разбиение Q пораждаемое ρ_{i+1} является подразбиением разбиения Q пораждаемое ρ_i.*

Доказательство. Если $(q', q'') \in \rho_{i+1} \Longrightarrow (q', q'') \in \rho_{i+1}$

Тоесть всякий класс $i+1$ неотличимых состояний Поэтому всякий класс неотличимых состояний является частью класса i неотличимых состояний. □

Теорема. *Пусть q' и q'' - отличимые состояния автомата $Ы = (A, B, Q, \varphi, \psi)$ тогда $\exists\alpha \in A^* \left((|\alpha| \leq |Q| - 1) \& \left(f^{q'}(\alpha) \neq\right.\right.$*

Доказательство. Для автомата Ы рассмотрим последповательность отношений i неотличимсти состояний.

$q_1 \supset q_2 \supset q_3 \supset \ldots q_i = q_{i+1} = \ldots$

То ρ_1 разбивает множество Q .

Это так поскольку если ρ_1 разбивает Q на одно множество, то все состояния автомата Ы являются 1-неотличимыми, что вличет неотличимость любых 2х состояний автомата Ы.

Тощда число множеств разбиения не меньше чем 3.

А кол во множеств разбиения

Максималдьное кол во множеств разбиения Q $= |Q| \Longrightarrow i + q \leq |Q| \Longrightarrow i \leq |Q| - 1$

Тоесть q' и q" -отличимые состояния автомата Ы должны различаться хотябы на одном слове, длинннна которого не превосходит$|Q| - 1$, иначе в противнгом случае они оказываются неотличтимыми.

Замечание. Доказанная теорема означает, что если Ы имеет N состояний, то отличимость любых 2х его состояний можно проверить если рассмотреть как автомат из этих состояний перерабатывает входные слова длинны не меньше n-1 , если они отличимы, то это будет распознано, а если неотличимы, то не других словах неотличимость проверять нет смысла по доказанной теореме.

Оченка мощность $|Q| - 1$ длинны проверяемых слов является точной, тоесть сущ автоматы для которых отличимость некоторых состояний требеет проверки отличимости некоторых состояний. □

1.5 Минимальный автомат.

Нерегулярность структуры венника.

Определение. Автоматы $Ы_1 = (A, B, Q_1, \varphi_1, \psi_1)$ и $Ы_2 = (A, B, Q_2, \varphi_2, \psi_2)$ называются эквиввалентными если $\forall q' \in Q_1 \exists q''; \in Q_2 \left(f^{q'}_{Ы_1} \equiv f^{q''}_{Ы_2}\right) \& \forall q'' \in Q_2 \exists q'; \in Q_1 \left(f^{q'}_{Ы_1} \equiv f^{q''}_{Ы_2}\right)$

Определение. Автомат Ы-минимальный-если любые 2 его состояний неотличимы.

Лемма. *О минимальном автомате.*

Для каждого автомата Ы существует эквивалентный ему минимальный автомат Т

Доказательство. Пусть сущ автомат Ы ,если он минимальный то теорема доказана))

$Ы = (A, B, Q, \varphi, \psi)$

$Q_1, \ldots Q_d$- классы неотличимых состояний. $Q = \{q_1, \ldots q_m\}$

$Q^* = \left\{q^1, \ldots q^d\right\}$

Определим $\mathrm{T} = (A, B, Q^*, \Phi, \Psi)$

$\xi : Q \longrightarrow \{1, \ldots d\}$

Данная фуннкция переводит состояния Ы классы неотличимых состояний этого автомата.

$\eta : \{1, \ldots d\} \longrightarrow \{1, \ldots m\}$

$\forall i = \overline{1, d}\ \ (\eta(i) = min(t))$; $q_t \in Q_i$

$\Phi : A \times Q^* \longrightarrow Q^*$

$\Phi\left(a, q^i\right) = q^{\xi\left(\varphi\left(a, q_{\eta(i)}\right)\right)}$

$\Psi\left(a, q^i\right) = \Psi\left(a, q_{\eta(i)}\right)$

Доказательство минималности Т и его эквивалентности автомату Ы.

1. Пусть $q^i \in Q^*$покажем $f_\tau^{q^i} \equiv f_{\text{Ы}}^{q_{\eta(i)}}$

   Покажем $f_T^{q^i}(\alpha) = f_{\text{ы}}^{q_{\eta(i)}}(\alpha)$индукцией по длинне слов $\alpha \in A^*$

   Для таких слов одновременно будем доказывать еще 1нг утверждение

   $\forall \alpha \in A^* \left(\Phi\left(\alpha, q^i\right) = q^s\right)$ токого что $q_{\eta(s)}$и $\varphi\left(\alpha, q_{\eta(i)}\right)$- неотличимы.

   Базис индукции

   $|\alpha| = 1$

   $f_T^{q^i}(a) = \Psi\left(a, q^i\right) = \Psi\left(a, q_{\eta(i)}\right) = f_{\text{Ы}}^{q_{\eta(i)}}(a)$

   $q^s = \Phi\left(a, q^i\right) = q^{\xi\left(\varphi\left(a, q_{\eta(i)}\right)\right)}$

   $\xi\ \varphi\left(a, q_{\eta(i)}\right) \in Q_s$тогда $q_{\eta(s)}$и $\varphi\left(\alpha, q_{\eta(i)}\right)$- неотличимы.

   $Q_s\ q_{\eta(s)} \in Q_s$

   Индуктивное предположение

   $|\alpha| = k$

   $\forall \alpha \in A^* \left(|\alpha| = k \longrightarrow f_T^{q^i}(\alpha) = f_{\text{ы}}^{q_{\eta(i)}}(\alpha)\right)$

   Индуктивный переход

   $|\alpha| = k + 1$

   $\alpha = \beta \cdot a$

   $|\beta| = k$

   $f_T^{q^i}(\alpha) = f_T^{q^i}(\beta) \cdot f_T^{q^s}(a)\ ;\ \left(q^s = \Phi\left(\beta, q^i\right)\right)$

   $f_{\text{ы}}^{q_{\eta(i)}}(\alpha) = f_{\text{ы}}^{\eta(i)}(\beta) \cdot f_{\text{ы}}^{\varphi\left(\beta, q_{\eta(i)}\right)}(a) = \Psi\left(a, q_{\eta(s)}\right) = \Psi\left(a, \varphi\left(\beta, q_{\eta(i)}\right)\right) \Longrightarrow$

   $\Psi\left(a, \varphi\left(\beta, q_{\eta(i)}\right)\right) = f_{\text{ы}}^{\varphi\left(\beta, q_{\eta(i)}\right)}(a)$

2. Докажем второе индуктивное утверждеие

   $q_{\eta(s)}$ и $\varphi\left(\beta, q_{\eta(s)}\right)$- неотличимы.

   $\varphi\left(a, q_{\eta(s)}\right)$ и $\varphi\left(\alpha, q_{\eta(i)}\right)$-тоже неотлшичимы

   Покажем что Т эквивалентно Ы.

   $\forall q^i \in Q^* \exists q_j \in Q \left(f_T^{q^i} \equiv f_{\text{ы}}^{q_j}\right)$

   Покажем что $\forall q_j \in Q \exists q^i \in Q^* \left(f_T^{q^i} \equiv f_{\text{ы}}^{q_j}\right)$

   Пусть $q_j \in Q \& q_i \in Q_k \Longrightarrow f_{\text{ы}}^{q_j} \equiv f_{\text{ы}}^{q_{\nu(k)}} \equiv f_T^{q^k}$

   Покажем что Т - минимальный.

   $q^i, q^j \in Q^* \&\ i \neq j$

   $$\begin{cases} f_T^{q^i} \equiv f_{\text{ы}}^{q_{\nu(i)}} \\ f_T^{q^j} \equiv f_{\text{ы}}^{q_{\nu(j)}} \end{cases} \Longrightarrow f_T^{q^i} \neq f_T^{q^j}$$

□

1.6 Распознование слов автоматами.

Пусть Ы $= (A, B, Q, \varphi, \psi)$ - произвольный автомат.

$q_0 \in Q$ - начальное состояние.

$D \subseteq Q$ - множество распознающих состояний.

Определение. $\alpha \in A^*$ Ы q_0 распознаются D $\overset{def}{\Longleftrightarrow} \varphi(\alpha, q_0) \in D$

Для пресставления автоматов распознающих слова, применяется специальная форма обозначения Ы $= (A, Q, \varphi, q_0, D)$

$U \subseteq A^*$- образует автоматный язык (Множество слов распознаваемых автоматом) для Ы $= (A, Q, \varphi, q_0, D)$ если, $\forall \alpha \in A^* \left(\alpha \in U \longleftrightarrow \varphi\left(\alpha, q_0\right) \in D\right)$

Пример. Пусть $A = \{S, O\}$, построем автомат распознабюший $\alpha = \alpha' sos \alpha''$

Теорема. *Ы*$_1 = \left(A, Q_1, \varphi_1, q_0^1, D_1\right)$
Ы$_2 = \left(A, Q_2, \varphi_2, q_0^2, D_2\right)$
$U_1 \, и \, U_2$
Ы$_3 = \left(A, Q_3, \varphi_3, q_0^3, D_3\right)$
$U_1 \cup U_2 \left(U_1 \cap U_2, U_1 \setminus U_2\right)$

Доказательство. Определим автомат Ы$_3$ так чтобы он моделировал работу автоматов Ы$_2$ и Ы$_1$ отновременно и независимо.
Ы$_3 = \left(A, Q_1 \times Q_2, \varphi_3, \left(q_0^2, q_0^1\right), D_3\right)$
$\varphi_3 : A \times Q_1 \times Q_2 \longrightarrow Q_1 \times Q_2$
$\varphi_3\left(a, q', q''\right) = \left(\varphi_1\left(a, q'\right), \varphi_2\left(a, q''\right)\right)$
Если Ы$_3$ должен распознавать $U_1 \cup U_2$ то в качестве D_3 возбмем множество $D_3 = D_1 \times Q_2 \cup Q_1 \times D_2$.
Покажем, что для так выбранного множества D_3- множество $U_3 = U_1 \cup U_2$.

1. Покажем, что $U_3 \subseteq U_1 \cup U_2$

   Пусть $\alpha \in U_3 \Longrightarrow \varphi_3\left(\alpha, \left(q_0^1, q_0^2\right)\right) \in D_3 \Longrightarrow \left(q', q''\right) \in D_1 \times Q_2 \cup D_2 \times Q_1 \Longrightarrow q' \in D_1 \vee q'' \in D_2 \Longrightarrow \alpha \in U_1 \vee \alpha \in U_2 \Longrightarrow \alpha \in U_1 \cup U_2 \; \left(\left(q', q''\right) = \left(q_0^1, q_0^2\right)\right)$

   Пусть $\alpha \in U_1 \cup U_2 \Longrightarrow \varphi_1\left(\alpha, q_0^1\right) \in D_1 \vee \varphi_2\left(\alpha, q_0^2\right) \in D_2 \Longrightarrow \left(\varphi_1\left(\alpha, q_0^1\right), \varphi_2\left(\alpha, q_0^2\right)\right) \in Q_1 \times D_2 \Longrightarrow \left(\varphi_1\left(\alpha, q_0^1\right), \varphi_2\left(\alpha, q_0^2\right)\right) \in D_1 \times Q_1 \Longrightarrow \alpha \in U_3$

2. Рассмотреть самостоятельно.

3. Рассмотреть самостоятельно.

□

1.7 Операции над автоматами.

1.7.1 Операции суперпозиции автоматов.

Пусть Ы$_1 = (A, B, Q_1, \varphi_1, \psi_1)$ - автомат, $q_0^1 \in Q_1$ - начальное состояние и Ы$_2 = (B, C, Q_2, \varphi_2, \psi_2)$ - автомат, $q_0^2 \in Q_2$ - начальное состояние, рассмотрим Ы$_3 = (A, C, Q_3, \varphi_3, \psi_3)$. Ы$_3$ - это:

$$\longrightarrow \left[\longrightarrow [?]_{\text{Ы}_1} \longrightarrow [?]_{\text{Ы}_2} \longrightarrow\right]_{\text{Ы}_3} \longrightarrow$$

$Q_3 = Q_1 \times Q_2$
$\varphi_3 : A \times Q_1 \times Q_2 \longrightarrow Q_1 \times Q_2$
$\psi_3 : A \times Q_1 \times Q_2 \longrightarrow C$
Ы$_3$ можно задасть с помощью следующих канонических уравнений:
$q_1\left(t_0\right) = q_0^1$
$q_2\left(t_0\right) = q_0^2$
$q\left(t_0\right) = \left(q_0^1, q_0^2\right)$
$q_1\left(t+1\right) = \varphi_1\left(x\left(t\right), q_1\left(t\right)\right)$
$q_2\left(t+1\right) = \varphi_2\left(\psi_1\left(x\left(t\right), q_1\left(t\right)\right), q_2\left(t\right)\right)$
$y\left(t\right) = \psi_2\left(\psi_1\left(x\left(t\right), q_1\left(t\right)\right), q_2\left(t\right)\right)$

1.7.2 Операции обратной связи.

Ы $= \left(A^2, A^2, Q, \varphi, \psi\right) \left\{A^2 = A \times A\right\}$

q_0- начальное состояние Ы.

З $= (A, A, A, \varphi_З . \psi_З)$

Если $A = \{a_1, \ldots a_n\}$, $q_0^З = a_1$, автомат задержки задается так:

$$\begin{cases} q\left(t_0\right) = a_1 \\ q\left(t+1\right) = x\left(t\right) \\ y\left(t\right) = q\left(t\right) \end{cases}$$

Автомат задержки реализует такое преобразование входных последовательностей символов, в котором такие символы появляются во входной последовательности, с задержкой на один шаг.

Рассмотрим авомат Ы$'$.

Замечание. Тут должна быть картинка, которую сделает Жорик.

Ы$'$ - Применение к Ы операцуии обратной связи.

Ы:

$q\left(t_0\right) = q_0$

$q\left(t+!\right) = \varphi\left(x_1\left(t\right), x_2\left(t\right), q\left(t\right)\right)$

$$\begin{cases} q_1\left(t\right) = \psi_1\left(x_1\left(t\right), x_2\left(t\right), q\left(t\right)\right) \\ q_2\left(t\right) = \psi_2\left(x_1\left(t\right), x_2\left(t\right), q\left(t\right)\right) \end{cases}$$

Ы$' = (A, A, Q \times A, \varphi', \psi')$

Каноничекские кравнения для Ы':

$$\left(q_0, a_1\right) \begin{cases} q_0\left(t_0\right) = q_0 \\ q_2\left(t_0\right) = a_1 \end{cases}$$

$q_1\left(t+1\right) = \varphi\left(x\left(t\right), q_2\left(t\right), q_1\left(t\right)\right)$

$q_2\left(t+1\right) = \psi_2\left(x\left(t\right), q_2\left(t\right), q_1\left(t\right)\right)$

$y\left(t\right) = \psi_1(x\left(t\right), q_2\left(t\right), q_1\left(t\right))$

1.8 Автоматные схемы. (23.3.2013)

Рассмотрим 4 автомата: &,V,,З.

$\left(\{0,1\}, \{0,1\}, \{q_0\}, \varphi_{\&}, \psi_{\&}\right)$

$\psi\left(\left(x_1\left(t\right), x_2\left(t\right)\right), g\left(t\right)\right) = \psi\left(\ldots\right)$

Из этих элементов состовляются автоматныне схемы по правилам:

1. Некоторое кол-во входов $x_1, \ldots x_p$
2. На каждый вход каждого элемента подается ровно один вход схемы или ровно 1 выход элеменат.
3. Выходы системы могут расщепляться.
4. Выходы элементов , не подсоеденненные к входам других элементов - выход схемы, они наменуются $y_1, \ldots y_s$
5. Автоматная схема может содержать цыклы, каждый из которых проходит хотябы через 1 элемент задержки.

1.9 Моделирование произвольных автоматов автоматными схемами.

Ы $= (A, B, Q, \varphi, \psi)$

$A = \{a_1 \ldots a_n\}$

$B = \{b_1 \ldots b_m\}$

$Q = \{q_0, q_1 \ldots q_r\}$

$\longrightarrow$ [Ы] $\longrightarrow$

$$\begin{matrix} x_1 & \longrightarrow \\ x_p & \longrightarrow \end{matrix} \left[\quad \text{Ы}' \quad \right] \begin{matrix} \longrightarrow & y_1 \\ \longrightarrow & y_s \end{matrix}$$

Символы входного авфавита автомата Ы будем кодировать 2чными наборами $]\log_2 n[$ - P.

Почему столько? - минимальная длинна 2чного набара, которых не меньше чем n.

Аналогично быдем представлять символы вывохного алфавита и множество состояний автомата Ы.

$]\log_2 m[= \ S$

$]\log_2 (r+!)\,[= \ d$

Иньективно сопоставим 2чные наборы длинны S,p,d их A,B,Q.

Рассмотрим каноническое уравнение для Ы.

$q(t_0) = q_0$

$q(t+1) = \varphi(x(t), q(t))$

$y(t) = \psi(x(t), q(t))$

Будем представлять входные символы Ы: $(x_1(t), \ldots, x_p(t))$

Будем представлятьвыходные символы Ы: $(y_1(t), \ldots, y_p(t))$

Будем представлять множество состояний Ы: $(q_1(t), \ldots, q_d(t))$

Тогда конон. Ы:

$\begin{cases} q_1(t_0) = 0 \\ q_d(t_0) = 0 \end{cases}$

$q_1(t+1) = \varphi_1(x_1(t), \ldots, x_p(t), q_1(t), \ldots, q_d(t))$

$\ldots$

$q_d(t+1) = \varphi_d(x_1(t), \ldots, x_p(t), q_1(t), \ldots, q_d(t))$

$y_1(t) = \psi_1(x_1(t), \ldots, x_p(t), q_1(t), \ldots, q_d(t))$

$\ldots$

$y_s(t) = \psi_s(x_1(t), \ldots, x_p(t), q_1(t), \ldots, q_d(t))$

$\varphi_1 \cdots \varphi_p, \psi_1 \cdots \psi_s$ - функции алгебралогики. Их можно отределить по автомату Ы и выбранной схеме кодирования входный, выходный символов и состояний.

$\varphi_1(x_1(t), \ldots, x_p(t), q_1(t), \ldots, q_d(t))$- как посчитать.

$\sigma_1 \ldots \sigma_p \ a \in A \ ; \ \delta_1 \ldots \delta_s \ q \in Q$

$\varphi(a, q) = q'$

$(\lambda_1 \ldots \lambda_d)$

Положим значение $\varphi_1(\sigma_1 \ldots \sigma_p, \delta_1 \ldots \delta_s) = \lambda_1$.

Тоесть для состояний Ы' кодирующих входные символы Ы, работает с точностью до кодирования элементов A,B,Q двоичными наборами.

Поскольку функции $\varphi_1 \ldots \varphi_d$ и $\psi_1 \ldots \psi_s$ - функции алгеьралогики , то тогда существует функции алгабралогики, вычисляющие эти функции.

Замечание. Тут схема. (Обращаться к Жоре)

Построенная автоматная схема моделирует работу Ы'. Это схема работает так же как и автомат Ы, тоесть если внекоторой момент времени на в...... состояния элемента задержки схемы состовляют код q, если при этом a вырабатывает выходной символ b....... а новое состояние элемента задержки состовляет код q'.

2 Рекурсивыные функция.

Будем рассматривать $f : N^K \longrightarrow N$, где $N = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \ldots\}$

2.1 Определим класс простецйших функций.

$o(x) = 0$ - нулевой.

$s(x) = x + 1$ - следование.

$I_n^m(x_1 \ldots x_n) = x^m \ ; \ (1 \leq m \leq n)$ - проэкция.

Всякая простейшая функция является интуитивно вычислиной.

2.2 Элеменарные функции.

Определение. Пусть заданы функции

$f(x_1 \ldots x_n)$

$g_1(x_{11} \ldots x_{1m_1})$

$\ldots$

$g_n(x_{n1} \ldots x_{nm_n})$

$h(x_1 \dots x_k) = f(g_1(x_{11} \dots x_{1m_1}), \dots, g_n(x_{n1} \dots x_{nm_n}))$
$x_1 \dots x_k$ - не дублируются.

Суперпозицией функций $g_1 \dots g_n$ это функция h.

Множество всех функций, получаемых из простейших функций с помощью конечного числа применений операций суперпозиции называется классом элементарных функций.

Класс элементарной фукнций состовляю функции существенно зависящие от не более 1ой переменной.

Всякая элементарная функция эвляется интуитивно вычислимой.

Для этого достаточно проверить, что $g_1 \dots g_n$ - вычислимые, то их сумеппозиция h тоже вычислима.

2.3 Примитивно-рекурсивные функции.

Определение. Пусть заданы функции $g(x_1 \dots x_n)$ и $h(x_1 \dots x_n, x_{n+1}, x_{n+2})$

Функция $f(x_1, \dots x_n, x_{n+1})$ получается из g и h с помошью операция примитивной рекурсии, если
$f(x_1 \dots x_n, 0) = g(x_1 \dots x_n)$
$f(x_1 \dots x_n, y+1) = h(x_1 \dots x_n, y, f(x_1 \dots x_n, y))$

В f :$x_1 \dots x_n$- параметры, которые в схеме не меняются, x_{n+1} - переменная рекурсии, x_{n+2} - процедура рекурсивногго обращения.

Примитивная рекурсия функция моделирует оператор арифмитического цикла.

Класс всех числдовых функций, которые могут быть получнеы из просетйших с помошью конечного числа приметений операции суперпозиции и припмитивной рекурсии называется классом примитиыно рекурсивных функцый.

2.4 Рекурсивные фукнции.

Будем рассматривать $f : N^k \longrightarrow N$, где $N = \{0, 1, 2, 3, 4 \dots\}$

1. Определим класс простейших функций.

   $$\left.\begin{array}{c} O(x) = 0 \\ S(x) = x+1 \\ I_n^m(x_1, \dots x_n) = x_m \end{array}\right\} \begin{array}{c} -\text{ нулем} \\ -\text{ следования} \\ -\text{ проэкцией} \end{array}$$

   Всякая простейшая функция является интуитивно вычислимой.

2. Элементарные функции

   Определение. Пусть заданы функции $f(x_1 \dots x_n)$, $g_1(x_{11} \dots x_{1m_1})$ $\dots$ $g_n(x_{n1} \dots x_{nm_n})$. Суперпозиция $f, g_1 \dots g_n$ - это $\varphi(x_1 \dots x_k) = f(g_1 \dots g_n)$, где $x_1 \dots x_k$ - все различные символы переменных функций $g_1 \dots g_n$.

   Множество всех функций, получаемых из простейших с помощью конечного числа применения суперпозиции называется классом элементарных функций.

   Пример. $S(S(x)) = x + 2$

   Класс элементарных функций зависит от не более 1ой переменной. Всякая элементарная функция является интуитивно вычислимой, достаточно проверить, что $f, g_1 \dots g_n$ - вычислимые, то их суперпозиция тоже будет вычислима.

3. Приметивно рекурсивные фукнции.

   Определение. Пусть задана $g(x_1 \dots x_n)$, $h(x_1 \dots x_n, x_{n+1}, x_{n+2})$. Функция $f(x_1 \dots x_{n+1})$ получается из g и h с помощью операции приметивной рекурсии если: $f(x_1 \dots x_n, 0) = g(x_1 \dots x_n)$ и $f(x_1 \dots x_n, y+1) = h(x_1 \dots x_n, y, f(x_1 \dots x_n, y))$. ($x_1 \dots x_n$ - ппараметры функции, x_{n+1} - переменная рекурсии, x_{n+2} - переменная рекурсивного вызова)

   Приметивная рекурсия моделирует цикл. Класс всех числовых функций, которые могут быть получены из простых функций с помощью операции суперпозиции и приметивной ркурсии называются классом приметивно - рекурсивных функций.

Пример. $f(x,y) = x + y$

$f(x,0) = I_1^1(x)$

$f(x,y+1) = S(f(x,y))$

Всякая приметивно рекурсивная функция является интуитивно вычислимой.

2.5 Частично рекурсивные функции. (ЧРФ)

Пусть $g(x_1 \ldots x_n, x_{n+1})$ где $x_1 \ldots x_n$ - пареметры функции, x_{n+1} - переменная минимизации.
$f(x_1 \ldots x_n, x_{n+1})$ получается из g с помощью операции минимизации если: $f(x_1 \ldots x_n, x_{n+1}) = \mu t\,(g(x_1 \ldots x_n, t) = x_{n+1})$, где:

1. В качестве значения t, $min \;-\; t$ для которого $g(x_1 \ldots x_n, t) = x_{n+1}$.
2. Значение $\left.\begin{array}{c} g(x_1 \ldots x_n, 0) \\ \ldots \\ g(x_1 \ldots x_n, t) \end{array}\right\}$ - определены.

Если g является интуитивно вычислимой, то f получается из g с помощью операции минимизации тоже вычислима.

Пример. $\varphi(x,y) = x + y$
$f(x,y) = \mu t\,(g(x,t) = y)$
$f = \begin{cases} y - x, & y \geq x \\ -, & \text{иначе} \end{cases}$
$g(x)$
$f(x) = \mu t\,(g(t) = x)$ - является всюдуопределенной, то и инбенктивной.
$f(x) = g^{-1}(x)$

Класс всех числовых функций, которые подучаются из простейших с помощью конечного применения операции минимизации, примитиыной рекурсии и суперпозиции - это класс частично рекурсивных функций(ЧРФ). ЧРФ - интуитивно понятная функция.

2.6 Тезис Черча (Cherch)

Утверждение. Класс интуитивно вычислимых числовых функций сопподает с классом частично рекурсивных функций.

Основные доводы в пользу справедливости тезиса Черча:

1. Любые рассматриваемые вычислимые функции имеют рекурсивные функции.
2. Примеры других формализаций понятия алгоритма пораждают классы вычислимых числовых функций, которые не шире ЧРФ.

2.7 Представление и определение ЧРФ нагруженнуженными деревьями.

Пусть А - алфавит, который составляют следующие символы.
$A = \{O, S, I, C, P, M, 1\}$.
Будем рассмтривать опред. ЧРФ с помощь. нагруженных деверьев, верщины которых размечены одним из символов А.

1. Представление простейших функций.

   $O(x) : O\bullet - - - \bullet < i >$ где $< i >= 1 \ldots 1$ - i раз.

   $S(x) : S\ \bullet - - - \bullet < i >$

   $I_n^m(x_{i_1} \ldots x_{i_n}) : I\ \bullet \quad \begin{array}{l} \nearrow\bullet\ < i_n > \\ \ \nearrow\bullet\ \quad \ldots \\ -\bullet\ \ < i_1 > \\ \searrow\bullet < m > \\ \searrow\bullet < n > \end{array}$

2. Представление операции суперпозиции.

   $h(x_1 \ldots x_k) = f(g_1(x_{11} \ldots x_{1m_1}) \ldots g_n(x_{n1} \ldots x_{nm_n}))$

   $$C \;\bullet\; \begin{array}{l} \nearrow \bullet D_f \\ \nearrow \bullet D_{g_1} \\ \searrow \bullet \ldots \\ \searrow \bullet D_{g_n} \end{array}$$

3. Представление операции приметивной рекурсии.

   $f(x_1 \ldots x_n, 0) = g(x_1 \ldots x_n)$

   $f(x_1 \ldots x_n, y+1) = h(x_1 \ldots x_n, y, f(x_1 \ldots x_n, y))$, то

   $$P \bullet \;\begin{array}{l} \nearrow \bullet Dg \\ \searrow \bullet Dh \end{array}$$

4. Моделирование минимизации.

   $f(x_1 \ldots x_n, x_{n+1}) = \mu t\,(g(x_1 \ldots x_n, t) = x_{n+1})$

   $M\bullet - \bullet Dg$

Замечание. Количество потомков каждой вершины дерева представляющего заданный ЧРФ полнстью определена разметкой вершины и в некоторых случаях поддеревом для левогог потомка.

Пример. $f(x, y) = x + y$

$f(x, 0) = I_1^1(x)$

$f(x, y+1) = S\left(I_3^3(x, y, f(x, y))\right)$

$$P \bullet \left| \begin{array}{l} \nearrow \bullet C \left| \begin{array}{l} \nearrow \bullet S \quad - \quad \bullet \langle 1 \rangle \\ \searrow \bullet I \left| \begin{array}{l} \nearrow \bullet \langle 111 \rangle \\ \nearrow \bullet \langle 11 \rangle \\ - \bullet \langle 1 \rangle \\ \searrow \bullet \langle 111 \rangle \\ \searrow \bullet \langle 111 \rangle \end{array} \right. \end{array} \right. \\ \searrow \bullet I \left| \begin{array}{l} \nearrow \bullet \langle 1 \rangle \\ - \bullet \langle 1 \rangle \\ \searrow \bullet \langle 1 \rangle \end{array} \right. \end{array} \right.$$

2.8 Нумерация ЧРФ

1. $\alpha_0, \alpha_1, \ldots$

   Рассмотрим $A = \{S, O, I, C, P, M, 1, \$\}$ последовательность , в которой расположены по возрастанию длинн, а слова равной длинны по алфавиту.

   Существует алгоритм, который по номеру i слова в последовательности 1 строит αi, тоесть существует алгоритм генерирующий последовательность 1, как последовательность $\alpha 1 \ldots \alpha i \ldots$.

2. $\beta_0, \beta_1, \ldots$

   Для распознавания того, является ли произволное взятое слово проследовательности 1 кодом дерева определения рекурсивной функции можно воспользоваться следующей процедурой:

   (a) Если 1й символ слова S (Конечная функция) то дальше в слове должна следовать последовательность едениц. $[S\$11 \ldots 1]$

   (b) O - аналогияно

   (c) Если начинается с символа I то распознается существование цепочки едениц непоследовательно следующие за I. Считанное количество еденгиц - количествово переменных в фугкции, в этом случае нужно проверить, что хвост состоит из $n+1$ группы едениц, разделенный маркером \$. $[I\$1 \ldots 1\$, \quad]$

   (d) C : Тогда оставшуюся часть слова можно разбить на 2 части, так что левая часть определяет некую функцию n перпеменных, а правая часть разбывается на n частей, каждая из которых определяет... $[C,\ D_f\ ,\ D_{q_1}, D_{q_2}, \ldots, D_{q_n}\ ]$

(e) P : To остав $[P,\ \alpha'\ ,\ \alpha''\ ]$; $\alpha'\ -\ g\,(x_1 \dots x_n)$; $\alpha''\ -\ h\,(x_1, \dots x_n, x_{n+1}, x_{n+2})$

(f) M - аналогично.

3. Опаределим последовательность 3, составленных деревьями определениями рекурсивных функция , кодами которых являются элементами последовательности 2. $D_0, D_1, \dots, D_i, \dots$ Сдесь последовательность 3 является конструктивной, поскольку существует алгоритм, посоледовательного построения этой последовательности.

4. Определим последовательность $f_0, f_1, \dots, f_i, \dots$ представимых деревьями из последовательности 3. Такая последовательность является вычислимым. Существует алгоритм, который по номеру i строит определение функции f_i и делает ее вычислимой на любом наборе значений ее переменных. $f_i\,(\dots;\dots)$

5. Определим последовательность 5 $h_0, h_1 \dots, h_i, \dots$ составленную функциями последовательностип 4, к которым применяется орерация отождествления переменных. $f_i\left(x_{i_1} \dots x_{i_{m_i}}\right) \implies h_i = f_i\left(I'_{\prime}\,(x_{\prime}), \dots I'_{\prime}\,(x_{\prime})\right)$ Сдесь последовательность 5 является вычислимой, поскольку существует алгоритм, посоледовательного построения этой последовательности. Можно вычестить $h_i\,(x)$.

Замечание. Последовательность 4 содержит все частично рекурсивные функции, а каждая такая функция встречается в 4 бесконечно много раз.

2.9 Универсальные функции.

Определение. $U\,(n,x)$ - называется универсальной для множества всех ЧРФ от 1ной переменной, если $\overset{def}{\iff}$ выполняются 2 требования:

1. $U\,(n,x)$ - вычислима.

2. $\forall i\ \exists n\,(h_i\,(x) = U\,(n,x))$

Теорема. *Существуют универсальные ЧРФ.*

Доказательство. Определим функцию $U\,(n,x) = h_n\,(x)$. Покажем, что $U\,(n,x)$ - вычислима. Пусть заданы значения n и x. Построим определения дерева D_n. $U\,(n,x) \implies \begin{array}{c} D_n \\ \Downarrow \\ f_n\left(x_{n_1} \dots x_{n_{m_n}}\right) \end{array} \implies D'_n$. Преобразуем D_n в D'_n являющееся корнем дерева 4. $D'_n = C \begin{cases} D_n \\ D_{I'_1(x_1)} \\ D_{I'_1(x_1)} \end{cases}$ Подставим в дерево D'_n вместо переменных значения x и выполним необходимые вычисления. Приведенная последовательность действий задает общий алгоритм вычисления $U\,(n,x)$, поэтому $U\,(n,x)$ является вычислимой. Поскольку $\forall n\ \ (U\,(n,x) = h_n\,(x))$ то справедливо и 2е утверждение. □

2.10 Разрешимость алгоретмических проблем.

1. Проверка эквивалентности программ по ее номеру. $f_n \equiv f_m$

2. ...

Определение. Характерестическа я функция $A \subseteq N^K$ называется фукнция $X_A\,(x_1 \dots x_k) = \begin{cases} 1, & (x_1 \dots x_k) \in A \\ 0, & (x_1 \dots x_k) \notin A \end{cases}$.

Определение. $A \subseteq N^K$ - разрешимое множество $\overset{def}{\iff}$ X_A - вычислимо.

Поскольку множество всех подмножеств множества натуральных чисел неравномощно мощности счетных бесконечных множеств, то существует неразрешимое множество

2.10.1 Неразрешимость проблемы остановки.

Проблема остановки: Если у вас на универсальной ЭВМ заданы начальные данные, вычисление завергиться или нет?

$A_1 = \{(n, x) \,|h_n(x) \ - \ \text{должно быть определено}\}$

Теорема. A_1 - *неразрештмое множество.*

Доказательство. Предположим противное. A_1- разрешимое множество.

$X_{A_1}(n, x) = \begin{cases} 1, & h_n(x) \ - \ \text{определено} \\ 0, & h_n(x) \ - \ \text{неопредеоено} \end{cases}$ - вычислима.

Определим вспомогательную функцию $p(x) = X_{A_1}(x, x)$

Поскольку X_{A_1} - вычислима, то $p(x)$ - вычислима.

$d(x) = \begin{cases} 1, & p(x) = 0 \\ -, & p(x) = 1 \end{cases}$

Так как p - вычислима, то d - тоже вычислима.

Поскольку $d(x)$ - вычислимая, то $\exists i \ \ (d(x) = h_i(x))$.

Возьмем подходящее значение i для которого $d(x) = h_i(x)$

Рассмотрим значение $d(i)$. Возможны 2 случая:

1. $d(i)$ - определено. $\Longrightarrow d(i) = 1 \Longrightarrow p(i) = 0 \Longrightarrow X_{A_1}(i, i) = 0 \Longrightarrow h_i(i) -$ не определено $\Longrightarrow d(i)$ - определено. Так как $(i, i) \notin A_1$. $h_i(i) = d(i)$. Этот случай невозможный.

2. $d(i)$- не определено. $\Longrightarrow p(i) = 1 \ \Longrightarrow \ X_{A_1}(i, i) = 1 \ \Longrightarrow \ h_i(i)$- определено$\Longrightarrow \ d(i)$ $-$определено.

Поскольку оба случая невозможны, то получаем противоречие, которое означает что A_1 - неразрешимое множество. ☐

Замечание.

d	0	1	2	...
h_0	$h_0(0)$	$h_0(1)$	$h_0(2)$	
h_1	$h_0(0)$	$h_0(0)$	$h_0(0)$	
h_2				
...				
h_i				
...				

2.10.2 Неразрешимость проблемы всюдуопределенности.

Определим множество $A_2 = \{n | d_n(x) \ - \ \text{всюдуопределена}\}$.

Теорема. A_2 - *неразрешиомое.*

Доказательство. Предположим противное.

$X_{A_2}(n) = \begin{cases} 1, & d_n \ - \ \text{всюдуопределена} \\ 0, & d_n \ - \ \text{невсюдуопределена} \end{cases}$

Вспомогательная функция $g(n)$

$g(0) = \mu t \,(d_t(x) \ - \ \text{всюдуопределена}) \ \Longrightarrow \ g(0) = \mu t \,(X_{A_2}(t) = 1)$

$g(n+1) = \mu t \,(t > g(n) \,\& X_{A_2}(t) = 1)$

$d_0, d_1 \ldots$ - 5я последовательность

$d_{g(0)}, d_{g(1)} \ldots$ - всюдуопределенная последовательность

Так X_{A_2} - вычислимое, тогда g - тоже вычисоимая.

1) $h(x) = d_{g(x)}(x) + 1$ - Хитрый способ, мы по нему не пойдем.

$U(n, x) = d_n(x)$

Определим функцию $H(n, x) = U(g(n), x)$.

$H(n, x)$ - универсальная для множества всех одноместных всюдуопределенных функций.

$H(o, x) = d_{g(0)}(x)$

$H(1, x) = d_{g(1)}(x)$

$H(2, x) = d_{g(2)}(x)$

...

Определим функцию $h(x) = H(x,x) + 1$

Поскольку H - вычислима, то h - тоже вычислима.

Поэтому $\exists i \ \left(h(x) = d_{g(i)}(x)\right)$

Тогда $h(i) = H(i,i) + 1 = d_{g(i)}(i) + 1$, а с другой стороны $h(i) = d_{g(i)}(i)$.

Поскольку $d_{g(i)}$ -всюдуопределенная, так как $d_{g(i)}(i) + 1$ и $d_{g(i)}(i)$ существуют и яявляются разными. Это противоречие.

Следовательно предположение о том что характеристическия функция X_{A_2} является вычислимой - не верно. Значит A_2 - неразремое множество. □

2.10.3 Неразрешимость проблемы эквивалентности.

Опреденим $A_3 \{(n,m) | d_n \equiv d_m\}$

Теорема. A_3 - *неразрешимо.*

Доказательство. От противного.

$X_{A_3}(n,m) = \begin{cases} 1, \ d_n \equiv d_m \\ 0, \ d_n \neq d_m \end{cases}$

Сведем эту теорему к 2му случаю(A_2 - неразрешиомое.).

Определим вспомогательную функцию, основанную на преобразовании $d_n(x) \implies sg(d_n(x)) + \overline{sg}(d_n(x)) = g_n(x)$; $n = 1,2,3 \ldots$ $sg(x) = \begin{cases} 1, \ x > 0 \\ 0, \ x = 0 \end{cases}$

$d_n \implies g_n$ - Обладает следующим свойством: $d_n(x)$ −всюдуопределена $\iff g_n(x) \equiv 1$.

Определим значения номеров, под которыми $g_n(x)$ встречается в последовательности 5.

Покажем, что существует такая вычислимая функция $p(n)$, что $g_n(x) = d_{p(n)}(x)$.

Уточним схему вычисления знеачения $p(n)$.

1. $d_n(x) \implies D_n = f_n(\ldots)$
2. Трансфопмируем это дерево $D_n \implies D'_n = d_n(x) = f_n\left(I_1^1(x) \ldots I_1^1(x)\right)$
3. преобразуем D'_n в D''_n - дерево определения функции $g_n(x) = sg(d_n(x)) + \overline{sg}(d_n(x))$

   $C - \left\langle \begin{matrix} D_+ \\ D_{sg(d_n(x))} \\ D_{\overline{sg}(d_n(x))} \end{matrix} \right. \implies$

   Пусть $\alpha_{D''_n}$ - код этого дерева $= \beta_m$. тогда положим $p(n) = m$. Поскольку D''_n является девевом определения функции $g_n(x)$, то β_m - код этогог дерева. Значит $d_m(x) = g_n(x)$.

   $d_0, d_1 \ldots d_n \ldots$

   $d_n \implies g_n(x) \implies D''_n = D_m$

Поэтому $\forall n \ \left(g_n(x) = d_{p(n)}(x)\right)$

Определим вспомогательную функцию $H(n) = X_{A_3}(i_0, p(n))$, $d_{i_0}(x) \equiv 1$

Тогда

$$H(n) = \begin{cases} 1, \ d_{i_0} \equiv d_{p(n)} \\ 0, \ d_{i_0} \neq d_{p(n)} \end{cases} \iff \begin{cases} 1, \ d_{p(n)} \equiv 1 \\ 0, \ d_{p(n)} \neq 1 \end{cases} \iff$$
$$\iff \begin{cases} 1, \ d_n \ - \ \text{всюдуопределена} \\ 0, \ d_n \ - \ \text{невсюдуопределена} \end{cases}$$

$\implies H(n)$ - Характеристическая функция A_2

Поскольку A_3 вычислима, $p(n)$ - вычислима, то $H(n)$ - вычислимая характеристическая функция множества A_2. Последнее противоречит неразрешимости множества A_2 , значит X_{A_3} не может быть вычислимой, а A_3 - неразрешимое. □

2.11 Вычислительная сложность програм.

Вычислительная сложность програм связана с оценкой потребляемых им ресурсов, времени и памяти.

Пусть P -некоторая программа, a - начальное данное этой программы. Временем работы $P(a)$, $T_P(a)$ называется количество действийь(команд) выполненных P при работе при начальном данном a.

Под длейтсвиями или операциями будут пониматся преобразования, физическое время выполнения которых яалвяется сравнимым на любых начальных данных. Для реальных программ действиями считаются $+1, -1$, сравнения.

Определим характеристику данных программы P называемого рамером данных. $(||a||)$ Данная система является имперической, и связанною с предпологаемыми алгоритмами обработки. Если A это число, то размером A можно взять длинну 2чного представдения A, если A- массив, при обработки которого не выполняются сложные вычисления с элементами массива, то тогда это число элемантов в этом массиве,а есть наоборот, то размер a - это сумма размеров его отдельных элементов. Тогда вычислительной сложностью программы P называется $t_P(n) = \max\limits_{||a||=n} (T_P(a))$. На практике $t_p(n)$ обычно рассматривают с точностью до порядка, как $O(t_P(n))$.

2.12 Классы функций сложности пролграм.

$$\left.\begin{array}{r}t_p(n) = o(c),\ c - const\\ t_p(n) = o(\log_2 n)\\ t_p(n) = o(n)\end{array}\right\}\text{Алгоритмы реального времени.}$$
$$\left.\begin{array}{r}t_p(n) = o(n \cdot \log_2 n)\\ t_p(n) = o\left(n^k\right)\end{array}\right\}\text{Практически приминимые.}$$
$t_p(n) = o(2^n)$ Не приминимые алгоритмы.

Пример. Рассмотрим задачу сортировки целочисленного массива, который должен быть упорядочен по вощростанию.

```
var a[1..n] -массив
for i:=1 to n-1 do
    begin
    for j:=i+1 to n do
       begin
       if a[i]>a[j] then
          a[i]↔a[j];
       end;
    end;
```

$\underline{\underline{n}}$

$S_1 + S_2 + \ldots + S_{n-1}$ - так вычисляется сложность программ.
$S_1 = 2 + (n-1) \cdot 6$
$S_2 = 2 + (n-2) \cdot 6$
$\ldots$
$S_{n-1} = 2 + 1 \cdot 6$
Общее время равно $2(n-1) + 6(1 + 2 + \ldots + (n-1)) = 2(n-1) + 6\left(\frac{n}{2}(n-1)\right) = o\left(n^2\right)$

Пример. Будем рассматтривать целочисленный массив, который должен быть упорядочен по возростанию. $(n \cdot \log_2 n)$

```
a[1..n]
```

Будем представлять массив a с помощью насыщеннгого бинарного дерева. В таком дереве ярусы полность запалнены в порялдке сверху вниз, за исключение последненго, который может быть заполнен чистично слева направа. ... в порядке обхода его вершин по ярусам, ...

```
7,14,2,3,8,9,6,3,10,5
```

$$\begin{array}{c} 7 \\ \swarrow\searrow \\ 14 \quad 2 \\ \swarrow\searrow\swarrow\searrow \\ 3 \quad 8 \quad 9 \quad 6 \\ \swarrow\searrow\ \searrow \\ 3 \quad 10 \quad 5 \end{array}$$

В таком дереве левые и правые потомки верщини $a[i]$ соответствуют элементам массива $a[2i]$ и $a[2i+1]$. Это свояство можнодоказать индукцией по i.

1. Безис $i = 1$

   $a[1]$ $a[2]$ $a[3]$

2. Индуктивное предположение. $i = k$

   $a[k]$ $a[2k]$ $a[2k+1]$

3. Индуктивный переход. $i = k + 1$

   $a[k+1]$

   $a[2k+1+1] = a\left[2\left(k+1\right)\right]$

   $a[2k+1+1+1] = a[2\left(k+1\right)+1]$

Оценим глубину(количество ярусов) этого дерева.
Если у нас k ярусов, то в массиве $2^0 + 2^1 + \ldots + 2^{k-1} \leq 2 \cdot 2^{k-1} \sim 2^k$
$k \sim \log_2 n$
Быстрая сортировка массива a выполняется в 2 этапа.

1. Преобразования a в массив соответствующий сортирующим деревом.(Дерево, сопоставленное массиву называется сортирующим, если в нем на каждом пути из корня в лист элементы упорядочены по возростанию.)

   Будем рассматривать элементы a в обратном порядке. На каждом шаге рассматривается очередной элемент a[i].

   (a) ... со значениями в вершанах потомков
   (b) Если a[i] меньше одного из своих потомков, по переставим его местами с меньшим из его потомков.
   (c) Для нового положения a[i] повторим это действие.

   Когда процесс построения сортирующего дерева закончится, то массив a будет преобразован в массив соответствующей сортирующему дереву. Это так потому что в противном случае в дереве будет такой элемент $a[i] > a[2i]$ либо $a[i] > a[2i+1]$. В обоих случаях это противоречит...

2. Выписывание упорядоченово массива по сортирующему дереву.

$$\begin{array}{c} 2 \\ \swarrow\searrow \\ 3 \quad 6 \\ \swarrow\searrow\swarrow\searrow \\ 3 \quad 5\ 9 \quad 7 \\ \swarrow\searrow\ \searrow \\ 14 \quad 10 \quad 8 \end{array} \Longrightarrow \begin{array}{c} 3 \\ \swarrow\searrow \\ 3 \quad 6 \\ \swarrow\searrow\swarrow\searrow \\ 10 \quad 5\ 9 \quad 7 \\ \swarrow \quad \searrow \\ 14 \quad 8 \end{array} \Longrightarrow \begin{array}{c} 3 \\ \swarrow\searrow \\ 5 \quad 6 \\ \swarrow\searrow\swarrow\searrow \\ 10 \quad 8\ 9 \quad 7 \\ \swarrow \\ 14 \end{array} \Longrightarrow \begin{array}{c} 5 \\ \swarrow\searrow \\ 8 \quad 6 \\ \swarrow \quad \swarrow\searrow \\ 10 \quad 9 \quad 7 \\ \swarrow \\ 14 \end{array} \text{ и так далее}$$

   На каждом шаге в массив добавляется наименьшее из значений осваышхся в массиве.

$n \cdot \log_2 n$ - время.

3 Системы Поста.

Будем рассматривать 3 алфавита A, B, V, которые назовем основным алфавитом, вспотогательным алфавитом и алфавитом переменных соответственно. $|A \cup B| \geq 2$ Будем считать A, B, V- дезьюнктные(попарно непересекающиеся).

Назовем $t \in (A \cup B \cup V)^*$ образцом, эталоном и тп.

Если t - некоторый образец, и $x_1 \ldots x_n$ - символы переменных t, то применением t называется результат замены $x_1 \ldots x_n$ на непустые слова в алфавите $A \cup B$.

Для формального опрнеделений применения образца, определим понятия подсттановки $\Theta = \begin{pmatrix} x_1 \ldots x_n \\ \alpha_1 \ldots \alpha_n \end{pmatrix} \alpha_1 \ldots \alpha_n \in (A \cup B)^*$($\alpha_1 \ldots \alpha_n$ - непустые) Тогда применения t с помощью Θ, это $t \cdot \Theta = \alpha$ слово, получаемое из из t заменой каждого вхождения $x_1 \ldots x_n$ на соответствующее слово в Θ. Если t образец то U_t - множество применений t, в которых переменные были заменены на непустые слова.

Теорема. *Если t_1, t_2 - образци и $U_{t_1} \equiv U_{t_2} \iff t_1, t_2$ - совподают с точностью до переименования переменных.*

Доказательство. " $\Longrightarrow$ "

$t_1 = \sigma_1 \ldots \sigma_n$

$t_2 = \delta_1 \ldots \delta_m$

$\Longrightarrow m = n$ поскольку в противном случае кратчайшее применение t_1 и t_2 имеет разные длинны.

Будем рассматривать пары $(\sigma_i; \delta_i)$.

1. Если $\sigma_i, \delta_i \in A \cup B$ они совподают, тогда если бы эти символы были разные, то кратчайшим применением t_1 и t_2, с - символы были бы разные, что противоречит $U_{t_1} \equiv U_{t_2}$

2. $\left.\begin{matrix} \sigma_i \in V \\ \delta_i \in A \cup B \end{matrix}\right\} \Longrightarrow$ кратчайший применений t_1 в i - позиции встречается не менее 2х разных символов, а в кратчайшем применении t_2 в i - позиции встречается ровно одн. $\delta_i \Longrightarrow$ противоречис с $U_{t_1} \equiv U_{t_2} \Longrightarrow$ невозможен 2й случай.

3. $\left.\begin{matrix} \delta_i \in V \\ \sigma_i \in A \cup B \end{matrix}\right\}$ - так же не возможен.

4. Пусть $\begin{matrix} \sigma_i \in V \\ \delta_i \in V \end{matrix}$. Пусть $\sigma_i = \sigma_j$, покажем что $\delta_i = \delta_j$. Если это не было так, то тогда кратчайшим применением t, были бы одинаковы в j и птi , а кратчайшее применение t_2 - были бы разные $\Longrightarrow$ противоречие $U_{t_1} \equiv U_{t_2}$. Значит совполающим в i и j символан t_1, в t_2 соответствуют сопвподающие. Поменяв в рассуждение t_1 и t_2 можно прийти в заключению, что одному символу из t_1 соответствует 1 символ из t_2 из $1 \ldots n \Longrightarrow$ что t_1 и t_2 совподают с точностью до переименования переменных.

" $\Longleftarrow$ "

Пусть t_1 и t_2 совподают с точностью до переименования переменных, докажем что $U_{t_1} \equiv U_{t_2}$.

Сведем к $U_{t_1} \subseteq U_{t_2}$ и $U_{t_1} \supseteq U_{t_2}$.

$U_{t_1} \subseteq U_{t_2}$

Пусть $t_2 = t_1 \cdot \Theta$, где Θ подстановка сопоставлемая переменным t_1 и t_2.

Пусть $\alpha \in U_{t_1} \Longrightarrow \alpha = t_1 \cdot \Theta' \Longrightarrow \alpha = \left(t_2 \cdot \Theta^{-1}\right) \Theta' = t_2 \Theta^{-1} \Theta' \Longrightarrow \alpha \in U_{t_2}$

$U_{t_1} \supseteq U_{t_2}$ - Доказывается аналогично. □

Определение. Продукцией называется выражение $\dfrac{t_1 \ldots t_n}{t_{n+1}}$, где $t_1 \ldots t_n$ - образци в A, B, V.

Продукции $t_1 \ldots t_n$ - посылки, t_{n+1} - заключение.

Если $n = 0 \Longrightarrow$ продукция это аксиома.

Пусть $\alpha_1 \ldots \alpha_n \in (A \cup B)^*$ тогда $\Pi = \dfrac{t_1 \ldots t_n}{t_{n+1}}$ применимы к словам $\alpha_1 \ldots \alpha_n$.

Если существует $\exists \Theta \ \forall i = \overline{1, n} \, (\alpha_i = t_i \Theta)$.

Результатом Π к $\alpha_1 \ldots \alpha_n$ с Θ - это слово $\alpha_{n+1} = t_{n+1} \Theta$.

Определение. Системой поста (продукционной системой) называется четверка $P = (A, B, V, \Pi)$, где A - основной алфавит, B - вендм. алфавит, V - переменные, Π - конечное множесмтво подукции этих алфавитов.

Определение. Последовательность слов $\alpha_1 \dots \alpha_n$ в $(A \cup B)$ образуют вывод к P если $\forall j = 1, 2 \dots n \dots \ \alpha_j$ - является применением некоторой аксиомы, либо применением одной из П к алфавиту уже включенным к выводу.

Понятие вывода обладает свойством недертишиноваси, тоесть $\alpha_1 \dots \alpha_n$ составляет вывод, то его можно продолжить. В качестве α_{i+1} можно записать α_1 и α_2 и т п., тоесть можно продолжить и без разници как $\alpha \in (A \cup B)^*$ выводимо в P, если в P существует вывод, содержащий α.

3.1 Задачи, решаемые системами поста.

Задачей в системах поста называется выражение $t-?$. Решение $t-?$ это все слова $\alpha \in (A \cup B)^*$ такое что $\exists \Theta\, (\alpha = t\Theta)$, тоесть решение, этот найти Θ.

Тогда если t содержит символы неизвестных переменных, то Θ указывает на значение этих переменных.

Одна и таже задача имеет бесконечно много решений, или нетимеет их совсем.

Для решения произвольных задач в системах поста, которая моделирует последовательность построения всевозможных выводов в этой системе.

Пусть $t-?$ - постановка задачи. Будем считать решением $t-?$ шагами , так что на шаге с номером i. строятся все выводы длинны i, в которых применяются Θ, заменяющие символы переменных на неболее чем i символьные слова.

$t-?$ - постановка задачи.

$P = (A, B, V, \Pi)$

$\Pi_1 \dots \Pi_r$ - аксиомы. $\Theta_1 \dots \Theta_S$ - подстановки, где все переменные заменяются на не более чем 1носимвольныет сллова.

1. Шаг 1. Пусть R - множество всевозможных подстановок вместо переменных из V слов $A \cap B$. Пусть $\pi^1 \dots \pi^t$ - все возможные аксиомы из П. Построим всевозможное применение этих аксиомЮ плочаемое из подстановок из R. $\alpha_1 \dots \alpha_n$

   Представим ..

2. Шаг 2. Пусть R - множество всевозможных подставовок, заменяющее .. переменых на неболле чем 2х символьные слова. Построим всевозможные применение аксиом $\pi^1 \dots \pi^t$ с помощью подстановок из R. Продолжим каждый из полученных начал выывода еще на 1н шаг. Для произвольного α_i' построим ... Добавим к таким выводам те, которые получаются применением продукции не аксиом из П к словам, которые уже включены в выводы. Продолжение выводов, начинающихся с α_i' начинающиххся .... выполняется по правилу:

   (a) Последовательно ..

   (b) Если выбрана некоторая продукция $\pi = \dfrac{t_1 \dots t_q}{t_{q+1}}$ то возьмем последователдьность $\alpha_i \alpha_i \dots \alpha_i$ содержащая q раз α_i .

   (c) Будем переберать подстановки из $\alpha_i \alpha_i \dots \alpha_i$ способом $\forall i = \overline{1, q}\, (\alpha_i' = t_i \Theta)$. Всякий раз когда условие выполнена, продолжим вывод длинны 1 до вывода длинны 2, добавив к нему слово $t_{q+1}\Theta$.

3. Шаг с номером k. Пусть R - множество всевозможных подстановок вместо переменных из V переменных из АиБ .. слов длинны k. R - конечное множество. Будем строить всевозмоджные выводы длинны k, к которым применяюттся подстановки по следующей схеме:

   Для такого начала построим всевозмыжные продолжения, добавив еще 1но слово. Сначала построим всевозможные продолжения, получаемых применением аксиом с помощью подстановок из R. Затем продолжим выбранное начало вывода длинны i с помощью применение продукции неаксиом. Для этого воспользуемся следующей схемой:

   (a) Будем последовательно просматриватть все продукции ...

   (b) Для каждой продукции $\dfrac{t_1 \dots t_q}{t_{q+1}}$ будем просматривать всевозможные последовательности слов $\beta_1 \dots \beta_q$ составляемые из слов $\alpha', \alpha'' \dots \alpha^{(i)}$. После этого среди подстановок из R выберает такие, для которых выполняется условие $\forall i = \overline{1, q}\, (\beta_i = t_i \Theta)$. Если выполнено, то добавляем слово.

и так далее.

Покажем что приведенный алгоритм построения выводов не все выводыт строит.

Пусть $\alpha \in \Omega_p$- (Ω_p - мнодество слов ..) если α выводится в p в составе вывода длинны k и в этом выводе применяются подстановки, заменяющие символы переменных на неболее чем s символьные слова. Тогда по приведенному алгоритма работы механизма вывода α будет встречаться в выводах на шаге $\max(k, s)$

3.2 Свойства множеств слов выводимыми системами поста.

Пусть заданы системы поста $P_1 = (A_1, B_1, V_1, \Pi_1)$ и $P_2 = (A_2, B_2, V_2, \Pi_2)$. Потребуем чтобы множества все кроме $\Pi_1 \cup \Pi_2$ были дезьюнктными. Обозначим как Ω_{p_1} и Ω_{p_2} - мнодество слов выводимые P_1 и P_2 не содержащие символов вспомогательного алфавита.

Теорема. *Существуют в системах поста $P_3 = (A, B, V, \Pi_3)$ которое .... $\Omega_{p_3} = \Omega_{p_2} \cup \Omega_{p_1}$ ($\Omega_{p_3} = \Omega_{p_2} \cap \Omega_{p_1}$)*

Доказательство. Определип системы поста P_3, положим $A = A_1 \cup A_2,\ B = B_1 \cup B_2,\ V = V_1 \cup V_2$. Пусть S и Q - 2 новых вспомогательных символа. Преобразуем продукции из P_1 по правиу $\dfrac{t_1 \dots t_n}{t_{n+1}} \implies \dfrac{St_1 \dots St_n}{St_{n+1}}$. Преобразуем продукции из P_2 : $\dfrac{t_1 \dots t_n}{t_{n+1}} \implies \dfrac{Qt_1 \dots Qt_n}{Qt_{n+1}}$. Обьеденим полученные множества продукции.

Добавим к ним еще 2 продукции. $\dfrac{Sx}{x}$ и $\dfrac{Qx}{x}$.

Покажем что $\Omega_{p_3} = \Omega_{p_1} \cup \Omega_{p_2}$.

Проверем включения (1) $\Omega_{p_3} \subseteq \Omega_{p_1} \cup \Omega_{p_2}$ и (2) $\Omega_{p_1} \cup \Omega_{p_2} \subseteq \Omega_{p_3}$

1. $\alpha \in \Omega_{p_3} \implies S\alpha \text{ } or \text{ } Q\alpha \implies \alpha \in \Omega_{p_1} \text{ } or \text{ } \alpha \in \Omega_{p_2} \implies \alpha \in \Omega_{p_1} \cup \Omega_{p_2}$
2. $S\alpha \text{ } or \text{ } Q\alpha \implies \alpha \in \Omega_{p_3} \text{ } or \text{ } \alpha \in \Omega_{p_3} \implies \alpha \in \Omega_{p_3}$

Замечание. В случае когда $\Omega_{p_3} = \Omega_{p_2} \cap \Omega_{p_1}$ достаточно преобразовать продукции P_1 и P_2 по приведенной схеме, и добавить к продукциям еще 1ну $\dfrac{Sx\ Qx}{x}$.

□

Пример. P_1 0 $\dfrac{x}{x0}$ выводятся любые последовательности нулей, P_2 1 $\dfrac{x}{x1}$- цепочки из 1ц, а если их обьеденить, то получится хлам.

3.3 Функции, вычисляемые системами поста.

Будем рассматривать представление произвольных чисел из N.

Определение. Числовая функция $f(x_1 \dots x_n)$ $(f : N^n \longrightarrow N)$ вычисляется в системе поста P только тогда когда $\forall x_1 \dots x_n \in N\, (f(x_1 \dots x_n) = y \longleftrightarrow$ в P_3 выводится слово $f(\overline{x_1} \dots \overline{x_n}) = \overline{y})$.

Теорема. *Всякая частично рекурсивная поста вычисляется системами поста.*

Доказательство.

1. Высислимость в системах поста простейших функций.
   (a) Представление чисел из N
   (b) $O(x) = 0$
   $\dfrac{Nx}{O(x) = 0}$
   (c) $S(x) = x + 1$
   $\dfrac{Nx0}{S(x0) = x1}$ $\dfrac{Nx1\ S(x) = y}{S(x1) = y0}$
   (d) $I_n^m(x_1 \dots x_n) = x_m$
   $Nx_1 \dots Nn$
   $I_n^m(x_1 \dots x_n) = x_m$

2. Вычислимость операции суперпозиции.

   Пусть $h(x_1\dots x_k) = f(g_1(x_{11}\dots x_{1m_1})\dots g_n(x_{n1}\dots x_{nm_n}))$ где $f(x_1\dots x_n)$ вычисляется в системах поста P_f , а $g_i(x_{i1}\dots x_{im_i})$ в P_{g_i}. Пусть $F, G_1\dots G_n$ - вспомогательные символы, которые не содержаться среди символов переменных в системах поста $P_f, P_{g_1}\dots P_{g_n}$. Преобразуем продукции P_f по правилу $\dfrac{t_1\dots t_k}{t_{k+1}} \longrightarrow \dfrac{Ft_1\dots t_k}{Ft_{k+1}}$, а $Pg_i,\ i=\overline{1,n}$ по правилу $\dfrac{t_1\dots t_k}{t_{k+1}} \longrightarrow \dfrac{G_it_1\dots G_it_k}{G_it_{k+1}}$. Обьеденим определенные выше множества продукции, и добавим в них еще 1ну продукцию $G_1g_1(x_{11}\dots x_{1m_1})\,G_2g_2(x_{21}\dots x_{2m_2})\dots G_ng_n(x_{n1}\dots x_n$ тогда $G_ig_i(x_{i1}\dots x_{im_i}) = y_i,\ i=\overline{1,n}$ и $Ff(y_1\dots y_n) = z$ тогда $h(x_1\dots x_k) = z$.

3. Вычислимоть операций примитивной рекурсии.

   Пусть $f(x_1\dots x_n, x_{n+1}),\ n\geq 0$ получается из $g(x_1\dots x_n)$ и $h(x_1\dots x_n, x_{n+1}, x_{n+2})$ в помощью операции примитивной рекурсии. $f(x_1\dots x_n, 0) = g(x_1\dots x_n)$ и $f(x_1\dots x_n, y+1) = h(x_1\dots x_n, y, f(x_1\dots x_n, y))$ Определим систему поста, в которой вычисляется функция f. $S(x) = x+1$ там встречается, так что оно в таком виде мешается. Обозначим G, H, Σ как вспомогательные символы, отсутствующие в алфавитах систем поста P_h, P_g, P_S. Преобрахуем продукции перечисленых систем по правилу $P_g : \dfrac{t_1\dots t_k}{t_{k+1}} \longrightarrow \dfrac{Gt_1\dots Gt_k}{Gt_{k+1}}$, $P_h : \dfrac{t_1\dots t_k}{t_{k+1}} \longrightarrow \dfrac{Ht_1\dots Ht_k}{Ht_{k+1}}$, $P_S : \dfrac{t_1\dots t_k}{t_{k+1}} \longrightarrow \dfrac{\Sigma t_1\dots \Sigma t_k}{\Sigma t_{k+1}}$. Обьезденим полдученные множества продукции и добавим к ним еще две. $g(x_1\dots x_n) = z$ тогда $f(x_1\dots x_n, 0) = z$ и еще 2е соотношение : Пусть $\Sigma S(y) = v$ тогда $f(x_1\dots x_n, y) = p$ и $Hh(x_1\dots x_n y, p) = z$ и тогда $f(x_1\dots x_n, v) = z$.

4. Вычислимость операции минимизации.

   Пусть $f(x_1\dots x_n, x_{n+1}) = \mu t\,(g(x_1\dots x_n, t) = x_{n+1}),\ n\geq 0$. $f(x_1\dots x_n, x_{n+1})$ - минимальное t такое что

   (a) $g(x_1\dots x_n, t) = x_{n+1}$

   (b) $g(x_1\dots x_n, i),\ i\leq t$ - определены

   Пусть P_g - система поста, в которой вычисляется функция g, и P_S - $S(x) = x+1$. Допистум что щас мы уже развели продукции, и не будем описывать щас вспомогательные символы. Определим вспомогательную функцию
   $$\varphi(x_1\dots x_n, x_{n+1}, x_{n+2}) = \begin{cases} 0, & \text{если } g(x_1\dots x_n, i) \text{ определены } i\leq x_{n+2} \text{ и не равны } x_{n+2} \\ 1, & \text{если } g(x_1\dots x_n, i) \text{ определены для } i\leq x_{n+2} \implies \\ & g(x_1\dots x_n, x_{n+1}, x_{n+2}) = x_{n+1}\ \forall j < x_{n+2}\,(g(x_1\dots x_{n1}) \neq x_{n+1}) \end{cases}$$

   Выпишим продукции, позволяющие вычислять значения функции φ. $g(x_1\dots x_n, 0) = y\ y\neq x_{n+1}\ \varphi(x_1\dots x_n, x_{n+1}, 0) =$ 0 это есть нет. Если да: $g(x_1\dots x_n, 0) = x_{n+1}\ \varphi(x_1\dots x_n, x_{n+1}, 0) = 1$. Если $\varphi(x_1\dots x_n, x_{n+1}, t) = 0$ $S(t) = d\ g(x_1\dots x_n, d) = y\ y\neq x_{n+1}$ - Это есть нет, то в этом случае делаем $\varphi(x_1\dots x_n, x_{n+1}, d) = 0$. Если да : $\varphi(x_1\dots x_n, x_{n+1}, t) = 0\ S(t) = d\ g(x_1\dots x_n, d) = x_{n+1}$ то $\varphi(x_1\dots x_n, x_{n+1}, d) = 1$ К выписанным продукциям добаыим еще 1ну $\dfrac{\varphi(x_1\dots x_n, x_{n+1}, x_{n+2}) = 1}{f(x_1\dots x_n, x_{n+1}) = x_{n+2}}$. Можно обойтись без 5ой, если исправить 2ю и 4ю, не запоминать, а сразу выводить ответ.

   Замечание.

   (a) В приведенных продукциях не приведена разметка P_g, P_S - это можно сделать, но на правельность это не влияет, так как мы уже это дклать умеем.

   (b) В системах поста, в которых выводится $\neq$ (неравенство) . Для этого неравенство должно быть определено отдельно.

□

Пример. $x\neq y$

$<$

$\dfrac{x \text{ меньше } y}{x\neq y}$ $\dfrac{x\neq y}{y\neq x}$

$$\frac{Nx\ Ny\ x \preceq y}{x \neq y} \qquad \frac{\begin{array}{l} \_ \preceq 0 \\ \_ \preceq 1 \quad x \preceq y \\ x \preceq y \end{array}}{\begin{array}{ccccc} & 0 & & 0 & \\ x & \begin{array}{c} 0 \\ 1 \end{array} & \preceq g & \begin{array}{c} 1 \\ 0 \end{array} & x \preceq y \\ & 1 & & 1 & \end{array}} \qquad \frac{Nv0x\ Nv1\ z\ xPz}{v0x < v1z} \qquad \frac{xPy}{\begin{array}{cccc} & 0 & & 0 \\ x & \begin{array}{c} 0 \\ 1 \end{array} & Py & \begin{array}{c} 1 \\ 0 \end{array} \\ & 1 & & 1 \end{array}}$$

Где $\preceq$ - меньше по значению и P - равно.

$$\frac{f(x, iz) = y}{x < y}$$

4 Потоки в сетях.

Определение. Сетью называется ориентированный граф $N = (V, U, \{I, S\})$, где V - множество вершин, I - множество ребер $\{I, S\}$ верщины, называемые истоком и стоком сети, такой что из I ребра тольковыходят, в S ребра только входят, для остальных верщин имеются как входдящие и выходящие ребра и для каждой влутренней верщины $v \in V \setminus \{I, S\}$ существует путь в S и из I существует путь в v.

Определение. Сеть $N = (V, U, \{I, S\})$ назхывается транстпортной сетью, если задано отображение $\psi : U \longrightarrow R^+$ (множество неотрицительных чисел), называемые пропускнгой способностью ребер. Будем представлять транстпорную сеть как $N = (V, U, \{I, S\}, \psi)$.

Определение. $\varphi : U \longrightarrow R^+$ называется потоков в транстпортной сети $N = (V, U, \{I, S\}, \psi) \overset{def}{\Longleftrightarrow}$

1. $\forall u \in U \left(0 \leq \varphi(n) \leq \psi(n)\right)$
2. $\forall v \in V \setminus \{I, S\} \left(\underset{u \in E^+(v)}{\Sigma\varphi(u)} = \underset{u \in E^-(v)}{\Sigma\varphi(u)}\right)$

   $E^+(v)$ - ребра, которые ведут в v

   $E^-(v)$ - ребра, которые выходят из v

Определение. Велечиной потока φ транстпортной сети N называется $N(\varphi) = \Sigma\varphi(u)\,,\ u \in E^-(I)$.

Теорема. *Для каждой транстпортной сети $N = (V, U, \{I, S\}, \psi)$ выполняется* $\underset{u \in E^-(I)}{\Sigma\varphi(u)} = \underset{u \in E^+(S)}{\Sigma\varphi(u)}$.

Доказательство. Разобьем доказательство на 2 этапа.

1. Пусть поток φ таков, что в N не существует элементарных циклов длинны не меньше чем 3, по которым течет ненулевой поток.

   Тогда удалим из N все рёбра, по которым течет 0ой поток. В рехультате получим новую сеть, в каторой все пути, выходящие из каждой вершины ведут только в сток. Сопоставим вершинам сети номера ярусов этих вершин, равные максимуму длин путей ведущих из истока в такие вершины. Разместим вершины сети по ярусам. Разобьем ребра, соединяющие рверщины несоседних ярусов вспомогательными точками. Ребра разбиения сохраняют величину потока, пропускную способность разбиваемых ребер. При этом полученной транстпортной сети величина входного потока равна $\underset{u \in E^-(I)}{\Sigma\varphi(u)} \Big| \underset{u \in E^+(S)}{\Sigma\varphi(u)}$. В последней сети суммарный входной поток полностью поступает в вершины 1го яруса. Из этих вершин в том же самом обьеме передается в вершины 2го яруса. Из вершин 2го яруса поток в том же обьеме переходит в вершины 3го яруса и так далее. На последнем шаге поток в том же самом обьеме передается из вершин предпоследнего яруса в сток. Поэтому $\underset{u \in E^-(I)}{\Sigma\varphi(u)} = \underset{u \in E^+(S)}{\Sigma\varphi(u)}$.

2. Предположим что существуют элементарные циклы длинны больше чем 3, по которым течет ненулевой поток.

   Пусть С - некоторый такой цикл. Обозначим как $d = \min\left(\varphi(u)\right),\ u \in C$. Преобразуем поток φ $\forall u \in U \left(\varphi_1(u) = \begin{cases} \varphi(u) - d & \text{,если } u \in C \\ \varphi(u) & \text{,иначе} \end{cases}\right)$.

(a) $\forall u \in U\left(0 \leq \varphi_1(u) \leq \psi(n)\right)$ Верно потому что поток на ребрах вне C не изменяется, а а ребрах C он уменьшается на величину для величину, для которой он не может быть отрицательной.

(b) $\forall v \in V\setminus\{I,S\}\left(\begin{array}{c}\Sigma\varphi(v)=\Sigma\varphi(v)\\ v\in E^-(I)\ \ v\in E^+(S)\end{array}\right)$ Верно поскольку для внутренних вершин не принадлежвщих C суммарный взодной и выходной потоки не изменяются и поэтому остаются равными, а для вершин принадлежашний C суммарные входные и выходные потоки уменьшается на d. Поэтому входные и выходные потоки внутренных верщин оежащих на C так же остаются равными.

Поскольку C не содерщих исток и сток $\begin{array}{c}\Sigma\varphi(u)=\Sigma\varphi_1(u)\\ u\in E^-(I)\ \ u\in E^-(I)\end{array}$ и $\begin{array}{c}\Sigma\varphi(u)=\Sigma\varphi_1(u)\\ u\in E^+(S)\ \ u\in E^+(S)\end{array}$. При преобразовании φ в φ_1 хотябы 1но ребро цикла C станет ребром, по которому течет 0ой поток. Если для φ_1 в сети N существуют элементарные циклы длинны не меньше чем 3 по которым те ненулевой поток, по повторим рассуждение. В результате получим поток φ_2 такой что $\begin{array}{c}\Sigma\varphi_2(u)=\Sigma\varphi_1(u)\\ u\in E^-(I)\ \ u\in E^-(I)\end{array}$ и $\begin{array}{c}\Sigma\varphi_2(u)=\Sigma\varphi_1(u)\\ u\in E^+(S)\ \ u\in E^+(S)\end{array}$ При этом в сети N появятся ребра, по которым течет 0ой поток. И так талее. Продолжаем процелл до тех пор, когда будет получен поток φ_k в котором не существует элементарных циклов ддлинны не меньше чем 3 по которым течет ненулевой поток. Тоесть $\begin{array}{c}\Sigma\varphi_k(u)=\Sigma\varphi_k(u)\\ u\in E^-(I)\ \ u\in E^+(S)\end{array}$.
Поскольку сумма $\Sigma\varphi(u)=\Sigma\varphi_1(u)=\Sigma\varphi_2(u)=\ldots=\Sigma\varphi_k(u),\ u\in E^-(I)$ (для каждого графа отдельно) и $\Sigma\varphi(u)=\Sigma\varphi_1(u)=\Sigma\varphi_2(u)=\ldots=\Sigma\varphi_k(u),\ u\in E^+(S)$(для каждого графа отдельно). И так как $\begin{array}{c}\Sigma\varphi_k(u)=\Sigma\varphi_k(u)\\ u\in E^-(I)\ \ u\in E^+(S)\end{array}$ то там все всему равно.

□

Определение. Поток φ транстпортной сети N называется максимальнмы потоком в данной сети, если $N(\varphi)=\max\left(N(\varphi')\right),\ \varphi'$ - поток в N.

Определение. Поток φ транстпортной сети N называется полным потоком если на каждом элементарном пути вежущем из истока в сток, встречается хотябы 1но ребро $u\left(\varphi(u)=\psi(u)\right)$.

Замечание. Всякий максимальный поток в N является полным потоков в сети N. Обратное утверждение не верно в общем случае.

Теорема. *Для всякой транстпортной сети существует полный поток.*

Доказательство. Проведем доказательство путем метода назыщения дуг.

Определим $\varphi_0 : U \longrightarrow R^+$ и $\forall u\left(\varphi(u)=0\right)$. Опроеделим поток φ_i в N если в сети чуществует хотябы 1 элементарный путь ведущий из истока в сток все ребра которого еще не догружены, то возьмем еще такой поток w определим величитну минимального недогруза ребер $d=\min\left(\psi(u)-\varphi_i(u)\right),\ u\in E(w)$. И преобразуем φ_i в φ_{i+1}. $\forall u\in U\left(\varphi_{i+1}(u)=\begin{cases}\varphi_i(u)+d & ,u\in E(w)\\ \varphi_i(u) & ,\text{иначе}\end{cases}\right)$.

$\forall u\in U\left(0\leq\varphi_{i+1}(u)\leq\psi(u)\right)$

$\forall v\in V\setminus\{S,I\}\left(\begin{array}{c}\Sigma\varphi_{i+1}(u)=\Sigma\varphi_{i+1}(u)\\ u\in E^-(I)\ \ u\in E^+(S)\end{array}\right)$

Поскольку на каждом шаге в сети получается хотябы 1но ребро, максимально нагруженный потоком, то через конечное число шагов данный процесс приведет в построенному максимальному потоку. □

4.1 Сечение сетей.

Разобраться самим.

Определение. Множество $L\in U$ транстпортной сети $N=(V,U,\{I,S\},\psi)$ любой путь в N ведущий из I в S содержит хотяы 1но ребро из L.

$\forall W=I,\ldots,S\ \exists u\in E(w)\left(u\in L\right)$

Если L сечение сети N, то пропускной способностью L называется величина $\begin{array}{c}L(N)=\Sigma\psi(u)\\ u\in L\end{array}$. Для любого сечения в сети N и любого потока в этой сети справедливо неравенство$N(\varphi)\leq L(N)$ (φ - поток, L - сечение, N - сеть)

Содержание