РОССИЙСКАЯ АКАДЕМИЯ НАУК
ОРДЕНА ЛЕНИНА ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
ИМЕНИ М.В. КЕЛДЫША

С.В. Попов

О ДЛИНЕ ВЫЧИСЛЕНИЙ АРИФМЕТИЧЕСКИХ ПРОГРАММ

Москва, 2005г.

УДК 519.95

Попов С.В. О длине вычислений арифметических программ. Препринт Института прикладной математики им. М.В. Келдыша РАН. Москва, 2005г.

Рассматриваются программы над полным базисом, вычисляющие двоичные всюду определенные функции. Доказывается, что если функция, вычисляемая программой, обладают энтропией H, то имеется, по меньшей мере, одно вычисление программы длины 2^{dH} для некоторой положительной константы d.

Popov S.V. About difficulties of the calculations of the arithmetical proGrams. Preprint of the Keldysh Institute of Applied Mathematics of RAS. Moscow, 2005.

There are considered program on full base, computing binary all determined functions. It is proved; if function, computed by the program, possess entropy H, there is calculation of the program of the length 2^{dH} for positive constant d.

Введение. В работе исследуются свойства программ над двоичным арифметическим базисом [1]. Инструментом исследования выступает обобщенный пропозициональный язык, формулы которого строятся над базисом, расширенным за счет введения бесконечных логических операций.

Обобщенные пропозициональные формулы представляют базисные арифметические и логические операторы. Поэтому средств обобщенного пропозиционального языка почти достаточно для представления арифметических программ. После преобразования программы в логическую формулу, в общем случае, получается формула, длина которой пропорциональна вычислению программы над множеством ее входных значений. Если длины вычисления программы не ограничена, то получается формула не ограниченной длины. Поэтому в работе рассматриваются программы для всюду определенных функций, длина вычисления которых ограничена некоторой функцией $p(|\boldsymbol{x}|)$ от длины входа $\boldsymbol{x}$ программы. В этом случае обобщенная функция $F(\boldsymbol{x}, \boldsymbol{y})$, представляющая функцию $\boldsymbol{y} = \varphi(\boldsymbol{x})$, вычисляемую программой $\pi(\boldsymbol{x})$, обладает длиной $c\,p(|\boldsymbol{x}|)$ для некоторой положительной константы c.

Пусть H_x^{φ} есть энтропия функции φ, определенная аргументом $\boldsymbol{x}$. Показывается, что длина формулы $F(\boldsymbol{x}, \boldsymbol{y})$ ограничена снизу величиной $2^{d H_x^{\varphi}}$ для некоторой положительной константы d. Отсюда следует, что имеется хотя бы одно вычисление программы $\pi(\boldsymbol{x})$, длина которого также не меньше, чем $2^{d H_x^{\varphi}}$.

1. Определение программ и их свойства. Пусть $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots\}$ - *входной*, а $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots\}$ – *рабочий* алфавиты. Буквы этих алфавитов назовем соответственно *входными* и *рабочими* переменными. Программой $\pi(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m)$ в базисе $f_1, f_2, \ldots, f_l, P_1, \ldots, P_h$, где $f_1, f_2, \ldots, f_l$ - общерекурсивные функции, $P_1, \ldots, P_h$ – рекурсивные предикаты, называется ориентированный граф, вершинами которого являются выражения следующих двух типов:

1) арифметические операторы $\boldsymbol{y}_j \Leftarrow f_i(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m)$, $1 \le i \le l$;

2) логические операторы $P_i(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q)$, где $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q$ – входные и рабочие переменные, $1 \le i \le h$.

При этом каждая арифметическая вершина имеет ровно одного последователя, логическая – двух (один из которых называется 0-последователем, другой 1-последователем), а заключительная вершина последователей не имеет. Заключительная вершина помечена в точности одной рабочей переменной, которая называется *выходной*. Кроме того, одна из вершин является *начальной* вершиной программы.

Полагаем, что рассматривается программа $\pi(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m)$ с фиксированными наборами входных и рабочих переменных, а также с фиксированной выходной переменной.

Полным состоянием программы $\pi(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m)$ называется пара (ξ, v), где v – вершина, а $\xi \in N^{n+m}$. Если вершина v – заключительная, то полное состояние (ξ, v) называется *заключительным*. Если $\xi = a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m$, то числа $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m$ называются *значениями переменных* соответственно $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m$ в состоянии (ξ, v).

На множестве полных состояний программы π определяется отношение $\rightarrow$: пусть $\xi_1 = a_1, a_2, \ldots, a_n, b'_1, b'_2, \ldots, b'_m$, $\xi_2 = a_1, a, \ldots, a_n, b''_1, b''_2, \ldots, b''_m$, тогда $(\xi_1, v_1) \rightarrow (\xi_2, v_2) \Leftrightarrow$ выполняется одно из следующих условий.

1. Если v_1 – арифметическая вершина $\boldsymbol{y}_j \Leftarrow f(\boldsymbol{y}_{j_1}, \boldsymbol{y}_{j_2}, \ldots, \boldsymbol{y}_{j_r})$, то $b''_j = f(c_1, c_2, \ldots, c_r)$, где $c_1, c_2, \ldots, c_r$ – значения соответствующих переменных, и v_2 является последователем v_1.

2. Если v_1 – логическая вершина $P(\boldsymbol{x}_{j_1}, \boldsymbol{x}_{j_2}, \ldots, \boldsymbol{x}_{j_h}, \boldsymbol{y}_{j_1}, \boldsymbol{y}_{j_2}, \ldots, \boldsymbol{y}_{j_l})$, то $\xi_1 = \xi_2$ и v_2 является $P(a_{j_1}, a_{j_2}, \ldots, a_{j_h}, b_{j_1}, b_{j_2}, \ldots, b_{j_l})$-последователем вершины v_1.

Вычислением программы π для значений $a_1, a_2, \ldots, a_n$ входных переменных называется такая последовательность Ξ полных состояний $(\xi_1, v_1), (\xi_2, v_2), \ldots, (\xi_l, v_l), \ldots$, что $\xi_1 = a_1, a_2, \ldots, a_n, 0, \ldots, 0$; вершина v_1 – начальная; $(\xi_i, v_i) \rightarrow (\xi_{i+1}, v_{i+1})$, $i = 1, 2, \ldots$. Говорим, что путь $\tau = v_1, v_2, \ldots, v_l, \ldots$ соответствует вычислению Ξ. Вычисление называется *терминальным*, если оно конечно и последнее его со-

стояние – заключительное. Путь, соответствующий терминальному вычислению, назовем *терминальным*. Путь из начальной вершины назовем *начальным* путем, а начальный путь, кончающийся заключительной вершиной, - *полным*.

Пусть $\tau = v_1, v_2, \ldots, v_l, \ldots$ - начальный путь в программе π. Для каждой вершины этого пути определим значение рабочих переменных $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m$ следующим образом.

1. В начальной вершине все их значения нулевые.

2. Если в арифметической вершине $\boldsymbol{y}_j \Leftarrow f(\boldsymbol{y}_{j_1}, \boldsymbol{y}_{j_2}, \ldots, \boldsymbol{y}_{j_r})$ переменные $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m$ имеют значения $b_1, b_2, \ldots, b_m$, то в ее последователе переменная $\boldsymbol{y}_j$ имеет значение $f(b_{j_1}, b_{j_2}, \ldots, b_{j_r})$, а остальные переменные не меняют своих значений.

3. В последователях логической вершины все переменные сохраняют свои значения.

Поскольку функции в арифметических вершинах не зависят от входных переменных, то тем самым для любого начального пути значения рабочих переменных во всех его вершинах определены однозначно.

Функцией, вычисляемой программой π, называется частичная функция f_π такая, что $f_\pi(a_1, a_2, \ldots, a_n) = b \Leftrightarrow$ существует терминальное вычисление, начинающееся состоянием $(a_1, a_2, \ldots, a_n, 0, 0, \ldots, 0; v_1)$ и для которого b является значением выходной переменной в заключительном состоянии.

Две программы *эквивалентны*, если они вычисляют одну функцию.

Вершина v программы называется *фиктивной*, если ни один полный путь, соответствующий какому-либо вычислению, не проходит через нее.

Справедливо утверждение.

***Лемма* 1.** *Любая программа эквивалентна программе, в которой нет фиктивных вершин.*

Доказательство. Пусть v есть фиктивная вершина программы. Рассмотрим возможности, которые существуют для различных путей в этой программе, которые завершаются вершиной v.

Пусть путь $\tau = v_1, v_2, \ldots, v_l, v, l > 0$, оканчивается фиктивной вершиной v, и все вершины $v_1, v_2, \ldots, v_l$ арифметические. В этом случае вершины $v_1, v_2, \ldots, v_l$ так же фиктивные, так как если хотя бы одно вычисление содержит вершину v_i, $i = 1, 2, \ldots, l$, то это вычисление содержит и вершину v. Поэтому, исключив из программы вершины $v_1, v_2, \ldots, v_l$ и направив все дуги, ведущие в эти вершины, в вершину v, получим программу эквивалентную исходной.

Пусть теперь путь $\tau = v_1, v$ оканчивается фиктивной вершиной v, вершина v_1 логическая и ее σ-выход ($\sigma \in \{0,1\}$) ведет в фиктивную вершину v. Выбросим вершину v_1, а ведущие в нее дуги направим в ее $(1-\sigma)$-последователя. Получим опять программу эквивалентную исходной.

Из этих преобразований видно, что, последовательно избавляясь от всех фиктивных вершин, мы получим эквивалентную программу, без фиктивных вершин.

Лемма доказана.

Полагаем, что рассматриваемые далее программы не содержат фиктивных вершин.

Будем рассматривать программы над двоичными числами над полным базисом $s(\boldsymbol{x})$, $0(\boldsymbol{x})$ и $\boldsymbol{x} = \boldsymbol{y}$. При этом полагаем, что в арифметических операторах $\boldsymbol{y} \Leftarrow f(\boldsymbol{x})$, переменные $\boldsymbol{y}$ и $\boldsymbol{x}$ совпадают.

Нам потребуется следующее представление двоичных чисел. Каждое двоичное число представляется бесконечной бинарной последовательностью, младшие разряды числа расположены слева, а справа располагается бесконечная последовательность из нулей.

Логическую вершину вида $\boldsymbol{x} = \boldsymbol{x}$ назовем *несущественной*. Легко доказывается

***Лемма* 2.** *Любая программа эквивалентна программе, в которой нет несущественных вершин.*

Поэтому полагаем, что рассматриваемые далее программы не содержат несущественных вершин.

Для удобства пусть начало всякой программы представляет собой m операторов $0(\boldsymbol{y}_1)$, …, $0(\boldsymbol{y}_m)$. Назовем эти операторы *установочными*.

Каждый путь $\tau = v_1, v_2, \ldots, v_l$ программы следующим образом определяет множество функций f_{τ_i}: $f_{\tau_i}(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m) = b'_i \Leftrightarrow$ существует вычисление, соответствующее пути τ, начинающееся состоянием $(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m; v_1)$, и оканчивающееся состоянием $(a_1, a_2, \ldots, a_n, b'_1, b'_2, \ldots, b'_m; v_l)$, $i = 1, 2, \ldots, m$. В свою очередь, каждый полный путь τ определяет единственную функцию f_τ: $f_\tau(a_1, a_2, \ldots, a_n) = b \Leftrightarrow$ существует терминальное вычисление, соответствующее пути τ и начинающееся состоянием $(a_1, a_2, \ldots, a_n, 0, 0, \ldots, 0; v_1)$, для которого b является значением выходной переменной в заключительном состоянии. Тогда функция, вычисляемая программой, представляет собой объединение функций, определяемых всеми ее полными путями.

Пусть $\tau = v_1, v_2, \ldots, v_l, \ldots$ - путь в программе π. Заменим каждый предикат $P(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_r)$ в нем предикатом $P^\sigma(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_r)$, где $\sigma \in \{0, 1\}$ таково, что следующая в τ за $P(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_r)$ вершина является его σ-последователем. Полученную таким образом последовательность назовем *размеченным* путем и обозначим $\boldsymbol{\tau}$.

Программа π может рассматриваться как диаграмма конечного автомата, состояниями которого являются логические и заключительная вершины, а входной алфавит образован из следующих «букв». Если из логической вершины P в логическую вершину P_1 ведет путь $P, v_1, \ldots, v_l, P_1$, где v_1 является σ-последователем вершины P, $v_1, \ldots, v_l$ – арифметические вершины и $l \geq 0$, то переход из состояния P в состояние P_1 происходит под воздействием автоматной «буквы» $P^\sigma, v_1, \ldots, v_l$. Событие, представимое таким автоматом, состоит из всех полных размеченных путей программы. Представим регулярное выражение R, задающее множество всех полных путей, в виде суммы $R_\pi = R_1 \vee R_2 \vee \ldots \vee R_r$ членов, не содержащих операции суммирования.

Покажем, как *всякое регулярное подвыражение R выражения R_i определяет совокупность функций, $i = 1, 2, \ldots, r$.*

1. Пусть R не содержит оператора *. По построению оно задает единственный размеченный путь $\boldsymbol{\tau}$, который определяет m частичных функций $f_{\tau_i}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m)$, $i = 1, 2, \ldots, m$, как описано выше. Будем говорить, что выражение R *определяет* эти функции. Если $\boldsymbol{\tau}$ есть полный путь, то определяемая им единственная функция $f_\tau(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ описывает зависимость выходной переменной программы от входных.

2. Пусть выражение $R = A^*$, регулярное выражение A задает совокупность $\{\boldsymbol{\tau}\}$ размеченных путей и $\{\boldsymbol{\tau}\}^*$ есть совокупность всех размеченных путей, полученных всеми возможными конкатенациями путей из $\{\boldsymbol{\tau}\}$. Каждый из этих путей определяет m частичных функций $f_{\tau_i}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m)$, $i = 1, 2, \ldots, m$, как описано выше. Будем полагать, что выражением R *определяет* совокупность всех этих функций.

3. Пусть выражение $R = AB$, регулярное выражение A определят совокупность F функций, а выражение B – совокупность G. Тогда выражение R *определяет* совокупность всех возможных суперпозиций вида Gf, где $G \in G$ и на места рабочих переменных функции G подставляются функции из F, задающие значения одноименных рабочих переменных и определяемые выражением A.

Из этого построения вытекает

***Лемма* 3.** *Каждое регулярное выражение R_i, $i = 1, 2, \ldots, r$, определяет функцию $\boldsymbol{f}_i$, описывающую зависимость выходной переменной от входных, а регулярное выражение R_π определяет функцию $f_\pi = \cup_{i=1,r} \boldsymbol{f}_i$, вычисляемую программой π.*

Доказательство. Каждое регулярное выражение R_i, $i = 1, 2, \ldots, r$, представляет некоторое множество полных путей программы. Последние, в свою очередь определяют совокупность функций, задающих выходное значение в зависимости от входных. Их объединение есть функция $\boldsymbol{f}_i$. Все регулярное выражение R_π представляет множество всех полных путей программы. Поэтому объединение $\cup_{i=1,r} \boldsymbol{f}_i$ есть функция, вычисляемая программой.

Лемма доказана.

Справедлива

***Лемма* 4.** *Всякое подвыражение* R_i, $i = 1, 2, \ldots, r$ *регулярного выражения* R_π *не может оканчиваться выражением вида* A^*.

Доказательство вытекает из того, что событие, представимое выражением R_π, состоит из всех полных размеченных путей программы π. Если выражение R_i, $i \in \{1, 2, \ldots, r\}$ оканчивается подвыражением A^*, то R_i не определяет терминального пути. Это противоречит способу построения регулярного выражения R_π.

Лемма доказана.

Следствие. *Всякое подвыражение* R_i, $i = 1,2, \ldots, r$ *регулярного выражения* R_π *оканчиваться автоматной буквой вида* $P^\sigma, v_1, \ldots, v_l$, $l \geq 0$, $\sigma \in \{0, 1\}$.

***Лемма* 5.** *Пусть* A^* *есть регулярное подвыражение выражения* R_π, $P_1^{\sigma^1}$ - *предикат, которым начинается первая автоматная буква выражения A, и* $P_q^{\sigma^q}$, $v_1, \ldots, v_l$ - *последняя автоматная буква выражения A. Тогда, если* $l > 0$, *то в программе вершина* v_l *соединена дугой с вершиной* P_1, *если* $l = 0$, *то* σ_q *-дуга, выходящая из вершины* P_q, *ведет в вершину* P_1.

Доказательство вытекает из способа построения регулярного выражения R_π.

***Лемма* 6.** *Пусть* A^*B *есть собственное регулярное подвыражение выражения* R_π *и* $P_1^{\sigma^1}$ - *предикат, которым начинается первая автоматная буква выражения A. Тогда первая автоматная буква выражения B начинается с предиката* $P_1^{1-\sigma^1}$.

Доказательство. Рассмотрим вначале простейший случай, когда выражение A не содержит оператора *. Тогда оно представляет собой последовательность $P_1^{\sigma^1} A_1 P_2^{\sigma^2} A_2 \ldots P_q^{\sigma^q} A_q$ автоматных букв. Выражение $P_1^{\sigma^1} A_1 P_2^{\sigma^2} A_2 \ldots P_q^{\sigma^q} A_q$ определяет размеченный путь τ, который проходит через логические вершины $P_1, P_2, \ldots, P_q$ и содержит дуги, ведущие соответственно в их $\sigma_1, \sigma_2, \ldots, \sigma_q$-последователи.

Так как выражение A^* содержит оператор *, то оно представляет совокупность размеченных путей, которые получаются n-кратным прохождением пути τ, при всяком $n \geq 0$. Поэтому τ есть цикл, начинающийся и заканчивающийся вершиной P_1, которая есть σ_q-последователь вершины P_q.

Покажем, что непосредственно после выражения A^* следует буква входного алфавита, начинающаяся с предиката $P_1^{1-\sigma_1}$. Действительно, после последнего прохождения пути τ следующий путь может начинаться только в логической вершине P_1 и вести в ее $(1-\sigma_1)$-последователя. Поэтому непосредственно правее A^* должно располагаться регулярное выражение, первая автоматная буква которого начинается с предиката $P_1{}^{1-\sigma_1}$.

Пусть теперь выражение A есть последовательность регулярных выражений: $A_1\ A_2 \ldots\ A_q$, причем каждое из них может содержать оператор *. Допустим далее, что выражение A_1 начинается с автоматной буквы, имеющей начальным предикат $P_1{}^{\sigma_1}$. В этом случае выражение $A_1\ A_2 \ldots\ A_q$ определяет совокупность $\{\tau\}$ размеченных путей, которые проходят через логическую вершину P_1 и содержит дугу, ведущую в ее σ_1-последователя.

Выражение A^* задает совокупность размеченных путей, которые получаются n-кратным прохождением путей совокупность $\{\tau\}$, при всяком $n \geq 0$. Поэтому каждый путь совокупности $\{\tau\}$ есть цикл, который начинается и заканчивается вершиной P_1. Тем самым выражение A^* представляет совокупность циклических вычислений программы, и выход из этих циклов возможен только по $(1-\sigma_1)$-дуге вершины P_1. Но это возможно лишь в случае, когда непосредственно после выражения A^* располагается автоматная буква, начинающаяся с предиката $P_1{}^{1-\sigma_1}$.

Рассмотрены все случаи. Лемма доказана.

Следствие. *Программа, которая реализует всюду определенную функцию, определяет регулярное выражение, не содержащее подвыражений вида* $((P^{\sigma}\ A)^*\ P^{1-\sigma}\ B)^*$. (*В данном случае выражение* $(P^{\sigma}\ A)^*\ P^{1-\sigma}\ B$ *следует понимать так*: *заключенное в скобках регулярное выражение* $P^{\sigma}\ A$ *имеет первой автоматную*

букву, которая начинается с предиката P^{σ}, а непосредственно после выражения $(P^{\sigma}\ A)^*$ следует выражение, первая автоматная буква которого начинается с предиката $P^{1-\sigma}$.)

Доказательство. Действительно, выражение $((P^{\sigma}\ A)^*\ P^{1-\sigma}\ B)^*$ определяет бесконечный цикл, проходящий через логическую вершину P. Если путь начинается в P и проходит через ее σ-последователь, то он включает размеченный путь, определяемый выражением $P^{\sigma}\ A$ и возвращается вновь в логическую вершину P. Если же путь начинается в P и проходит через ее (1-σ)-последователь, то он включает размеченный путь, определяемый выражением $P^{1-\sigma}\ B$ и возвращается вновь в логическую вершину P. Противоречие с тем, что программа реализует всюду определенную функцию.

Следствие доказано.

Исходя из этого Следствия и того, что мы рассматриваем программы без фиктивных вершин, мы ограничимся лишь такими программами, для которых регулярные выражения не содержат подвыражений вида $((P^{\sigma}\ A)^*\ P^{1-\sigma}\ B)^*$.

В последующем будем полагать, что число шагов, осуществляемое программой π при любом вычислении, ограничено некоторой конечной величиной, определяемой значениями входных переменных. В частности ограничимся рассмотрением вычислений программы над некоторым множеством входных значений, длина которых ограничена конечным числом p.

Исходя из этого, по выражению R_{π} построим так называемое *параметрическое выражение* R_{π}'. Для этого представим каждое регулярное выражение R_i в параметрическом виде R'_i так, что при указанном ограничении на вычисления программы параметрическое выражение R_{π}' задает то же множество полных путей, что и исходное R_{π}. Выражение R'_i строим индукцией по выражению R_i. Для этого введем понятие *текущего параметра*, которое определяется следующим образом.

0. Полагаем, что текущий параметр каждого выражения R_i равен p, i = 1, 2, …, m. После этого просматриваем каждое выражение R_i слева направо, анализируя его подвыражения и определяя текущий параметр для каждого из них.

1. Если рассматриваемый оператор арифметический a_i или логический P_j, то записываем его в виде соответственно $(a_i)^1$или $(P_j)^1$, одновременно уменьшая текущий параметр на 1.

2. Если рассматриваемый оператор A^*, то ставим ему в соответствие выражение $(A)^r$, где r –текущий параметр. При этом текущий параметр не меняется.

3. После очередного просмотра выражений R_i, i = 1, 2, …, m, мы переходим к рассмотрению выражений вида $(A)^r$, где A - его собственное регулярное подвыражение, для которого текущим параметром является r.

4. Разбор выражения R_i, i = 1, 2, …, m, происходит до тех пор, пока не будут просмотрены все арифметические и логические операторы.

Сформулируем некоторые свойства параметрических выражений. Справедливы следующие леммы.

***Лемма* 7.** *Параметрическое выражение R_π' определяет множество всех полных путей программы π при введенном ограничении p на число шагов вычисления.*

Доказательство. Число вхождений каждого слова в событие, представимое регулярным выражением R_π', совпадает с числом вхождений каждого слова в событие, представимое исходным регулярным выражением. Это вытекает из того, что мы рассматриваем множество входов программы, для которых число шагов программы ограничено значением p.

Лемма доказана.

Тем самым, параметрическое выражение R_π' для рассматриваемого множества входов определяет ту же функцию, что и регулярное выражение R_π.

Теперь из леммы 5 следует

***Лемма* 8.** *Если в регулярном выражении* R_π *встречается выражение* $(A\ B^*\ C)^*$, *а в параметрическом* R_π' *ему соответствует выражение* $(A\ B^{m_1}\ C)^{m_2}$, *то* $m_1 < m_2$.

3. Представление программ обобщенными формулами. Расширим булевский язык над обычным базисом $\{\vee, \wedge, \neg\}$ путем введения обобщений дизъюнкции и конъюнкции для не более, чем счетного множества аргументов. Для обобщенных дизъюнкции и конъюнкции будем использовать выражения соответственно $\cup_{i=k,\ h}$ и $\cap_{i=k,\ h}$, где i называется *индексом* этой связки, k- *нижней границей*, а h – *верхней*. k и h – это либо константы, либо символ бесконечности - ω, либо функции, зависящие от других индексов.

Для дальнейшего нам потребуется следующее соглашение. Всякой двоичной переменной $\boldsymbol{y}$ программы будем сопоставлять счетную последовательность $y_0y_1\ldots y_i\ldots$ логических переменных. Когда $\boldsymbol{y}$ принимает двоичное значение, то i-ая логическая переменная означивается i-м разрядом этого числа, $i = 0, 1, \ldots$. Напомним, что мы изображаем двоичные числа, младшие разряды которых располагаются левее. Таким образом, логические переменные $y_0y_1\ldots y_i\ldots$ будут имеет слева некоторый конечный отрезок переменных, означенных в соответствии с двоичным числом, справа от которого будет располагать бесконечная последовательность переменных, означенных нулями.

Будем говорить, что логическая формула $F(\boldsymbol{x}, \boldsymbol{y})$, где $\boldsymbol{x} = x_0x_1\ \ldots\ x_i\ldots$, $\boldsymbol{y} = y_0y_1\ \ldots\ y_i\ldots$ суть последовательности логических переменных, *представляет* двоичную функцию $\boldsymbol{y} = \varphi(\boldsymbol{x})$, если при означивании переменных $\boldsymbol{x}$ бинарным вектором σ_x таким, что $\sigma_x \in \mathrm{Def}\ \varphi$, формула $F(\sigma_x, \sigma_y)$ истинна тогда и только тогда, когда $\varphi(\sigma_x) = \sigma_y$.

***Пример* 1.** Обозначим $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}$ наборы двоичных переменных соответственно: $x_0, x_1, \ldots, x_n, \ldots$; $y_0, y_1, \ldots, y_n, \ldots$ и $u_0, u_1, \ldots, u_n, \ldots$. Нетрудно увидеть, что формула

$$S(\boldsymbol{x}, \boldsymbol{y}): (y_0 \equiv \overline{x_0}) \wedge \cap_{i=1,\,\omega} (y_i \equiv x_i + \cap_{j=0,\,i-1} x_i)$$

представляет функцию $s(\boldsymbol{x})$ прибавления единицы двоичной арифметики. Подформула $\cap_{j=0,\,i-1} x_i$ представляет перенос в i-ый разряд, он равен 1 тогда и только тогда, когда все разряды до $(i - 1)$-го включительно равны 1.

В точности так же и формула

$$S(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}): u_0 \equiv x_0 \wedge \cap_{i=0,\omega} (u_{i+1} \equiv u_i \wedge x_i) \wedge (y_0 \equiv \overline{x_0}) \wedge \cap_{i=1,\omega} (y_i \equiv x_i + u_i)$$

представляет функцию прибавления единицы. Легко увидеть, что новая переменная u_i определяет перенос в i-ый разряд, который вычисляется после прибавления 1 к вектору с разрядами от нулевого до $(i - 1)$-го включительно. Перенос в нулевой разряд совпадает с нулевым разрядом самого числа.

Базисная функция $0(\boldsymbol{x})$ представима формулой $\cap_{i=0,\omega} \overline{x_i}$.

Предикат $\boldsymbol{x} = \boldsymbol{y}$ представим формулой $E(\boldsymbol{x}, \boldsymbol{y}) = \cap_{i=0,\omega} (x_i \equiv y_i)$.

Предикат $\boldsymbol{x} \neq \boldsymbol{y}$ представим формулой $\overline{E}\,(\boldsymbol{x}, \boldsymbol{y}) = \cup_{i=0,\omega} \neg (x_i \equiv y_i)$.

***Теорема* 9.** *Пусть формулы* $F(\boldsymbol{x}, \boldsymbol{y})$ *и* $G(\boldsymbol{y}, \boldsymbol{z})$ *представляют функции соответственно* $\boldsymbol{y} = f(\boldsymbol{x})$ *и* $\boldsymbol{z} = G(\boldsymbol{y})$ *двоичной арифметики. Тогда конъюнкция* $F(\boldsymbol{x}, \boldsymbol{y}) \wedge G(\boldsymbol{y}, \boldsymbol{z})$ *представляет суперпозицию* $\boldsymbol{z} = f(G(\boldsymbol{y}))$.

Доказательство. Пусть бинарные векторы $\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y, \boldsymbol{\sigma}_z$ предствляют двоичные числа, такие, что $\boldsymbol{\sigma}_y = f(\boldsymbol{\sigma}_x)$ *и* $\boldsymbol{\sigma}_z = G(\boldsymbol{\sigma}_y)$. Для тех же бинарных векторов логические функции $F(\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y)$ и $G(\boldsymbol{\sigma}_y, \boldsymbol{\sigma}_z)$ истинны. Но тогда истинна и конъюнкция $F(\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y) \wedge G(\boldsymbol{\sigma}_y, \boldsymbol{\sigma}_z)$.

В обратную сторону. Пусть для бинарных векторов $\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_z$ предствляющих двоичные числа, имеет место равенство $\boldsymbol{\sigma}_z = G(f(\boldsymbol{\sigma}_x))$. Из того, что f есть функция двоичной арифметики следует, что существует единственный бинарный вектор $\boldsymbol{\sigma}_y$, представляющий двоичное число, для которого имеет место равенство $\boldsymbol{\sigma}_y$

$= f(\sigma_x)$. Но тогда конъюнкция $F(\sigma_x, \sigma_y) \wedge G(\sigma_y, \sigma_z)$ истинна в силу того, что формулы $F(\boldsymbol{x}, \boldsymbol{y})$ и $G(\boldsymbol{y}, \boldsymbol{z})$ представляют функции соответственно $\boldsymbol{y} = f(\boldsymbol{x})$ и $\boldsymbol{z} = G(\boldsymbol{y})$.

Разобраны все случаи. Теорема доказана.

***Пример* 2.** Используя последнюю теорему о суперпозиции функций, легко показать, что при фиксированном целом $l > 0$ функция $\boldsymbol{y} = \boldsymbol{x} + l$ представима обобщенной формулой (обозначим ее $S^l(\boldsymbol{x}, \boldsymbol{y})$). Приведем ряд формул, представляющих арифметические функции прибавления некоторых констант.

Арифметические функции	Представляющие формулы обобщенной логики
$\boldsymbol{y} = \boldsymbol{x} + 2$	$y_0 \equiv x_0 \wedge y_1 \equiv \overline{x}_1 \wedge \cap_{i=2,\omega}\, y_i \equiv x_i + \cap_{j=1,\, i\text{-}1}\, x_j$
$\boldsymbol{y} = \boldsymbol{x} + 3$	$y_0 \equiv \overline{x}_0 \wedge y_1 \equiv x_0 + \overline{x}_1 \wedge \cap_{i=2,\omega}\, y_i \equiv x_i + (x_1 + x_0\ \overline{x}_1) \cap_{j=2,\, i\text{-}1}\, x_j$

Не всякая двоичная арифметическая функция представима обобщенной формулой. В частности, не представимо умножение.

Базисные операторы программ представимы обобщенными формулами. Поэтому возникает вопрос о представлении функции, вычисляемой программой π, обобщенной логической формулой. Исходя из этого, построим по параметрическому выражению R_π' обобщенную формулу, определив вначале соответствующую формулу для его каждого дизъюнктивного члена, а затем дизъюнктивно объединив их. Поэтому наши построения будут касаться каждого параметрического выражения R'_i, $i = 1, 2, \ldots, m$.

Во-первых, несколько преобразуем каждое выражение R'_i, сделав его более удобным для последующего построения. Если в выражении R'_i оператор $s(\boldsymbol{y})$ встречается подряд l раз, где $\boldsymbol{y}$ – рабочая переменная, то заменим его на оператор $s^l(\boldsymbol{y})$. Аналогично, если в выражении R'_i оператор $0(\boldsymbol{y})$ встречается подряд l раз, где $\boldsymbol{y}$ – рабочая переменная, то заменим эту последовательность одним оператором $0(\boldsymbol{y})$.

Дальнейшее построение логической формулы заключается в замене арифметических и логических операторов, из которых состоят автоматные буквы регулярного выражения, на представляющие их обобщенные формулы. В качестве

переменных используемых при этом логических формул будем использовать так называемые *метапеременные*. С этой целью вначале заменим все рабочие переменные $y_1, y_2, \ldots, y_m$ в параметрическом выражении на метапеременные соответственно $\alpha_1, \alpha_2, \ldots, \alpha_m$. Считаем, что эти метапеременные порождены рабочими переменными и поэтому назовем их *рабочими*. Входные переменные мы не заменяем, для единообразия считая их также метапеременными, которые назовем *входными*. Если необходимо подчеркнуть, что метапеременная есть входная переменная, это будет оговорено специально.

Отметим следующее.

Регулярное выражение R_π включает в качестве компонентов арифметические и логические операторы, содержащие переменные, областью определения которых являются не отрицательные целые числа.

В точности так же, как мы представляем двоичные переменные счетными последовательностями булевских переменных, считаем, что каждая метапеременная $\boldsymbol{\alpha}$ представима в виде последовательности $\alpha_1\alpha_2 \ldots \alpha_i \ldots$ логических метапеременных.

Дальнейшее преобразование параметрического выражения приводит к появлению логических формул, которые имеют в качестве аргументов метапеременные.

Введем следующие соглашения.

1. Пусть $\boldsymbol{\alpha}$ есть метапеременная и ее булевское представление выглядит так: $\alpha_0\alpha_1\alpha_2 \ldots \alpha_m \ldots$. Тогда $0(\alpha)$обозначим обобщенную формулу $\cap_{i=0,\omega}\ \bar{\alpha}_i$. Очевидно, что формула $\cap_{i=0,\omega}\ \bar{\alpha}_i$ представляет арифметический оператор $0(\boldsymbol{\alpha})$. В последующем, если это необходимо, будем оговаривать особо, когда $0(\boldsymbol{\alpha})$ понимается как арифметический оператор, а когда как обобщенная формула.

2. Если $\boldsymbol{\alpha} = \alpha_0\alpha_1\alpha_2 \ldots \alpha_i \ldots$., $\boldsymbol{\beta} = \beta_0\beta_1\beta_2 \ldots \beta_i \ldots$. - это метапеременные, то обозначим $E(\boldsymbol{\alpha}, \boldsymbol{\beta})$ бесконечную конъюнкцию $\cap_{i=0,\,\omega}\ \alpha_i \equiv \beta_i$, а $\bar{E}\,(\boldsymbol{\alpha}, \boldsymbol{\beta})$ - дизъюнкцию $\cup_{i=0,\,\omega}\ \neg(\alpha_i \equiv \beta_i)$.

Просматриваем каждое параметрическое выражение R'_i, $i = 1, 2, \ldots, m$ слева направо. При первом просмотре мы игнорируем параметры, которыми был заменен оператор *. В зависимости от вида очередного подвыражения осуществляем следующие действия.

Пусть таким выражением является автоматная буква P^σ, $v_1, \ldots, v_l$. В этом случае анализируем по очереди вид логического и арифметических операторов и ставим им в соответствие определенные логические формулы.

1. Предикат $P^\sigma(\alpha, \beta)$ заменяем формулой $E^\sigma(\alpha, \beta)$.

2. Оператор $s^l(\alpha)$ заменяем формулой $S^l(\alpha, \beta)$, где β - это новая рабочая метапеременная, которая не использовалась ранее. Затем везде в этом выражении правее оператора $s^l(\alpha)$ заменяем все вхождения метапеременной α на β. Назовем α – *входной* метапеременной формулы $S^l(\alpha, \beta)$, а β – *выходной*. При этом β назовем *последователем* α, а α - *предшественником* β.

3. Всякий установочный оператор $0(\alpha)$ заменяем формулой $0(\alpha)$. Если $0(\alpha)$ не установочный арифметический оператор, то заменяем его на формулу $0(\beta)$, где β - это новая рабочая метапеременная, которая не использовалась ранее. Затем везде правее этого оператора заменяем все вхождения метапеременной α на β.

Замечание. Арифметический оператор $0(\boldsymbol{y})$ в программе устанавливает нулевое значение рабочей переменной $\boldsymbol{y}$ не зависимо от ее предыдущего значения. Прежнее значение этой переменной как бы забывается и его восстановление невозможно. Поэтому не установочный оператор $0(\alpha)$ заменяем на формулу $0(\beta)$, где β - новая рабочая метапеременная. Последняя также считается порожденной той же рабочей переменной, которая порождала метапеременную α. При этом β назовем *последователем* α, которую в свою очередь назовем *предшественником* β.

Полагаем, что оба отношения «быть последователем» и «быть предшественником», определенные в двух последних пунктах, – транзитивные.

Из определения отношения «быть последователем» вытекает, что если метапеременная β есть последователь α, то в выражении R_π' они порождены одной рабочей переменной. То же относится к отношению «быть предшественником».

Введем следующее определение. Пусть в результирующем выражении R', полученном из исходного, метапеременная β есть последователь метапеременной α. Назовем β - *тупиковым последователем* метапеременной α в R', если β не имеет в нем последователя. Аналогично, α назовем *тупиковым предшественником* метапеременной β в R', если α не имеет предшественника в R'. Напомним, что R' есть подобие регулярного выражения, в котором арифметические и логические операторы заменены представляющими их обобщенными формулами.

В результате описанных преобразований параметрическое выражение R_π' преобразуется в конструкцию R_π'', которая вместо арифметических и логических операторов содержит представляющие их логические формулы, а вместо входных и рабочих переменных – соответственно рабочие и входные метапеременные. Каждая формула S^l обладает входными и выходными метапеременными, между которыми установлены отношения последовательности и предшествования. При этом входная метапеременная формулы S^l не встречается правее нее. Несколько раз могут встретиться лишь входные переменные программы и метапеременные, которые встречаются в формулах, представляющих логические операторы программы.

Пусть P^σ, v_1, …, v_l – это автоматная буква рассматриваемого параметрического выражения, которая задает размеченный путь τ. Этот путь, в свою очередь, определяет совокупность $f_{\tau i}$, i = 1, 2, ..., m частичных функций, каждая из которых определяет значение одной рабочей переменной в зависимости от значений переменных перед выполнением логического оператора P. Пусть рабочие переменные $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, …, $\boldsymbol{y}_m$ этой автоматной буквы порождают некоторое множество метапеременных, из которого выделим тупиковые предшественники α_1, α_2, …, α_m и соответственно тупиковые последователи β_1, β_2, …, β_m относительно буквы P^σ, v_1, …, v_l.

В соответствии с первым этапом нашего построения в результирующем выражении R_π'' отдельным компонентам буквы P^σ, v_1, …, v_l соответствуют представляющие их формулы $E^\sigma(\alpha, \beta)$, W_1, ..., W_l. Поставим в соответствие автоматной букве P^σ, v_1, …, v_l формулу

$$F = E^\sigma(\alpha,\beta) \wedge W_1 \wedge \ldots \wedge W_l.$$

Справедливо утверждение.

***Лемма* 10.** *Формула F представляет все функции, определяемые автоматной буквой P^σ, v_1, …, v_l с точностью до переименования рабочих переменных порожденными ими метапеременными.*

Доказательство. Напомним, что если логическая функция $F(\boldsymbol{x}, \boldsymbol{y})$ представляет вычисляемую функцию $\boldsymbol{y} = f(\boldsymbol{x})$, функция $G(\boldsymbol{z}, \boldsymbol{w})$ - функцию $\boldsymbol{w} = G(\boldsymbol{z})$, то суперпозиция функций $\boldsymbol{y} = f(G(\boldsymbol{x}))$ представляется конъюнкций $F(\boldsymbol{w}, \boldsymbol{y}) \wedge G(\boldsymbol{x}, \boldsymbol{w})$.

Если при выполнении программы π предикат P^σ принимает значение *истина*, то значения рабочих переменных определяются этим путем следующим образом. Если рабочая переменная y встречается в операторах v_1, …, v_l и не обнуляется, то ее новое значение есть сумма ее входного значения и некоторой константы, определяемой присутствующими в этом пути операторами прибавления единицы. Если же эта рабочая переменная $\boldsymbol{y}$ обнуляется, то ее значение равно некоторой константе, которая совпадает с числом единиц, прибавляемых к этой переменной после последнего ее обнуления в рассматриваемом пути. Но в точности так же вычисляется значение соответствующей тупиковой метапеременной β функцией, представимой конъюнкцией

$$E^\sigma(\alpha, \beta)\ W_1\ \ldots\ W_l.$$

Лемма доказана.

Таким образом, для автоматной буквы P^σ, v_1, …, v_l , которая входит в исследуемое параметрическое выражение, верны следующие утверждения:

буква P^σ, v_1, …, v_l определяет совокупность $f_{\tau i}$, $i = 1, 2, \ldots, m$, частичных функций, задающих значения рабочих переменных;

все эти функции представляются формулой F при введенном соответствии рабочих переменных и метапеременных.

Для подслова $a_1a_2\ldots a_q$ рассматриваемого параметрического выражения, которое состоит из нескольких автоматных букв, имеет место

***Лемма* 11.** *Пусть* $F_{a_1}, F_{a_2}, \ldots, F_{a_q}$ *– формулы, поставленные в соответствие автоматным буквам соответственно* $a_1, a_2, \ldots, a_q$. *Тогда формула* $F = F_{a_1} \wedge F_{a_2} \wedge \ldots \wedge F_{a_q}$ *представляет все частичные функции, которые определяются регулярным выражением* $a_1a_2\ldots a_q$.

Доказательство вытекает из того, что тупиковые выходные метапеременные каждого выражения, определяемого буквой a_i, $i = 1, 2, \ldots, q$ -1, совпадает с соответствующими тупиковыми входными метапеременными выражения, определяемого буквой a_{i+1}. Тем самым, если буква a_i определяет совокупность из m частичных функций (по числу рабочих переменных), то слово $a_1, a_2, \ldots, a_q$ определяет частичные функции, полученные определенной суперпозицией из указанных, а формула F представляет все эти функции.

Лемма доказана.

Дальнейшее преобразование параметрического выражения в логическую формулу носит индуктивный характер. Из описанных ранее свойств параметрического выражения R_π' вытекает, что каждое его подвыражение вида B^h можно представить следующим образом: $(P^\sigma A)^h$, где P есть логический оператор программы и A – оставшаяся часть выражения B. В простейшем случае выражение A является последовательностью арифметических операторов.

<u>*Базис индукции*</u>. Пусть выражение B не имеет вложенных параметрических выражений, построенных по регулярным выражениям вида Q^*. Поэтому оно определяет единственный размеченный путь τ, начинающийся в логической вершине P и включающий ее σ–последователя. Будем полагать, что выражение $P^\sigma A$ уже представляет формула F.

Пусть α и β– это последовательности соответственно тупиковых входных и выходных метапеременных выражения B. Поэтому формулу F обозначим сле-

дующим образом: $E^{\sigma}(\alpha) \wedge F(\alpha, \beta)$. Последовательность α входных метапеременных может состоять из входных переменных программы и метапеременных, которые порождены рабочими переменными. Следовательно, длина последовательности α не превосходит $n + m$. Представим α в виде объединения двух последовательностей α^w и α^I. Здесь α^w есть метапеременные, полученные по рабочим переменным, а α^I - входные переменные.

Последовательность β выходных метапеременных может состоять только из метапеременных, которыми были заменены рабочие переменные, и ее размерность не превосходит m. Для простоты полагаем, что последовательность α имеет $n + m$ метапеременных, включает входные переменные и метапеременные, полученные по рабочим переменным, а последовательность β состоит только из m метапеременных, полученных по рабочим переменным. Тогда отношения последовательности и предшествования для этих последовательностях устанавливаются просто. Полагаем, что $\alpha^w = \alpha_1, \alpha_2, \ldots, \alpha_m$ и $\beta = \beta_1, \beta_2, \ldots, \beta_m$.

Образуем h различных множеств метапеременных: $\beta^1, \beta^2, \ldots, \beta^h$, каждое мощности m и между каждой парой множеств $\beta, \beta^1, \ldots, \beta^h$ определено изоморфное соответствие φ. Назовем *соответствующими* следующие метапеременные: в множествах α и β – это метапеременные между которыми установлено отношение быть последователем, а между каждой парой множеств из $\beta, \beta^1, \ldots, \beta^h$ соответствие определяется отображением φ. Таким образом, для каждой входной метапеременной из множества α в каждом множестве из $\beta, \beta^1, \ldots, \beta^h$ можно выделить единственную метапеременную, которую также назовем *выходной*. И таких метапеременных в каждом из множеств $\beta, \beta^1, \ldots, \beta^h$ в точности m. Поэтому $\beta^i = \beta^i_1, \beta^i_2, \ldots, \beta^i_m$.

Итак, для каждой входной метапеременной из α, полученной по рабочей переменной, в любом множестве β^i всегда имеется в точности одна соответствующая выходная метапеременная. Обозначим $\alpha^1, \alpha^2, \ldots, \alpha^h$ множества метапеременных, которые получены добавлением к входным переменным множеств соответственно $\beta^1, \beta^2, \ldots, \beta^h$. Полагаем, что множества α и $\alpha^1, \alpha^2, \ldots, \alpha^h$ упорядоче-

ны так, что сохраняется одинаковый порядок на соответствующих метапеременных.

Введем такое соглашение. Пусть выражение $\alpha^w = \beta$ обозначает m конъюнкций $\cap_{i=1,\,m} E(\alpha_i^w, \beta_i)$, где $E(\alpha_i^w, \beta_i)$ – обобщенная формула, представляющая равенство $\alpha_i^w = \beta_i$ метапеременных. Аналогично $\beta^j = \beta$ обозначает конъюнкцию $\cap_{i=1,\,m} E(\beta_i^j, \beta_i)$, где $E(\beta_i^j, \beta_i)$ обобщенная формула, представляющая равенство $\beta_i^j = \beta_i$, $i = 1, 2, \ldots, m$.

Теперь выражению B^h ставим в соответствие логическую формулу

$$F^h = E^{1-\sigma}(\alpha)(\alpha^w = \beta) \vee E^{\sigma}(\alpha)\, F(\alpha, \beta^1)\, E^{1-\sigma}(\alpha^1)\,(\beta^1 = \beta) \vee$$

$$E^{\sigma}(\alpha)\;\; F(\alpha, \beta^1)\, E^{\sigma}(\alpha^1)\, F(\alpha^1, \beta^2)\, E^{1-\sigma}(\alpha^2)\,(\beta^2 = \beta) \vee$$

$$\ldots$$

$$E^{\sigma}(\alpha)F(\alpha, \beta^1)\, E^{\sigma}(\alpha^1)\, F(\alpha^1, \beta^2) \ldots E^{\sigma}(\alpha^{h-1})\, F(\alpha^{h-1}, \beta^h)\, E^{1-\sigma}(\alpha^h)(\beta^h = \beta).$$

В этой формуле $E^{\sigma}(\alpha^i)$ есть формула, представляющая предикат $P^{\sigma}(\alpha^i)$, в которой вместо метапеременных из α подставлены соответствующие метапеременные из α^i, $i = 1, 2, \ldots, h$.

Покажем, что верна

***Лемма* 12.** *Формула F^h представляет все функции, определяемые параметрическим подвыражением B^h.*

Доказательство. Во-первых, покажем, что каждый дизъюнктивный член этой формулы представляет определенное множество функций, объединение которых совпадает с семейством функций, определяемых выражением B^h.

1. Конъюнкция $E^{1-\sigma}(\alpha) \wedge (\alpha = \beta)$ истинна тогда и только тогда, когда формула $E^{1-\sigma}(\alpha)$ истинна и выходные переменные набора β совпадают с соответствующими входными набора α. Среди функций, определяемых выражением B^h, имеются функции, которые порождены нулевым прохождением пути τ, что происходит, когда значения аргументов предиката P^{σ} превращают его в *ложь*. В этом случае рабочие переменные после перехода в (1-σ)-последователя вершины P не меняются. Следовательно такой переход по дуге из вершины P в 1 - σ по-

следователя соответствует тождественной функции. Но именно такую функцию представляет рассматриваемая конъюнкция.

2. Рассмотрим теперь i-кратное прохождение ($0 < i \le h$) размеченного пути τ, завершающееся прохождением дуги δ, ведущей в (1-σ)-последователя вершины P. Такой путь порождает частичные функции, которые суть суперпозиции глубины i функций, порождаемых самим путем τ. При этом последний переход по дуге, ведущее в (1-σ)-последователя вершины P не меняет значений рабочих переменных, которые были получены после последнего прохождения вершины P.

Отсюда видно, что конъюнкция

$$E^{\sigma}(\alpha) \wedge F(\alpha, \beta^1) \wedge E^{\sigma}(\alpha^1) \wedge F(\alpha^1, \beta^2) \wedge \ldots \wedge E^{\sigma}(\alpha^{i-1}) \wedge F(\alpha^{i-1}, \beta^i) \wedge E^{1-\sigma}(\alpha^i)\ (\beta^i = \beta)$$

представляет все функции, определяемые i-кратным прохождением размеченного пути τ, завершающееся прохождением дуги δ, ведущей в (1-σ)-последователя вершины P. Действительно, всякая формула

$$E^{\sigma}(\alpha) \wedge F(\alpha, \beta^1) \wedge E^{\sigma}(\alpha^1) \wedge F(\alpha^1, \beta^2) \wedge \ldots \wedge E^{\sigma}(\alpha^j) \wedge F(\alpha^j, \beta^{j+1})$$

представляет все функции, определяемые j-кратным прохождением пути τ, в результате заданным способом совмещения входных и выходных метапеременных. После j-кратного прохождения пути τ выход из цикла происходит лишь в том случае, когда предикат $P^{1-\sigma}$ *истинен*. Этот предикат представляется формулой $E^{1-\sigma}(\alpha^i)$. Поэтому выходные метапеременные из β принимают те значения, которые принимают соответствующие метапеременные набора β^i.

Вся построенная дизъюнкция представляет объединение функций, которые представляются каждым из ее членов. Отсюда вытекает искомое утверждение.

Индуктивные рассуждения. Если параметрическое выражение $A = A_1A_2$, и для выражений A_1 и A_2 построены логические функции соответственно F_1 и F_2, представляющие все функции порождаемые выражениями A_1 и A_2, то конъюнкция $F_1 \wedge F_2$ представляет все функции, порождаемые выражением A. Это вытекает из того, что конструируемая формула имеет в заключительных частях своих конъюнкций метапеременные, совпадающие с тупиковыми выходными метапе-

ременными выражений, по которым они строятся. Для выражения $B = A_1 A_2 \ldots A_n$ рассуждения аналогичны.

Если выражение B имеет вложенные выражения, полученных из регулярных выражений вида Q^*, то B определяет конечное семейство $\{\tau\}$ размеченных путей, начинающихся в логической вершине P и включающих ее σ–последователя. Но тогда выражение B^h определяет все функции, которые определяются размеченными путями, полученными из $\{\tau\}$ не более, чем h-кратными конкатенациями. В этом случае в силу конечности числа путей, участвующих в конкатенациях, наличие представляющей логической формулы для функций, порождаемых выражением B^h, следует из предыдущего абзаца индуктивных рассуждений.

Лемма доказана.

Из этих рассуждений вытекает, что при любом означивании метапеременных α формулы F^h истинным является в точности один из дизъюнктивных членов, в точности при одном означивании оставшихся метапеременных. Таким образом, все значения метапеременной α формулы F^h разбиваются на $h + 1$ классов эквивалентности.

3. О некоторых свойствах формул, порожденных программами. Пусть формула G представляет параметрическое выражение R'_π и программа π обладает ограничением $p(|\boldsymbol{x}|)$ на длину вычисления в зависимости от входных значений. Величину $p(|\boldsymbol{x}|)$ назовем длиной формулы G. Нашей задачей будет описание зависимости минимальной длины вычислений программы и, следовательно, длины формулы G от вида функции, которую вычисляет программа.

Рассмотрим свойства означиваний переменных логической формулы G. Для простоты считаем, что программа обладает единственной входной переменной, т.е. вычисляет функцию от одного аргумента. При этом полагаем, что означен начальный отрезок входных переменных, а метапеременные, порожденные рабочими переменными, означиваются следующим образом.

1. Если в формуле G встречается формула $0(\boldsymbol{\alpha})$, где $\boldsymbol{\alpha} = \alpha_0, \ldots, \alpha_m \ldots$, то $\alpha_0 = \ldots = \alpha_m = \ldots = 0$. Очевидно, это единственное означивание логических метапеременных, при котором формула $0(\boldsymbol{\alpha})$ истинная.

2. Если в формуле G встречается формула $S^l(\boldsymbol{\alpha}, \boldsymbol{\beta})$, $\alpha = \alpha_0, \ldots, \alpha_m, \ldots, \boldsymbol{\beta} = \beta_0, \ldots, \beta_m, \ldots$, и логические метапеременные $\alpha_0, \ldots, \alpha_m, \ldots$ уже означены, то означивание логических метапеременных $\beta_0, \ldots, \beta_m, \ldots$ получается из двоичного представления числа $|\boldsymbol{\alpha}| + l$. Здесь $|\boldsymbol{\alpha}|$ - это двоичное число, полученное из бесконечной последовательности значений метапеременных $\alpha_0, \ldots, \alpha_m, \ldots$ отбрасыванием правой бесконечной нулевой части, расположенной правее последнего единичного компонента. Очевидно, что при таком означивании метапеременных $\boldsymbol{\alpha}, \boldsymbol{\beta}$ формула S^l истинная. При любом другом означивании логических метапеременных $\beta_0, \ldots, \beta_m, \ldots$ она ложна.

Из описанных в п.п. 1, 2 означиваний метапеременных следует, что все Рабочие метапеременные означиваются единственным образом, если мы накладываем условие, что все формулы, представляющие арифметические операторы, должны быть истинными. Назовем такое означивание метапеременных, порожденных рабочими переменными программы, *правильным*.

3. Отметим, что если в формуле $E^\sigma(\boldsymbol{\alpha}, \boldsymbol{\beta})$ обе метапеременные $\boldsymbol{\alpha}$ и $\boldsymbol{\beta}$ рабочие, то при правильном означивании она превращается в константу (0 или 1). Это следует из того, что все рабочие метапеременные означены конкретными значениями.

Из этого следует, что при правильном означивании формула $E^\sigma(\boldsymbol{\alpha}, \boldsymbol{\beta})$ отлична от константы лишь в случае, когда один из ее аргументов является входной метапеременной.

***Лемма* 13**. *Всякая формула F, которая соответствует регулярному подвыражению и не содержит входных метапеременных, при правильном означивании принимает значение истина. При не правильном она ложна.*

Проведем исследование отношения эквивалентности, которое порождается формулой E при различных начальных означиваниях ее аргументов.

Пусть $E(\alpha, \beta) = \cap_{i=0,\,\omega}\alpha_i \equiv \beta_i$, σ_α и σ_β - начальные означивания метапеременных соответственно α и β. Для определенности полагаем, что $|\sigma_\alpha| \le |\sigma_\beta|$, т.е. $\sigma_\beta = \sigma'_\beta\sigma''_\beta$ и $|\sigma_\alpha| = |\sigma'_\beta|$. Рассмотрим возможные случаи.

1. $\sigma_\alpha = \sigma'_\beta$. После означивания наборами σ_α и σ_β метапеременных формула $E(\alpha, \beta)$ эквивалентно преобразуется в формулу, у которой начальный отрезок метапеременных β означен значениями набора σ''_β. Обозначим ее через $E(\alpha, \sigma''_\beta)$.

2. $\sigma_\alpha \ne \sigma'_\beta$. После означивания наборами σ_α и σ_β метапеременных формула $E(\alpha, \beta)$ преобразуется в тождественно ложную формулу.

Таким образом, при начальных означиваниях σ_α и σ_β аргументов α и β выполняется следующее:

если набор σ_α является префиксом набора σ_β, то формула $E(\alpha, \beta)$ превращается в новую формулу, не являющуюся тождественной константой. При этом результирующая формула определяется длиной вектора σ_α и видом вектора σ''_β;

если набор σ_α не является префиксом набора σ_β, то формула $E(\alpha, \beta)$ превращается в тождественную ложь.

Для отрицания $\overline{E}(\alpha, \beta) = \cup_{i=0,\,\omega}\neg(\alpha_i \equiv \beta_i)$ и для начальных означиваний σ_α и σ_β таких, что $|\sigma_\alpha| \le |\sigma_\beta|$, т.е. $\sigma_\beta = \sigma'_\beta\sigma''_\beta$ и $|\sigma_\alpha| = |\sigma'_\beta|$, аналогично доказывается следующее.

Если набор σ_α является префиксом набора σ_β, то формула $\overline{E}(\alpha, \beta)$ превращается в новую формулу, не являющуюся тождественной константой. Ее вид определяется длиной вектора σ_α и видом вектора σ''_β;

Если набор σ_α не является префиксом набора σ_β, то формула $\overline{E}(\alpha, \beta)$ превращается в тождественную истину.

Рассмотрим функцию

$$F^h = E^{1-\sigma}(\alpha)\,(\alpha^w = \beta) \vee E^{\sigma}(\alpha)\,F(\alpha, \beta^1)\,E^{1-\sigma}(\alpha^1)\,(\beta^1 = \beta) \vee$$
$$E^{\sigma}(\alpha)\,F(\alpha, \beta^1)\,E^{\sigma}(\alpha^1)\,F(\alpha^1, \beta^2)\,E^{1-\sigma}(\alpha^2)\,(\beta^2 = \beta) \vee$$
$$\ldots$$

$$E^{\sigma}(\alpha)\, F(\alpha, \beta^1)\, E^{\sigma}(\alpha^1)\, F(\alpha^1, \beta^2)\, \ldots E^{\sigma}(\alpha^{h-1})\, F(\alpha^{h-1}, \beta^h)\, E^{1-\sigma}(\alpha^h)\, (\beta^h = \beta).$$

Исследуем свойства означиваний ее метапеременных. Особенность этой формулы состоит в том, что ее подформула F может быть конъюнкцией логических формул, представляющих лишь арифметические операторы, так и дизъюнкцией наподобие самой формулы F^h. В первом случае формула F получается по регулярному выражению, не содержащему оператора *, во втором – с оператором *. Поэтому рассуждения в данном случае носят индуктивный характер.

Базис индукции. Вначале полагаем, что подформула F построена по регулярному выражению, не содержащему оператора *.

Представим формулу F^h в виде $H_0 \vee H_1 \vee \ldots \vee H_h$. Справедливо утверждение.

***Теорема* 14.** *При начальном означивании входных метапеременных формулы F^h и некотором единственном $i \geq 0$:*

либо в точности одна формула H_i превращается в логическую единицу, а все остальные $H_0, H_1, \ldots, H_{i-1}, H_{i+1}, \ldots, H_h$ – в логические нули;

либо все формулы $H_0, H_1, \ldots, H_i$ превращаются в тождественные нули, а остальные $H_{i+1}, H_{i+2}, \ldots, H_h$ - в формулы, отличные от константы.

Доказательство. Полагаем, что при начальном означивании метапеременных α в формулах $S^l(\alpha, \beta)$ метапеременные β означиваются единственным максимальным набором, при котором сама формула $S^l(\alpha, \beta)$ превращается в тождественную 1. Это происходит по следующей причине.

Формула $S^l(\alpha, \beta)$ появляется вместо арифметического оператора $s^l(\alpha)$, где β - это новая метапеременная. Поэтому слева от формулы $S^l(\alpha, \beta)$ метапеременная β не встречается (но может встречаться правее). Следовательно, означивание всех метапеременных слева от этой формулы не означивает метапеременной β.

Метапеременная α представляется бесконечной последовательностью $\alpha_0, \ldots, \alpha_m, \ldots$ булевских метапеременных. Следовательно, означивание логических метапеременных $\alpha_0, \ldots, \alpha_m, \ldots$ представляет собой некоторое начальное не нулевое означивание, за которым следует бесконечная последовательность нулей. Было показано, что при таком означивании метапеременной α имеется лишь

единственное означивание метапеременной β, при котором формула $S^l(\alpha, \beta)$ истинная. При всех остальных значениях метапеременной β формула $S^l(\alpha, \beta)$ ложна. И полученное для формулы $S^l(\alpha, \beta)$ означивание β является означиванием и всех остальных ее вхождений, которые расположены правее.

Все метапеременные формулы $0(\alpha)$ означиваются нулями.

Возможные означивания аргументов формул E^σ приводят, как к тождественной константе, так и к некоторой формуле, не являющейся тождественной константой. Как было показано, это определяется тем, содержит ли эта формула аргументом – входную переменную программы.

Пусть σ_α есть начальное означивание входных метапеременных рассматриваемой формулы. Возможны следующие случаи.

1. Формула $E^{1-\sigma}(\alpha)$ при этом означивании есть тождественная 1. Тогда выходные метапеременные β принимают значения, которые определяются означиваниями метапеременных α^w. В итоге метапеременные β также приобретают конкретное значение. Все остальные подформулы $H_1, H_2, \ldots, H_i, \ldots, H_h$ становятся тождественно ложными, как легко увидеть из их конструкции.

2. Формула $E^{1-\sigma}(\alpha)$ при этом означивании есть тождественный 0. В остальных формулах $H_1, H_2, \ldots, H_i, \ldots, H_h$ присутствуют подформулы $E^\sigma(\alpha)$, которые при означивании σ_α становятся тождественно истинными. Логический оператор $E^\sigma(\alpha)$ не порождает означивания новых метапеременных, как это делают арифметические операторы, и поэтому мы переходим к рассмотрению функции G вида

$$E^{1-\sigma}(\alpha^1)\,(\beta^1 = \beta) \vee E^\sigma(\alpha^1)\,F(\alpha^1, \beta^2)\,E^{1-\sigma}(\alpha^2)\,(\beta^2 = \beta) \vee$$

$$\ldots$$

$$E^\sigma(\alpha^1)\,F(\alpha^1, \beta^2) \ldots E^\sigma(\alpha^{h-1})\,F(\alpha^{h-1}, \beta^h)\,E^{1-\sigma}(\alpha^h)\,(\beta^h = \beta),$$

для которой задано начальное означивание σ_{α^1} ее входных метапеременных. Далее применимы индуктивные рассуждения.

3. При этом означивании функция $E^{1-\sigma}(\alpha)$ преобразуется в функцию, отличную от константы. В этом случае все формулы $H_1, H_2, ..., H_i, ..., H_h$ превращаются в новые формулы, отличные от констант.

Рассуждая по индукции, заметим, что переход от исходной формулы к формуле G осуществляется только при условии, что функция $E^{1-\sigma}(\alpha)$ при этом означивании превращается в тождественный 0. Поэтому, если некоторая подформула H_i превращается в функцию не константу, то это означает, что все предыдущие функции $H_1, H_2, ..., H_{i-1}$ тождественно ложны.

С другой стороны, если какая-либо формула H_i превращается в *истину*, то все остальные $H_{i+1}, ..., H_h$ превращаются в тождественную *ложь*. Это следует из того, что все функции $H_0, H_1, ..., H_i, ..., H_h$ в определенном смысле попарно ортогональны. А именно, для каждой формулы вида E^{σ} из H_i во всех остальных формулах $H_0, H_1, ..., H_{i-1}, H_{i+1}, ..., H_h$ найдутся подформулы с конъюнктивным членом $E^{1-\sigma}$ и с теми же аргументами. Отметим, что только в последнем случае выходные метапеременные β приобретают конкретные значения, которые определяются видом формулы H_i.

Теорема доказана.

Таким образом, логические формулы E^{σ} определяют в исходной формуле F^h различные классы эквивалентности в зависимости от длины означивающих векторов. Увеличение длины означивающих векторов происходит в результате применения арифметических операторов. Поэтому, чтобы из нулевого вектора получить вектор, содержащий единичное значение в l-м компоненте, необходимо совершить 2^l операций прибавления 1. Т.е. увеличение длины происходит в результате длинной последовательности арифметических операций программы, т.е. относительно редко.

Индукционный переход предполагает, что рассматриваемая формула F^h построена по регулярному выражению, содержащему оператор *. Отметим, что в базисном случае, т.е. когда формула F^h соответствует линейному регулярному выражению, значения выходных метапеременных β определены лишь в единственном случае, когда в точности одна подформула H_i при заданном означива-

нии истинная. И таких формул, превращающихся в тождественную единицу при означивании σ_α – не более одной. Эта ситуация соответствует траектории вычисления программы, когда мы проходим цикл необходимое число раз после чего получаем определенное выходное значение.

Ситуация, когда исходная формула F^h превращается в логическую формулу, отличную от константы, соответствует тому, что мы проходим несколько ($k \geq 0$) раз цикл и потом логическое условие не имеет конкретного логического значения. Так как логическое значение не определено, то вычисление не продолжается. Поэтому при рассмотрении индукционного перехода мы рассматриваем две возможности: первая – когда означивание переменных формулы F^h приводит к тому, что эта формула становится истинной при соответствующем означивании выходным метапеременных, и вторая - когда эта формула превращается в логическую функцию, отличную от константы. В первом случае формула F^h становится константой. Во втором случае формула F^h превращается в логическую не константную функцию.

Следовательно, при начальном означивании входной переменной вид формулы F^h определяется лишь видом подформул E, в которых один аргумент - входная переменная. Очевидно, что число вхождений в программу логических вершин ограниченно некоторой константой. Всякое вхождение логической вершины с входным аргументом влечет появление в формуле F^h неопределенности, т.е. логической формулы, отличной от константы. И таких вхождений может быть несколько, пусть $E(\sigma_x,\alpha_1),\ \ ...,E(\sigma_x,\alpha_h)$.

Так как эти формулы не являются логическими константами, то выполняются следующие условия: ***x***-набор σ_x является префиксом каждого значения метапеременной α_i, $i = 1, 2, \ldots, h$, и выполняются неравенства $\alpha_1 \leq \ldots \leq \alpha_h$. Так как все значения $\alpha_1,\ldots,\alpha_h$ обладают одинаковым префиксом, то между ними расположены формулы S^l, сумма верхних индексов которых не менее $2^{|\sigma_x|}$. Поэтому можно считать, что $h = 1$ и в каждой формуле F встречается в точности одна

подформула *E*, в которой один аргумент – входная метапеременная, а второй – метапеременная, полученная по одной и той же рабочей переменной.

3. О длине вычислений программ. В этом разделе покажем, что длина вычисления программы вычисляющей некоторую функцию, определяется ее энтропией.

Пусть задано множество $M_{\boldsymbol{x}}$ всех начальных означиваний длины n входной переменной $\boldsymbol{x}$ программы π, вычисляющей функцию $\varphi(\boldsymbol{x})$, и эти означивания характеризуются энтропией H_x^{φ}. Если обобщенная формула $F(\boldsymbol{x}, \boldsymbol{y})$ представляет функцию $\boldsymbol{y} = \varphi(\boldsymbol{x})$, то для того же множества означиваний аргумента $\boldsymbol{x}$ имеем $H_x^F = H_x^{\varphi}$. Допустим, что формула $F(\boldsymbol{x}, \boldsymbol{y})$, представляющая функцию $\boldsymbol{y} = \varphi(\boldsymbol{x})$, построена по программе π, как описано выше. Каждый класс эквивалентности, на которые разбивается множество $M_{\boldsymbol{x}}$ означиваний, получается в результате означивания аргументов подформул E^{σ} из $F(\boldsymbol{x}, \boldsymbol{y})$, у которых в точности один аргумент является входной переменной.

Пусть σ_x есть некоторое начальное означивание из $M_{\boldsymbol{x}}$, у которого самая правая единица располагается в *l*-м разряде. При правильном означивании второй аргумент всякой подформулы E^{σ} из $F(\boldsymbol{x}, \boldsymbol{y})$, которая определяет отнесение набора σ_x к соответствующему классу эквивалентности, должен также иметь единицу в *l*-м разряде правее которого следует еще некоторая последовательность ненулевых компонентов. Этим аргументом является рабочая метапеременная. Назовем такие последовательности *дополнительными*. С помощью этих дополнительных последовательностей осуществляется кодирование классов эквивалентности в следующем смысле: как показано, отнесение функции к тому или иному классу происходит на основании тех разрядов, которыми характеризуются дополнительные последовательности. Поэтому каждый класс обладает фиксированным набором дополнительных последовательностей.

Из этих рассуждений следует

***Теорема* 15**. *Пусть* $\pi(\boldsymbol{x})$ - *арифметическая программа, вычисляющая функцию* $\varphi(\boldsymbol{x})$ *и* $F(\boldsymbol{x}, \boldsymbol{y})$ *– обобщенная функция, построенная по* $\pi(\boldsymbol{x})$. *Тогда, если*

$H_{\boldsymbol{x}}$ – есть энтропии, определяемая некоторым множеством всех начальных означиваний аргумента $\boldsymbol{x}$ *одинаковой длины, то* $F(\boldsymbol{x}, \boldsymbol{y})$ *имеет длину не менее* $2^{dH_{\boldsymbol{x}}}$ *для некоторой положительной константы* d.

Доказательство. Если формула $F(\boldsymbol{x}, \boldsymbol{y})$ для заданного множества означиваний аргумента $\boldsymbol{x}$ обладает энтропией $H_{\boldsymbol{x}}$, число классов k эквивалентности не менее, чем $2^{H_{\boldsymbol{x}}}$. Кодирование классов эквивалентности, которое получается при правильном означивании, позволяет получить такое их число лишь при условии, что означивания некоторых рабочих метапеременных, обладают единичным компонентом в k-м разряде. Но чтобы получить такой вектор требуется не менее 2^k операций прибавления единицы. Поэтому длина формулы $F(\boldsymbol{x}, \boldsymbol{y})$ должна быть не менее $2^{H_{\boldsymbol{x}}}$.

Теорема доказана.

Следствие. *Если все начальные наборы входной переменной обладают длиной* n *и всюду определенная функция* $\varphi(\boldsymbol{x})$ *при таком означивании обладает энтропией* H, *то существуют вычисления программы* $\pi(\boldsymbol{x})$ *длины не менее* 2^{dH} *для некоторой положительной константы* d.

Литература.

1. Янов Ю.И. Сб. Проблемы кибернетики, вып. 32, М.:Наука, 1977.