Пример полиномиального запроса, не распознаваемого в недетерминированной логарифмической памяти *

М.А.Тайцлин
Кафедра информатики
Тверской государственный университет
Горького, 108, квартира 148, Тверь, 170005
телефон: (7 0822) 315358
Michael.Taitslin@tversu.ru

20 июня 2005 г.

Аннотация

Мы предлагаем некоторый запрос, лежащий в PTIME, который, как мы полагаем, не лежит в NLogSpace. К сожалению, доказательство непринадлежности этого запроса к NLogSpace неполно. Но если предлагаемый запрос действительно не лежит в NLogSpace, то это решает давно открытую проблему Кука, поставленную ещё в 1969 году в [1].

We propose a problem in PTIME which, we believe, is not in in NLogSpace. If it is the case, it would settle a long staying open problem of Stephen A.Cook posted in 1969 (see [1]).

1 Введение

В общей теории сложности вычислений одной из наиболее глубоких является известная теорема Кука (см. [2]):

Следующие три условия эквивалентны для любого множества A слов в алфавите Γ и любой функции $L(n) \geqslant \log_2 n$ на положительных целых числах.

(a) A задаётся детерминированной стековой машиной Тьюринга, посещающей не более $L(n)$ ячеек на каждой рабочей ленте.

(b) A задаётся недетерминированной стековой машиной Тьюринга, посещающей не более $L(n)$ ячеек на каждой рабочей ленте.

(c) A задаётся детерминированной машиной Тьюринга, делающей не более $T(n) = 2^{cL(n)}$ тактов для некоторой константы c.

Здесь n — это длина входа.

В [2], как и в [1], спрашивается, можно ли в этой теореме Кука заменить стековую машину Тьюринга на обычную многоленточную машину Тьюринга. См. также [3] и [4].

Я думаю, что это сделать нельзя, и предлагаю запрос, заведомо лежащий в PTIME, который, как я думаю, не лежит в NLogSpace.

2 Формулировка теоремы

Пусть

$$\vartheta_1, \vartheta_2, \ldots, \vartheta_{(LAST+1)^3}$$

*Работа поддержана Российским фондом фундаментальных исследований (номер проекта: 04-01-00015).

представляет собой перечисление всех трёхэлементных последовательностей, составленных из элементов множества

$$|\mathcal{A}| = \{0,\ 1, \ldots, LAST\},$$

в порядке возрастания, другими словами,

$$\vartheta_1 = \langle 0,\ 0,\ 0\rangle,\ \vartheta_2 = \langle 0,\ 0,\ 1\rangle,\ \ldots,\ \vartheta_{(LAST+1)^3} = \langle LAST,\ LAST,\ LAST\rangle.$$

Мы рассматриваем конечные алгебраические системы

$$\mathcal{A} = (\{0, 1, \ldots, LAST\},\ f,\ \ a,\ c,\ ',\ 0,\ LAST)$$

сигнатуры

$$\Omega = \langle f^{(2)},\ \ a,\ c,\ ',\ 0,\ LAST\rangle,$$

где $LAST$ — это положительное целое число, a и c — это элементы множества $\{0, 1, \ldots, LAST\}$, f — это двухместная операция на $\{0, 1, \ldots, LAST\}$, а i' есть $i+1$ (следующее целое число), если i меньше $LAST$, и

$$LAST' = LAST.$$

Каждую такую систему мы называем **стандартной**.

Стандартное задание стандартной системы есть следующее слово в алфавите $\{0,\ 1,\ *\}$:

$$\#(LAST) * \#(a) * \#(c) * \#(f),$$

где для неотрицательного целого числа i слово

$$\#(i)$$

представляет собой запись числа i в двоичной системе счисления, а

$$\#(f)$$

есть определяемый ниже код операции f.

Последовательность

$$\xi = \xi_1 \xi_2 \ldots \xi_\ell,$$

составленная из 0 и 1, длины $\ell = (LAST+1)^3$ называется **кодом** операции f, если для любого целого положительного числа $u \leqslant (LAST+1)^3$ выполняется следующее условие:

$$\xi_u = \begin{cases} 1, \text{ если } \vartheta_u \in graph(f), \\ 0, \text{ если } \vartheta_u \notin graph(f). \end{cases}$$

Для любых $i, j, k \in \{0, 1, \ldots, LAST\}$ здесь запись $\langle i, j, k\rangle \in graph(f)$ означает, что $f(i, j) = k$.

Мы рассматриваем следующий запрос:

по стандартному заданию стандартной алгебраической системы определить, принадлежит ли c подсистеме, порождённой a в

$$(\{0, 1, \ldots, LAST\},\ f).$$

Тривиально (и хорошо известно), что этот запрос лежит в $PTIME$. Мы полагаем, что этот запрос не лежит в $NLogSpace$.

3 Схемы программ

Иногда мы пишем (uv) вместо $f(u, v)$.

Определение 1. *Тест есть выражение одного из видов $v_1 = v_2$ или $v_1 \neq v_2$, где v_1 и v_2 являются переменными. Эти переменные v_1 и v_2 входят в этот тест. Любая переменная, отличная от v_1 и v_2, не входит в этот тест.*

Схемой программы сигнатуры Ω называется каждое из следующих выражений:

(1) $x \leftarrow a$, *где* x *является переменной; эта схема программы использует переменную* x *и не использует других переменных;*

(2) $x \leftarrow c$, *где* x *является переменной; эта схема программы использует переменную* x *и не использует других переменных;*

(3) $x \leftarrow x'$, *где* x *является переменной; эта схема программы использует переменную* x *и не использует других переменных;*

(4) $x \leftarrow 0$, *где* x *является переменной; эта схема программы использует переменную* x *и не использует других переменных;*

(5) $x \leftarrow LAST$, *где* x *является переменной; эта схема программы использует переменную* x *и не использует других переменных;*

(6) $x \leftarrow (yz)$, *где* x, y *и* z *являются различными переменными; эта схема программы использует переменные* x, y *и* z *и не использует других переменных;*

(7) $x \leftarrow y$, *где* x *и* y *являются переменными; эта схема программы использует переменные* x *и* y *и не использует других переменных;*

(8) `if` φ `then` Π_1 `else` Π_2 `fi`, *где* φ *— это тест, а* Π_1, Π_2 *— схемы программ; переменная используется этой схемой программы тогда и только тогда, когда эта переменная используется* Π_1 *или* Π_2, *а также, когда эта переменная входит в* φ*;*

(9) `while` φ `do` Π `od`, *где* φ *— это тест, а* Π *— схема программы; переменная используется этой схемой программы тогда и только тогда, когда эта переменная используется* Π, *а также, когда эта переменная входит в* φ*;*

(10) $\Pi_1; \Pi_2$, *где* Π_1, Π_2 *— это схемы программ; переменная используется этой схемой программы тогда и только тогда, когда эта переменная используется* Π_1 *или* Π_2*;*

(11) Π_1 `OR` Π_2, *где* Π_1, Π_2 *— это схемы программ; переменная используется этой схемой программы тогда и только тогда, когда эта переменная используется* Π_1 *или* Π_2.

Схема программы одного из видов (1)—(7) называется присваиванием.

Схема программы называется схемой программы с переменными из X *тогда и только тогда, когда* X *есть множество всех переменных, используемых этой схемой программы.*

Пусть Π *— это схема программы с переменными из* X_1, X *— множество переменных,* $X \supseteq X_1$, *а* $\mathcal{M}$ *— алгебраическая система сигнатуры* Ω. Π *определяет бинарное отношение на множестве всех* X*-оценок для* $\mathcal{M}$ *(другими словами, на множестве всех отображений* $\sigma : X \longrightarrow |\mathcal{M}|$*) следующим образом.*

(12) $(x \leftarrow t)(\sigma_1, \sigma_2)$ *тогда и только тогда, когда* $\sigma_1(t) = \sigma_2(x)$ *и* $\sigma_1(y) = \sigma_2(y)$ *для всех* $y \in X$, *отличных от* x*;*

(13) $($`if` φ `then` Π_1 `else` Π_2 `fi`$)(\sigma_1, \sigma_2)$ *тогда и только тогда, когда либо* $\sigma_1 \models \varphi$ *и* $\Pi_1(\sigma_1, \sigma_2)$, *либо* $\sigma_1 \models \neg\varphi$ *и* $\Pi_2(\sigma_1, \sigma_2)$;

(14) $($`while` φ `do` Π `od`$)(\sigma_1, \sigma_2)$ *тогда и только тогда, когда либо* $\sigma_1 \models \neg\varphi$ *и* $\sigma_1 = \sigma_2$, *либо* $\sigma_1 \models \varphi$ *и существует такая* X*-оценка* σ_3, *что* $\Pi(\sigma_1, \sigma_3)$ *и* $($`while` φ `do` Π `od`$)(\sigma_3, \sigma_2)$.

(15) $(\Pi_1; \Pi_2)(\sigma_1, \sigma_2)$ *тогда и только тогда, когда существует такая* X*-оценка* σ_3, *что* $\Pi_1(\sigma_1, \sigma_3)$ *и* $\Pi_2(\sigma_3, \sigma_2)$*;*

(16) $(\Pi_1$ `OR` $\Pi_2)(\sigma_1, \sigma_2)$ *тогда и только тогда, когда* $\Pi_1(\sigma_1, \sigma_2)$ *или* $\Pi_2(\sigma_1, \sigma_2)$.

Для любого множества $X \supseteq X_1$ *переменных любая схема программы с переменными из* X_1 *определяет глобальный предикат местности* $2\,|X|$, *где* $|X|$ *— это число элементов множества* X.

Именно, если $X = \{x_1, \ldots, x_n\}$, для $a_1, \ldots, a_n, b_1, \ldots, b_n \in |\mathcal{M}|$ тогда и только тогда $\Pi(a_1, \ldots, a_n, b_1, \ldots, b_n)$, когда $\Pi(\sigma_1, \sigma_2)$ для таких σ_1, σ_2, что $\sigma_1(x_i) = a_i$ и $\sigma_2(x_i) = b_i$ для $i = 1, \ldots, n$.

Для X-оценки σ_1 для Ω-системы $\mathcal{M}$ и схемы программы Π сигнатуры Ω скажем, что Π останавливается на σ_1, если существует такая X-оценка σ_2, что $\Pi(\sigma_1, \sigma_2)$.

Иногда мы пишем $(\sigma_1 \Pi \sigma_2)$ вместо $\Pi(\sigma_1, \sigma_2)$.

Схема программы Π с переменными из $X = \{x_1, \ldots, x_n\}$ называется *замкнутой*, если Π есть $\Pi_1; \Pi_2$, где Π_1 есть

$$x_1 \leftarrow b_1; \ldots; x_n \leftarrow b_n$$

для некоторых $b_1, \ldots, b_n \in \{a, c, 0, LAST\}$. Замкнутая схема программы Π останавливается на системе $\mathcal{M}$ сигнатуры Ω тогда и только тогда, когда Π останавливается на некоторой X-оценке для Ω-системы $\mathcal{M}$. Две замкнутые схемы программ Π_1 и Π_2 называются эквивалентными, если для любой Ω-системы $\mathcal{M}$ схема Π_1 останавливается на $\mathcal{M}$ тогда и только тогда, когда Π_2 останавливается на $\mathcal{M}$.

Глобальный предикат — это некоторое множество стандартных алгебраических систем сигнатуры Ω. Глобальный предикат лежит в $NLogSpace$, если существует такая машина Тьюринга M из $NLogSpace$, что M воспринимает стандартное задание стандартной системы тогда и только тогда, когда эта система принадлежит этому глобальному предикату.

Следующий результат легко доказывается и содержится в [7].

Теорема 3.1 ([7], лемма 3.3.3). *Для всякого глобального предиката из $NLogSpace$ существует такая схема программы сигнатуры Ω, что эта схема останавливается на стандартной алгебраической системе тогда и только тогда, когда эта система принадлежит рассматриваемому глобальному предикату.*

Таким образом, достаточно доказать, что не существует такой замкнутой схемы программы сигнатуры Ω, что эта схема останавливается на стандартной системе тогда и только тогда, когда c принадлежит подсистеме, порождённой a в

$$(\{0, 1, \ldots, LAST\},\ f).$$

Зафиксируем конечное множество X переменных.

Тест с переменными из X — это выражение одной из форм $v_1 = v_2$ или $v_1 \neq v_2$, где v_1 и v_2 являются переменными из X. Тест вида $v_1 = v_2$ называется положительным, а тест вида $v_1 \neq v_2$ — отрицательным.

Смысл теста с переменными из X в следующем: тест не меняет значения никакой переменной; следующая X-оценка существует и равна заданной X-оценке тогда и только тогда, когда этот тест выполняется для заданной X-оценки. Любой тест с переменными из X может быть представлен как схема программы с переменными из X. В самом деле, тест φ можно представить как

$$\texttt{while } \neg\varphi \texttt{ do } v \leftarrow v \texttt{ od},$$

где v — это переменная из X.

Линейная программа с переменными из X есть присваивание с переменными из X и имеет длину 1, тест с переменными из X и имеет длину 1 или выражение вида $\Pi_1; \Pi_2$, где Π_1 и Π_2 являются линейными программами с переменными из X. В последнем случае длина этой линейной программы равна сумме длин линейных программ Π_1 и Π_2.

Программа с переменными из X есть объединение линейных программ с переменными из X. Программа с переменными из X называется замкнутой, если эта программа является объединением замкнутых линейных программ с переменными из X. Программа с переменными из X называется регулярной, если эта программа является регулярным языком в алфавите всех используемых присваиваний и тестов. Так как при фиксированном конечном X множество всех возможных присваиваний и тестов конечно, то алфавит, состоящий из всех испольуемых присваиваний и тестов, тоже конечен.

Каждая линейная программа с переменными из X становится схемой программы после замены тестов на соответствующие схемы программ. Эта схема программы для каждой стандартной Ω-системы $\mathcal{M}$ определяет бинарное отношение на множестве всех X-оценок для

$\mathcal{M}$. Заданная линейная программа определяет то же самое бинарное отношение. Программа определяет такое бинарное отношение на множестве всех X-оценок для $\mathcal{M}$, которое является объединением всех бинарных отношений, определяемых линейными программами, входящими в рассматриваемую программу. Более подробно, для любых двух X-оценок σ_1 и σ_2 и программы Π с переменными из X тогда и только тогда $(\sigma_1 \Pi \sigma_2)$, когда существует такая линейная программа Π_1 в Π, что $(\sigma_1 \Pi_1 \sigma_2)$.

Следующая теорема совершенно тривиальна.

Теорема 3.2 ([7], предложение 1.2.2)**.** *Для любой схемы программы* Π *с переменными из* X *можно эффективно построить такую регулярную программу* Π_1 *с переменными из* X, *что* Π *и* Π_1 *для любой стандартной* Ω*-системы* $\mathcal{M}$ *определяют то же самое бинарное отношение на множестве всех* X*-оценок для* $\mathcal{M}$.

Если при этом схема программы Π *с переменными из* X *является замкнутой, то построенная регулярная программа* Π_1 *с переменными из* X *тоже будет замкнутой.*

Программа Π с переменными их X и схема программы Π_1 с переменными из X называются **строго эквивалентными**, если Π и Π_1 для любой Ω-системы $\mathcal{M}$ определяют то же самое бинарное отношение на множестве всех X-оценок для $\mathcal{M}$.

Замкнутая программа останавливается на стандартной системе $\mathcal{M}$, если в ней существует линейная программа, которая останавливается на $\mathcal{M}$.

Замкнутые программа и схема программы называются **эквивалентными**, если для любой стандартной Ω-системы $\mathcal{M}$ либо эти программа и схема программы обе останавливаются на $\mathcal{M}$, либо эти программа и схема программы обе не останавливаются на $\mathcal{M}$.

4 Идемпотентные квазигруппы

Группоид $(G, \circ)$ называется идемпотентным, если

$$(x \circ x) = x$$

для всех $x \in G$.

Группоид $(G, \circ)$ называется квазигруппой, если для любых $a, b \in G$ каждое из уравнений

$$(a \circ x) = b, \quad (y \circ a) = b$$

имеет и при том единственное решение в $(G, \circ)$.

Рассмотрим следующую четырёхэлементную идемпотентную квазигруппу $(Q_4, \circ)$.

Элементами $(Q_4, \circ)$ являются 1, 2, 3 и 4.

Операция $\circ$ определяется в $(Q_4, \circ)$ следующими правилами:

$$(1 \circ 1) = (2 \circ 3) = (3 \circ 4) = (4 \circ 2) = 1;$$

$$(2 \circ 2) = (1 \circ 4) = (4 \circ 3) = (3 \circ 1) = 2;$$

$$(3 \circ 3) = (1 \circ 2) = (2 \circ 4) = (4 \circ 1) = 3;$$

$$(4 \circ 4) = (2 \circ 1) = (1 \circ 3) = (3 \circ 2) = 4.$$

Пусть $\circ_1$ есть $\circ$.

Дальнейшие построения используют индукцию по i. Допустим, что идемпотентная квазигруппа $(Q_{4^i}, \circ_i)$, содержащая 4^i элементов, уже определна. Пусть $a_1, a_2, \ldots, a_{4^i}$ — перечисление всех элементов квазигруппы Q_{4^i}. Рассмотрим четыре изоморфные копии квазигруппы $(Q_{4^i}, \circ_i)$, попарно не имеющие общих элементов. Пусть $Q_{4^{i+1}}$ будет объединение всех элементов всех этих четырёх копий. Тогда $Q_{4^{i+1}}$ содержит 4^{i+1} элементов. Для $j = 1, 2, 3, 4$ пусть τ_j будет изоморфизм между $(Q_{4^i}, \circ_i)$ и копией номера j. Для $k = 1, 2, \ldots, 4^i$ и $j = 1, 2, 3, 4$ пусть $a_{k,j}$ обозначает $\tau_j(a_k)$. Тогда для $k_1, k_2 \in \{1, 2, \ldots, 4^i\}$ и $j_1, j_2 \in \{1, 2, 3, 4\}$ пусть

$$(a_{k_1, j_1} \circ_{i+1} a_{k_2, j_2}) = \tau_{(j_1 \circ j_2)}(a_{k_1} \circ_i a_{k_2}).$$

Очевидно, что $(Q_{4^{i+1}}, \circ_{i+1})$ является идемпотентной квазигруппой.

По индукции, идемпотентные квазигруппы $(Q_{4^i}, \circ_i)$ определены для всех целых положительных чисел i.

Ясно, что для любого целого положительного числа i любой элемент a квазигруппы $(Q_{4^i}, \circ_i)$ определяет такое 1-1 отображение ρ_a из Q_{4^i} на Q_{4^i}, что $(b \circ_i \rho_a(b)) = a$ для любого $b \in Q_{4^i}$. При этом $\rho_a(a) = a$.

В дальнейшем мы пишем $\circ$ вместо $\circ_i$, если и так понятно, какое i имеется в виду.

5 Регистровая сложность термов

Определение 2. *Зафиксируем список переменных $X = z_1, \ldots, z_m$ и положительное целое число I. Рассмотрим множество Q_{4^I}.*

С каждой последовательностью α присваиваний, использующих переменные только из X, единственный символ f операции и символы констант только из Q_{4^I} (каждое такое присваивание имеет один из видов $u \leftarrow v$, $u \leftarrow a$ или $u \leftarrow (vw)$, где $u, v, w \in X$ и $a \in Q_{4^I}$), мы связываем отображение $\mu(\alpha)$ из X во множество $\langle f,\ a \mid a \in Q_{4^I} \rangle$-термов. Это отображение определяется индукцией по длине s последовательности α.

Если α является пустой последовательностью (значит, $s = 0$), то $\mu(\alpha)(x) = x$ для каждого $x \in X$.

Если α есть $\alpha'; \alpha_s$, рассмотрим отдельно три случая в зависимости от того, что представляет собой присваивание α_s.

Case 1. Если α_s есть $x_i \leftarrow x_j$, то

$$\mu(\alpha)(y) = \mu(\alpha')(y)$$

для $y \in X \setminus \{x_i\}$ и

$$\mu(\alpha)(x_i) = \mu(\alpha')(x_j).$$

Case 2. Если α_s есть $x_i \leftarrow a$, где a — это константный символ из Q_{4^I}, то

$$\mu(\alpha)(y) = \mu(\alpha')(y)$$

для $y \in X \setminus \{x_i\}$ и

$$\mu(\alpha)(x_i) = a.$$

Case 3. Если α_s есть $x_i \leftarrow (x_{j_1} x_{j_2})$, то

$$\mu(\alpha)(y) = \mu(\alpha')(y)$$

для $y \in X \setminus \{x_i\}$ и

$$\mu(\alpha)(x_i) = (\mu(\alpha')(x_{j_1}) \mu(\alpha')(x_{j_2})).$$

Скажем, что такая последовательность α присваиваний вычисляет терм t в переменной $y \in X$, если t есть $\mu(\alpha)(y)$. При этом число переменных во множестве X называется числом переменных, используемых при вычислении терма t.

Регистровая сложность rc*(t) терма t — это наименьшее из таких целых положительных чисел n, что t вычисляется последовательностью присваиваний, использующей только n переменных.*

Пример 1. Терм

$$((a(aa))(((aa)(aa))(((aa)(aa))((aa)(aa)))))$$

нельзя вычислить, используя только две переменные.

Пример 2. Для вычисления терма

$$((a(aa))((a(aa))((a(aa))(a(aa)))))$$

достаточно двух переменных.

Через $g(i)$ для неотрицательного целого числа i мы обозначаем терм, определяемый индукцией по i следующим образом:

$$g(0) = x_1, \quad g(i+1) = (g(i)g^*(i)),$$

где $s = 2^i$ и $g^*(i) = g(i)(x_{s+1}, \ldots, x_{2s})$.

В частности,

$$g(1) = (x_1x_2), \quad g(2) = ((x_1x_2)(x_3x_4)),$$
$$g(3) = (((x_1x_2)(x_3x_4))((x_5x_6)(x_7x_8))),$$
$$g(4) = ((((x_1x_2)(x_3x_4))((x_5x_6)(x_7x_8)))(((x_9x_{10})(x_{11}x_{12}))((x_{13}x_{14})(x_{15}x_{16}))))$$

и так далее.

Обычно $\langle f,\ a \mid a \in Q_{4^I}\rangle$-термы изображаются конечными бинарными деревьями, висячие вершины которых помечены переменными или элементами множества Q_{4^I}. Тогда $g(i)$ является полным бинарным деревом высоты i, в котором различные висячие вершины помечены различными переменными.

Для любого $a \in Q_{4^I}$, если каждая висячая вершина полного бинарного дерева высоты i помечена a, то это дерево изображает терм, обозначаемый как $(\mathfrak{i})_a$. Следовательно,

$$(\mathfrak{i})_a = g(i)(a, \ldots, a)$$

($(\mathfrak{i})_a$ получается из $g(i)$ заменой каждой переменной на a).

Понятно, что регистровая сложность терма $(\mathfrak{i})_a$ равна 1.

В дальнейшем $\mathfrak{i}$ обозначает один из термов $(\mathfrak{i})_a$ для некоторого $a \in Q_{4^I}$. Так что $\mathfrak{i}$ обозначает любой из возможных 4^I термов.

Определение 3. *$\langle f,\ a \mid a \in Q_{4^I}\rangle$-термы t_1 и t_2 независимы тогда и только тогда, когда они различны.*

Пусть $s = 2r$ и $r = 2^i$.

$\langle f,\ a \mid a \in Q_{4^I}\rangle$-термы $t_1, \ldots, t_s$ независимы тогда и только тогда, когда выполняются следующие условия:

- *термы $t_1, \ldots, t_r$ независимы;*
- *термы $t_{r+1}, \ldots, t_s$ независимы;*
- *для каждого подтерма t' терма $g(i)$ терм $t'(t_1, \ldots, t_r)$ отличен от любого из термов $t_{r+1}, \ldots, t_s$;*
- *для каждого подтерма t' терма $g(i)$ терм $t'(t_{r+1}, \ldots, t_s)$ отличен от любого из термов $t_1, \ldots, t_r$.*

Теорема 5.1 ([6], [5]).

Пусть $s = 2^i$ и $\langle f,\ a \mid a \in Q_{4^I}\rangle$-термы $t_1, \ldots, t_s$ являются независимыми. Тогда $\mathrm{rc}(g(i)(t_1, \ldots, t_s)) \geqslant i + 1$.

Пример 3. Для любого целого числа $i > 2$ термы $\mathfrak{1},\ \mathfrak{2}, \ldots,\ \mathfrak{s}$, где $s = 2^i$, независимы. По теореме 5.1, $\mathrm{rc}(g(i)(\mathfrak{1},\ \mathfrak{2}, \ldots,\ \mathfrak{s})) \geqslant i + 1$.

6 Пример

Теорема 6.1. *Не существует такой замкнутой схемы программы сигнатуры*

$$\Omega = \langle f^{(2)},\ a,\ c,\ ',\ 0,\ LAST\rangle,$$

что для любой стандартной системы

(1) $$(G,\ f,\ a,\ c,\ ',\ 0,\ LAST)$$

сигнатуры Ω эта замкнутая схема программы останавливается на этой системе тогда и только тогда, когда c принадлежит подсистеме, порождённой a в $(G,\ f)$.

Мы пытаемся доказывать эту теорему методом от противного.

Допустим, что такая схема программы Π существует.

Значит, для любой стандартной системы (1) сигнатуры Ω схема программы Π останавливается на этой системе тогда и только тогда, когда c принадлежит подсистеме, порождённой a в $(G,\ f)$.

Имеет место следующая

Лемма 6.2 ([7], лемма 1.3.4)**.** *Любая замкнутая схема программы эквивалентна замкнутой схеме программы вида*

$$\Pi';\ \texttt{while}\ \varphi\ \texttt{do}\ \Pi_*\ \texttt{od}$$

где φ — это тест, Π' — последовательность присваиваний и Π_* является схемой программы, не содержащей `while`.

Мы предполагаем, что Π имеет именно такой вид.

Нам важно также не использовать выражения вида $LAST''^{\cdots}{}'$. С этой целью в Π мы заменим присваивания вида $x \leftarrow x'$ на

$$y \leftarrow LAST;\ \texttt{if}\ x = y\ \texttt{then}\ x \leftarrow x\ \texttt{else}\ x \leftarrow x'\ \texttt{fi},$$

где y является переменной, не используемой схемой программы Π.

Пусть все переменные, используемые схемой программы Π, содержатся в списке $z_1, \ldots, z_n$.

По теореме 3.2, существует такая замкнутая регулярная программа Π_1 сигнатуры Ω с переменными из $z_1, \ldots, z_n$, что для любой стандартной системы $\mathfrak{M}$ этой сигнатуры Π и Π_1 задают одно и то же бинарное отношение на множестве всех $\{z_1, \ldots, z_n\}$-оценок для $\mathfrak{M}$.

Более того, любая схема программы без `while` строго эквивалентна программе, которая является конечным объединением линейных программ. По этой причине мы предполагаем, что для линейных программ $\Pi^1, \ldots, \Pi^q$ с переменными из $z_1, \ldots, z_n$ программа Π_1 имеет вид

$$\Pi'; (\varphi; (\Pi^1 + \cdots + \Pi^q);)^* \neg\varphi,$$

где, как обычно, $+$ обозначает объединение множеств и * обозначает итерацию множества. Мы считаем, что $q < n$ и что для $i = 1, \ldots, q$ длина линейной программы Π^i меньше n.

Линейная программа, регулярная программа или схема программы сигнатуры Ω называется *канонической*, если для некоторых переменных u и v она имеет вид

$$u \leftarrow a; v \leftarrow c; \Pi_*$$

и все присваивания в Π_* имеют для некоторых различных переменных x, y, z один из видов $x \leftarrow (yz)$, $x \leftarrow x'$, $x \leftarrow y$, $x \leftarrow 0$ или $x \leftarrow LAST$.

Понятно, что любая замкнутая схема программы сигнатуры Ω эквивалентна некоторой замкнутой канонической схеме программы сигнатуры Ω. Поэтому мы предполагаем, что программа Π_1 является канонической.

Рассмотрим следующий класс $\mathfrak{K}$ стандартных алгебраических систем сигнатуры Ω. Каждая структура из $\mathfrak{K}$ имеет вид

$$(E_{nJ},\ f_{nJ},\ a,\ c,\ ',\ 0,\ LAST). \tag{2}$$

для некоторого целого положительного числа J. При этом группоид $(E_{nJ},\ f_{nJ})$ определяется следующим образом.

Выберем некоторое целое положительное число I. Q_{4^I}-термами назовём все элементы множества Q_{4^I} и все выражения вида $(t_1 t_2)$, в которых t_1 и t_2 уже являются Q_{4^I}-термами. Естественным образом определяется понятие подтерма для Q_{4^I}-терма. Другими словами, Q_{4^I}-термами мы называем не содержащие переменных $\langle f,\ a \mid a \in Q_{4^I} \rangle$-термы. В записи Q_{4^I}-терма все вхождения знака операции f мы опускаем.

По этому определению, Q_{4^I}-термами являются некоторые слова в алфавите

$$\{d,\ (,\ )\mid d \in Q_{4^I}\}.$$

Для $d \in Q_{4^I}$ длина Q_{4^I}-терма d равна 1. Длина Q_{4^I}-терма $(t_1 t_2)$ равна увеличенной на два сумме длин Q_{4^I}-термов t_1 и t_2. Таким образом, длина любого Q_{4^I}-терма равна длине задающего этот Q_{4^I}-терм слова в алфавите $\{d,\ (,\ )\mid d \in Q_{4^I}\}$.

Для целого положительного числа J и $s = 2^{nJ}$ рассмотрим множество G_{nJ} всех подтермов всех Q_{4^I}-термов $g(nJ)(\mathfrak{1}, \ldots, \mathfrak{s})$.

Для t_1 и t_2 из G_{nJ} мы определяем операцию f_{nJ} следующим образом:

$$f_{nJ}(t_1, t_2) = \begin{cases} (t_1 t_2), \text{ если } (t_1 t_2) \text{ лежит в } G_{nJ}, \\ t_1 \text{ в другом случае.} \end{cases}$$

Пусть k и ℓ — целые неотрицательные числа, $s_\ell = 2^k(\ell + 1)$ и $r_\ell = 2^k \ell + 1$.

Для $b \in Q_{4^I}$ пусть $\bar{b}^{k,\ell}$ обозначает Q_{4^I}-терм

$$g(k)((\mathfrak{r}_\ell)_b,\ \ldots,\ (\mathfrak{s}_\ell)_b).$$

Для $s = 2^{nJ}$ рассмотрим конгруэнтность $=$ на $(G_{nJ},\ f_{nJ})$, определяемую следующими правилами:

- $$(\bar{d}^{k,\ell}\ \bar{b}^{k,\ell+1}) = \overline{(d \circ b)}^{\ k+1,\ell/2}$$

  для любых $d, b \in Q_{4^I}$, $k = 0,\ 1,\ \ldots,\ nJ - 1$ и любых чётных неотрицательных целых чисел $\ell < 2^{nJ-k}$.

Таким образом, для $k = 0,\ 1,\ \ldots,\ nJ-1$ и любого чётного целого неотрицательного числа $\ell < 2^{nJ-k}$ терм

$$g(k+1)(\mathfrak{r}_\ell,\ \ldots,\ \mathfrak{s}_{\ell+1})$$

эквивалентен одному из попарно не эквивалентных термов

$$\{\bar{b}^{k+1,\ell/2} \mid b \in Q_{4^I}\}.$$

Для таких k и ℓ мы говорим, что терм $\bar{b}^{k+1,\ell/2}$ располагается на $(\ell/2)$-ой вершине уровня $(nJ - k - 1)$.

Пусть $(E_{nJ},\ f_{nJ})$ является фактор-группоидом группоида $(G_{nJ},\ f_{nJ})$ по этому отношению $=$.

Далее мы фиксируем I, J и $a \in Q_{4^I}$ и полагаем $m = nJ$ и $s = 2^{nJ}$.

Так как любая система (2) из $\mathfrak{K}$ является стандартной, то E_m есть

$$\{0,\ 1,\ \ldots,\ LAST\}.$$

Другими словами, существует одно-однозначная нумерация множества E_m всеми числами из $\{0,\ 1,\ \ldots,\ LAST\}$. Значит, любой класс конгруэнтности по отношению $=$ термов из G_m имеет номер в этой нумерации. Этот же номер мы приписываем также каждому терму из этого класса конгруэнтности. Итак, каждый Q_{4^I}-терм из G_m имеет номер.

Вычислим теперь $LAST$ через n, I и J. Имеется $(i+1)$ различных подтермов у $\mathfrak{i}$. Значит, имеется ровно

$$4^I(1 + 2^{nJ} + 2^{nJ-1} \cdots + 2 + 1) = 4^I 2^{nJ+1}$$

элементов во множестве E_m. Пусть $F(nJ) = 2^{nJ+1}$. Тогда $LAST = 4^I F(nJ) - 1$.

Нам достаточно рассматривать не все, а только *допустимые* нумерации множества E_m.

Нумерацию множества E_m называем допустимой, если она получается следующим образом.

Сначала, используя подряд числа 0, 1, 2 и так далее, произвольным образом приписываем номера элементам множества $\{b \mid b \in Q_{4^I}\}$.

После этого, продолжая использовать числа подряд, мы приписываем номера всем термам из множества

$$\{(\mathfrak{i})_b \mid i = 1, \ldots, s;\ b \in Q_{4^I}\}.$$

Для этого выбираем некоторую перестановку ρ множества $\{1, 2, \ldots, s\}$. Сначала мы нумеруем термы для $i = \rho(1)$. Затем для $i = \rho(2)$. Затем для $i = \rho(3)$ и так далее. Наконец, для $i = \rho(s)$.

После этого мы нумеруем все другие элементы множества E_m таким способом, что для любых $d, b \in Q_{4^I}$, любых $k_1, k_2 = 0,\ 1,\ \ldots,\ nJ$ и любых целых неотрицательных чисел $\ell_1 < 2^{nJ-k_1}$ и $\ell_2 < 2^{nJ-k_2}$ номер терма $\bar{d}^{\ k_1,\ell_1}$ меньше номера терма $\bar{b}^{\ k_2,\ell_2}$, если либо $k_1 < k_2$, либо $k_1 = k_2$ и $\ell_1 < \ell_2$.

Зафиксируем некоторую такую нумерацию.

Все другие допустимые нумерации получаются из неё следующим образом.

Для любых $k = 0,\ 1,\ \ldots,\ nJ$ и чётного неотрицательного числа $\ell < 2^{nJ-k}$ выберем некоторый автоморфизм $\tau_{k,\ \ell}$ квазигруппы $(Q_{4^I}, \circ_i)$. Для любых $k = 0,\ 1,\ \ldots,\ nJ-1$ and нечётного целого положительного числа $\ell < 2^{nJ-k}$ выберем такой автоморфизм $\tau_{k,\ \ell}$ квазигруппы $(Q_{4^I}, \circ_i)$, что

$$(\overline{\tau_{k,\ \ell}(b)}^{\ k,\ell}\ \overline{\tau_{k,\ \ell+1}(d)}^{\ k,\ell+1}) = \overline{\tau_{k+1,\ \ell/2}(b \circ d)}^{\ k+1,\ell/2} \tag{3}$$

для любых $b, d \in Q_{4^I}$. Из идемпотентности квазигруппы $(Q_{4^I}, \circ_i)$ тогда следует, что

$$(\overline{\tau_{k,\ \ell}(a)}^{\ k,\ell}\ \overline{\tau_{k,\ \ell+1}(a)}^{\ k,\ell+1}) = \overline{\tau_{k+1,\ \ell/2}(a)}^{\ k+1,\ell/2}. \tag{4}$$

Для любого такого множества автоморфизмов

$$\{\tau_{k,\ \ell} \mid k = 0,\ 1,\ \ldots,\ nJ;\ \ell < 2^{nJ-k}\},$$

для которого все условия (3) выполняются, рассмотрим такую нумерацию множества E_m, что для любых $k = 0,\ 1,\ \ldots,\ nJ$, $b \in Q_{4^I}$ и целого неотрицательного числа $\ell < 2^{nJ-k}$ номер элемента $\bar{b}^{\ k,\ell}$ в этой нумерации совпадает с номером элемента $\overline{\tau_{k,\ \ell}(b)}^{\ k,\ell}$ в зафиксированной нумерации. Пусть также номер любого элемента из $\{b \mid b \in Q_{4^I}\}$ в рассматриваемой нумерации совпадает с номером этого элемента в зафиксированной нумерации.

Номер элемента x из E_m в допустимой нумерации τ обозначается через $\nu_\tau(x)$. Кроме того $\nu_\tau(x)$ также обозначает элемент этого номера в зафиксированной нумерации.

Для различных x и y из E_m определим $f_\tau(x, y)$ так, чтобы $\nu_\tau(f_\tau(x, y))$ совпадал с $f_m(\nu_\tau(x), \nu_\tau(y))$. Пусть, кроме того, $f_\tau(x, x) = f_m(x, x)$ для $x \in E_m$, если x принадлежит подгруппоиду, порождённому a в $(E_m,\ f_m)$. Пусть, наконец, $f_\tau(x, x) = x$ для $x \in E_m$, если x не принадлежит подгруппоиду, порождённому a в $(E_m,\ f_m)$.

Это определение гарантирует, что для попарно различных номеров n_1, n_2 и n_3, $f_\tau(n_1, n_2) = n_3$ тогда и только тогда, когда $f_m(n_1, n_2) = n_3$. Значит, номер элемента $f_\tau(n_1, n_2)$ для различных номеров n_1, n_2 не зависит от рассматриваемой допустимой нумерации τ. Мы пишем (xy) вместо $f_\tau(x, y)$, если ясно, какая допустимая нумерация τ имеется в виду.

Из (4) следует, что c принадлежит подгруппоиду, порождённому a в $(E_m,\ f_m)$ тогда и только тогда, когда c принадлежит подгруппоиду, порождённому a в $(E_m,\ f_\tau)$.

Будем рассматривать класс $\mathfrak{K}_1$ стандартных систем. Стандартная система

$$(E_m,\ f_\tau,\ a,\ c,\ ',\ 0,\ LAST). \tag{5}$$

лежит в $\mathfrak{K}_1$ тогда и только тогда, когда τ является допустимой нумерацией и c принадлежит подгруппоиду, порождённому a в $(E_m,\ f_\tau)$.

Возмём в качестве c элемент $g(nJ)((\mathfrak{1})_a, \ldots, (\mathfrak{s})_a)$. По теореме 5.1 и примеру 3, $\mathrm{rc}(c) \geqslant nJ + 1$. Мы далее не различаем c и содержащий c класс конгруэнтности по отношению $=$. Тогда $c \in E_m$.

Так как c принадлежит подгруппоиду, порождённому a в $(E_m,\ f_\tau)$, программа Π_1 останавливается на (5). Более того, Π_1 останавливается на (5) для любой допустимой нумерации τ множества E_m.

Для допустимой нумерации τ множества E_m выбирем такую линейную программу $\Pi_2 \in \Pi_1$, что Π_2 останавливается на (5). Этот выбор $\Pi_2 \in \Pi_1$ зависит от рассматриваемой допустимой нумерации множества E_m.

Линейную программу Π_2 из Π_1 назовём *корректной*, если существует допустимая нумерация τ множества E_m, для которой Π_2 останавливается на (5).

Для дальнейших исследований нам нужны некоторые дополнительные определения.

Мы будем дальше использовать понятие значения переменной после выполнения линейной программы. Это значение определяется индукцией по длине программы.

Мы скажем для линейной программы Ξ с переменными из $X = \{z_1, \ldots, z_n\}$, что $\Xi(\sigma)$ есть NULL для заданной X-оценки σ, если Ξ не останавливается на σ.

Рассмотрим замкнутую линейную программу

$$\alpha_1; \ldots; \alpha_q$$

с переменными из $X = \{z_1, \ldots, z_n\}$.

Итак, для любых $i = 1, 2, \ldots, q$ и $j = 1, 2, \ldots, n$ линейная программа

$$\alpha_1; \ldots; \alpha_i$$

приписывает значение $\mu(\alpha_1; \ldots; \alpha_i)(z_j)$ переменной z_j. Это значение $\mu(\alpha_1; \ldots; \alpha_i)(z_j)$ или есть NULL, или есть некоторый терм, построенный из a, c, 0 и $LAST$ с помощью операций f и $'$.

Подробнее, сначала всем переменным присваиваются значения из множества $\{a,\ c,\ 0,\ LAST\}$. Полученная X-оценка отлична от NULL.

Если полученная X-оценка есть NULL, то каждая переменная получает значение NULL и следующая X-оценка есть NULL, каким бы ни были следующий тест или следующее присваивание.

Рассмотрим случай, когда полученная X-оценка отлична от NULL.

Тест не меняет значений переменных. Следующая X-оценка совпадает с рассматриваемой X-оценкой, если для рассматриваемой X-оценки этот тест выполняется. В противном случае следующая X-оценка есть NULL.

Любое присваивание вида $x \leftarrow u$ не меняет значений переменных, отличных от x. При этом следующая X-оценка отлична от NULL. Присваивание $x \leftarrow 0$ приписывает x значение 0. Присваивание $x \leftarrow LAST$ приписывает x значение $LAST$. Присваивание $x \leftarrow x'$ приписывает x значение t', если t — это старое значение переменной x. Присваивание $x \leftarrow y$ приписывает x значение t, если t — это старое значение переменной y. Наконец, присваивание $x \leftarrow (yz)$ приписывает x значение $(t_1 t_2)$, если t_1 — это старое значение переменной y и t_2 — это старое значение переменной z.

Получаемые значения переменных можно классифицировать следующим образом.

Стандартный терм — это a, номер или $(t_1 t_2)$, где t_1 и t_2 — стандартные термы. Номер — это 0 или t', где t — это стандартный терм. a-терм — это a или $(t_1 t_2)$, где t_1 и t_2 — a-термы.

С каждым стандартным термом мы сопоставляем некоторое множество стандартных термов. Если стандартный терм есть a, мы сопоставляем с ним множество $\{a\}$. Если стандартный терм есть номер n_1, мы сопоставляем с ним множество $\{n_1\}$. Мы сопоставляем со стандартным термом $(t_1 t_2)$ объединение множеств, сопоставленных с t_1 и с t_2. Если стандартному терму t сопоставлено множество S и $S = \{r_1, \ldots, r_\ell\}$, мы пишем $t(r_1, \ldots, r_\ell)$.

Заданная стандартная нумерация задаёт номер стандартного терма $(t_1 t_2)$ для заданных номеров стандартных термов t_1 и t_2. Поэтому каждый стандартный терм равен некоторому целому неотрицательному числу, не превосходящему $LAST$. С другой стороны, каждое такое число мы рассматриваем как номер, представляя это число как полученное из 0 подходящим числом применений операции $'$.

Пусть $i \in \{1, \ldots, \ell\}$. Результат замены номера r_i на $(t_1 t_2)$ в $t(r_1, \ldots, r_\ell)$ есть $(t_1 t_2)$, если $t(r_1, \ldots, r_\ell)$ есть r_i. Результат замены номера r_i на $(t_1 t_2)$ в $t(r_1, \ldots, r_\ell)$ есть $(t_5 t_6)$, если $t(r_1, \ldots, r_\ell)$ есть $(t_3 t_4)$, t_5 есть результат замены номера r_i на $(t_1 t_2)$ в t_3, а t_6 есть результат замены номера r_i на $(t_1 t_2)$ в t_4. $t(r_1, \ldots, r_{i-1}, (t_1 t_2), r_{i+1}, \ldots, r_\ell)$ обозначает результат замены номера r_i на $(t_1 t_2)$ в $t(r_1, \ldots, r_\ell)$.

Пусть

$$\Pi_2 = \alpha_1; \ldots; \alpha_p.$$

Для любых таких целых чисел i и j, что $0 < i \leqslant j \leqslant p$, пусть

$$\Pi_{2,i} = \alpha_1; \ldots; \alpha_i$$

и

$$\Pi_{2,i,j} = \alpha_i; \ldots; \alpha_j.$$

Прежде всего Π_2 можно выбрать так, что p не превосходит $n^2(F(nJ)4^I)^n$.

Действительно, если $0 < i < j \leqslant p$ и

$$\mu(\alpha_1; \ldots; \alpha_i)(z_\ell) = \mu(\alpha_1; \ldots; \alpha_j)(z_\ell)$$

для любого $\ell \in \{1, \ldots, n\}$ и тот же самый оператор того же самого члена объединения

$$(\Pi^1 + \cdots + \Pi^q)$$

является последним оператором как в $\Pi_{2,i}$, так и в $\Pi_{2,j}$, то можно выбросить

$$\alpha_{i+1}; \ldots; \alpha_j$$

из Π_2.

Если α_i является тестом $u = v$, $\mu(\Pi_{2,i})(u)$ есть $t_1{}'$, $\mu(\Pi_{2,i})(v)$ есть $t_2{}'$ для некоторых стандартных термов t_1 и t_2, то скажем, что из α_i следуют равенства $t_1{}' = t_2{}'$ и $t_2{}' = t_1{}'$. Если из α_i следует равенство $t_1{}' = t_2{}'$ для некоторых стандартных термов t_1 и t_2, то скажем, что из α_i следует равенство $t_1 = t_2$.

Если из α_i следует равенство $c = (t_1 t_2)$, а стандартные термы t_1 и t_2 равны числам n_1 и n_2 в рассматриваемой нумерации, n_1 отлично от номера c в рассматриваемой нумерации и $i \leqslant j \leqslant p$, то из $\Pi_{2,j}$ следует, что c равен стандартному терму $(n_1 n_2)$. В этом случае, если t_1 не является номером, то из $\Pi_{2,j}$ следует, что n_1 равен стандартному терму t_1, и, если t_2 не является номером, то из $\Pi_{2,j}$ следует, что n_2 равен стандартному терму t_2.

Если из α_i следует равенство $\ell = (t_1 t_2)$, в котором ℓ является номером, а стандартные термы t_1 и t_2 равны числам n_1 и n_2 в рассматриваемой нумерации, n_1 отлично от ℓ и $i \leqslant j \leqslant p$, то из $\Pi_{2,j}$ следует, что номер ℓ равен стандартному терму $(n_1 n_2)$. В этом случае, если t_1 не является номером, то из $\Pi_{2,j}$ следует, что n_1 равен стандартному терму t_1, и, если t_2 не является номером, то из $\Pi_{2,j}$ следует, что n_2 равен стандартному терму t_2. В этом случае, если t_1 есть a или t_2 есть a, то из $\Pi_{2,j}$ следует, что ℓ равняется a-терму (aa).

Пусть d либо есть c, либо является номером. Пусть $1 \leqslant j \leqslant p$. Если из $\Pi_{2,j}$ следует, что d равен $t(r_1, \ldots, r_k)$ и что r_i раняется стандартному терму $(t_1 t_2)$, то из $\Pi_{2,j}$ следует, что d равняется стандартному терму

$$t(r_1, \ldots, r_{i-1}, (t_1 t_2), r_{i+1}, \ldots, r_k).$$

Если из $\Pi_{2,j}$ следует, что c равняется стандартному терму $t(r_1, \ldots, r_k)$, $1 \leqslant i \leqslant k$ и r_i является номером элемента, расположенного на некоторой вершине некоторого уровня, мы скажем, что эта вершина этого уровня посещается линейной программой $\Pi_{2,j}$. Если при этом $\ell \leqslant j$ и номер элемента $\mu(\Pi_{2,\ell})(v)$ есть r_i для некоторой переменной v, мы говорим, что в момент ℓ эта вершина этого уровня посещается линейной программой $\Pi_{2,j}$.

Если $1 \leqslant \ell \leqslant \ell_1 \leqslant j$ и в момент ℓ некоторая вершина некоторого уровня посещается линейной программой $\Pi_{2,j}$, мы говорим, что и в момент ℓ_1 эта вершина этого уровня тоже посещается линейной программой $\Pi_{2,j}$. Мы также говорим, что эта вершина этого уровня помечена числом r_i в этом посещении. Элемент, номером которого является r_i, называется меткой этой вершины в этом посещении.

Каждая вершина бинарного дерева, отличная от корня, имеет в этом дереве одного непосредственного предка. Непосредственный предок является предком этой вершины. Непосредственный предок любого предка вершины тоже является предком этой вершины. Симметрично, непосредственный потомок является потомком вершины. Непосредственный потомок потомка вершины тоже является потомком этой вершины.

Если в момент, когда $\Pi_{2,j}$ посещает некоторую вершину, все предки этой вершины уже были посещены, это посещение этой вершины называется правым.

Посещение вершины, не являющееся правым, называется левым, если все потомки этой вершины уже были посещены. Понятно, что не бывает левого посещения никакой вершины, уровень которой меньше $m - n$. Иначе, для некоторых независимых $t_1, \ldots,$ t_{2^n} число $\mathrm{rc}(g(n)(t_1, \ldots, t_{2^n}))$ не превосходит n.

Если из $\Pi_{2,j}$ следует, что c равняется стандартному терму $t(a)$, мы говорим, что $\Pi_{2,j}$ посещает каждую висячую вершину дерева $g(nJ)$ для c и проверяет, что эта вершина помечена a для c. В этом случае мы также говорим, что $\Pi_{2,j}$ производит (или делает) полный обход дерева.

Ясно, что каждая корректная Π_2 обязана посетить каждую висячую вершину дерева $g(nJ)$ для c и проверить, что эта вершина помечена a для c. В противном случае можно было бы заменить метку этой висячей вершины таким образом, что для полученного элемента e линейная программа Π_2 останавливается на

$$(E_m,\ f_\tau,\ a,\ e,\ ',\ 0,\ LAST) \tag{6}$$

из $\mathfrak{K}_1$, но e не принадлежит подгруппоиду, порождённому a в $(E_m,\ f_m)$.

Для любых $k \in \{0,\ 1,\ \ldots,\ nJ-1\}$ и неотрицательного целого числа $\ell < 2^{nJ-k}$ каждая корректная Π_2, делая полный обход дерева, посещает ℓ-ую вершину уровня k, когда эта вершина помечена элементом $\bar{a}^{k,\ell}$.

Невозможно, чтобы α_i было тестом $u = v$ или $v = u$, $\mu(\Pi_{2,i})(u)$ было c и $\mu(\Pi_{2,i})(v)$ было a-термом. В противном случае $\mathrm{rc}(g(nJ)(t_1, \ldots, t_{2^m})) < n+1$ для некоторых независимых $t_1,\ldots,$ t_{2^m}.

Делая полный обход, Π_2 посещает все вершины дерева $g(nJ)$ для c. Если при этом Π_2 посещает вершину, расположенную на некотором уровне, мы говорим, что Π_2 спускается на этот уровень или что Π_2 посещает эту вершину при этом спуске.

Для Π_2 мы рассматриваем следующий параметр. Найдём число всех таких целых чисел i, что $0 < i \leqslant p$ и $\Pi_{2,i,p}$ начинает и делает полный обход. Это число назовём **числом полных обходов**, которые делает Π_2.

Для каждой допустимой нумерации τ из всех Π_2, останавливающихся на (5) для этой нумерации, мы выберем такую, которая делает наименьшее число полных обходов. Обозначим выбранную линейную программу через $\Pi_2(\tau)$.

Пусть $J = 2J_1$. Всего имеется $(4^I F(nJ))^n$ возможных наборов значений переменных $z_1, \ldots, z_n$. Всего имеется 2^{nJ_1} возможностей для первой посещённой программой Π_2 вершины уровня nJ_1, когда Π_2 делает первый полный обход дерева. Всего имеется $(4^I)^{nJ_1}$ возможных допустимых нумераций элементов вида $\bar{a}^{k,\ell}$, расположенных на предках этой первой посещённой вершины. Каждая такая нумерация порождается многими допустимыми нумерациями множества E_m. Но мы выберем для каждой допустимой нумерации элементов вида $\bar{a}^{k,\ell}$, расположенных на предках этой первой посещённой вершины, некоторую одну порождающую эту нумерацию этих элементов нумерацию элементов множества E_m.

Ясно, что для достаточно больших I и J имеется более

$$\frac{(4^I)^{nJ_1}}{2^{nJ_1+1}(4^I F(nJ))^n n^2}$$

таких попарно различных выбранных допустимых нумераций τ множества E_m, что для любой из этих нумераций та же самая вершина уровня nJ_1 впервые посещается выбранной линейной программой $\Pi_2(\tau)$ для этой нумерации τ, те же самые вершины меньших (nJ_1+1) уровней посещались до этого посещения и во время этого посещения каждая переменная имеет одно и то же значение для всех этих нумераций. Кроме того, для всех этих нумераций в момент первого посещения при первом полном обходе вершины уровня nJ_1 выполняется тот же самый оператор той же линейной программы из $\Pi^1,\ldots,\Pi^q$. Эти нумерации назовём хорошими.

При первом полном обходе имеется nJ_1 моментов, когда осуществляется очередной спуск. Между этими моментами происходит как построение меток вершин спуска, так и построение номеров, для которых делаются проверки. При этом проверкой называется тест $x = y$, из которого следует равенство $t_1 = t_2$, где стандартный терм t_1 является номером, а стандартный терм t_2 является a-термом регистровой сложности 1.

Если предположить, что при первом полном обходе сначала делаются абсолютно все проверки, а только потом начинается построение меток вершин для первого спуска, то закончить доказательство просто. После всех сделанных проверок имеется только $(4^I F(nJ))^n$ возможных наборов значений переменных $z_1, \ldots, z_n$. Поэтому имеется много хороших нумераций, для которых сразу после выполнения всех проверок выполняется один и тот же оператор одной и той же линейной программы из $\Pi^1,\ldots,\Pi^q$, а наборы значений переменных $z_1,\ldots,z_n$ в этот момент одинаковы. Среди них можно найти две различные такие τ и ρ, что начало первого полного обхода в линейной программы $\Pi_2(\rho)$ до первого посещения вершины уровня nJ_1 не содержит начала полного обхода в линейной программе $\Pi_2(\tau)$ до первого посещения вершины уровня nJ_1. Поэтому можно выбрать из этих нумераций две разные, заменить в первой из них в первом полном обходе часть от завершения всех проверок до первого посещения вершины уровня nJ_1 на соответствующую часть второй и получить для первой нумерации линейную программу из Π_1, делающую меньше полных обходов.

Для доказательства этого предположения нужно заметить, что проверка и спуск осуществляются в различных переменных, после чего можно проверки и спуски осуществлять различными линейными программами из $\Pi^1,\ldots,\Pi^q$.

Наша более слабая гипотеза состоит в том, что при первом полном обходе перед очередным спуском $\Pi_2(\tau)$ сначала делает все проверки, а только потом строит метки вершин для этого спуска.

В каждый из этих моментов сразу после очередного спуска и потом непосредственно перед началом построения меток вершин для следующего спуска выполняются какие-то операторы каких-то линейных программ из $\Pi^1,\ldots,\Pi^q$. Значит, имеется более

$$\frac{(4^I)^{nJ_1}}{2^{nJ_1+1}(4^I F(nJ))^n n^2 n^{4nJ_1}}$$

хороших нумераций, для которых в каждый из этих моментов выполняется один и тот же оператор той же самой из этих линейных программ. Такие нумерации назовём прекрасными.

Среди прекрасных нумераций выберем любую одну τ. После этого для каждой из прекрасных нумераций ρ в каждой из линейных программ $\Pi_2(\rho)$ заменим каждый кусок сразу после очередного спуска и до начала построения меток вершин для следующего спуска между двумя соседними моментами спуска при первом полном обходе на соответствующий кусок программы $\Pi_2(\tau)$. Полученную линейную программу обозначим через $\Pi_3(\rho,\tau)$. Ясно, что для любой прекрасной нумерации ρ линейная программа $\Pi_3(\rho,\tau)$ лежит в Π_1.

Для достаточно больших I и J среди линейных программ $\Pi_3(\rho,\tau)$ для всевозможных прекрасных нумераций ρ имеется много таких, для которых во время первого посещения вершины уровня nJ_1 при первом полном обходе каждая переменная имеет одно и то же значение.

Наша гипотеза состоит в том, что найдётся такая прекрасная нумерация τ, для которой среди таких находится и линейная программа $\Pi_2(\tau)=\Pi_3(\tau,\tau)$.

Если эта гипотеза выполняется, то завершение доказательства также просто.

В самом деле, начало $\Pi_2(\tau)$ можно заменить на соответствующее начало $\Pi_3(\rho,\tau)$. Это для нумерации τ строит линейную программу с меньшим, чем у $\Pi_2(\tau)$ числом полных обходов, но тоже останавливающуюся на (5). А это противоречит выбору $\Pi_2(\tau)$.

Другой подход может состоять в подсчёте числа проверок до первого посещения вершины уровня nJ_1 при первом полном обходе. Если число проверок ограничено функцией, зависящей от n, но не зависящей от I и J, то при достаточно больших I и J существует очень много хороших нумераций, осуществляющих одни и те же проверки. Опять среди них можно найти две различные такие ρ и τ, что начало первого полного обхода в линейной программы $\Pi_2(\rho)$ для первой из них не содержит начала полного обхода в линейной программе $\Pi_2(\tau)$ для второй из них, и заменить начало второй на начало первой. После этого получим линейную программу из Π_1, останавливающуюся на (5) для нумерации τ с меньшим числом полных обходов.

Если же никаких предположений не делать, то трудность состоит в том, что для любых разных хороших нумераций τ и ρ начало $\Pi_2(\rho)$ может делать проверки, противоречащие проверкам в $\Pi_2(\tau)$.

7 Благодарности

Я очень благодарен Сергею Михайловичу Дудакову, указавшему мне на ошибочность моих первоначальных примеров. Я благодарен профессору Stephen A. Cook за полезные замечания

и интерес к моей работе.

Список литературы

[1] S.A. Cook. Variations on pushdown machines. In *Proc. 1st ACM Symp. on Theory of Computing*, pages 229–232, 1969.

[2] S.A. Cook. Characterizations of push-down machines in terms of time-bounded computers. *Journal of the ACM*, 18(1):4–18, 1971.

[3] S.A. Cook and R. Sethi. Storage requirements for deterministic polynomial time recognizable languages. In *Proc. 6th ACM Symp. on Theory of Computing*, pages 33–39, 1974.

[4] S.A. Cook and R. Sethi. Storage requirements for deterministic polynomial time recognizable languages. *Journal of Computer and System Sciences*, 13(1):25–37, 1976.

[5] I.Kh. Musikaev and M.A. Taitslin. Limitations of the program memory and the expressive power of dynamic logics. *Information and Computation*, 103(2):195–203, 1993.

[6] И.Х. Мусикаев и М.А. Тайцлин. О динамических теориях свободных алгебр (On dynamic theories of free algebras). *Математический сборник (Math. USSR Sbornik)*, 180(66)(3 (2)):307–321 (313–327), 1989 (1990). Имеется перевод на английский.

[7] А.П. Столбоушкин и М.А. Тайцлин. Динамические логики. В.А. Мельников, редактор, *Кибернетика и вычислительная техника*, том 2, стр. 180–230. Наука, Москва, 1986.