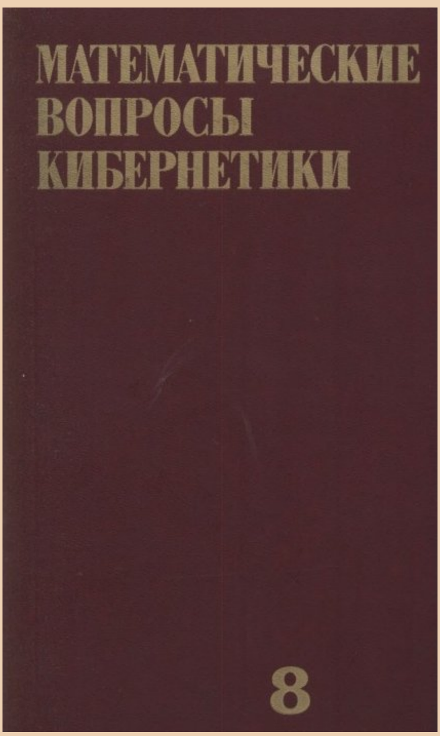


**В. А. Захаров**

# Об эффективной разрешимости проблемы эквивалентности линейных унарных рекурсивных программ

**Рекомендуемая форма библиографической ссылки:** Захаров В. А. Об эффективной разрешимости проблемы эквивалентности линейных унарных рекурсивных программ // Математические вопросы кибернетики. Вып. 8. — М.: Физматлит, 1999. — С. 255–272. URL: http://library.keldysh.ru/mvk.asp?id=1999-255

# ОБ ЭФФЕКТИВНОЙ РАЗРЕШИМОСТИ ПРОБЛЕМЫ ЭКВИВАЛЕНТНОСТИ ЛИНЕЙНЫХ УНАРНЫХ РЕКУРСИВНЫХ ПРОГРАММ*)

***В. А. ЗАХАРОВ***

(МОСКВА)

Наиболее общая формулировка проблемы эквивалентности такова: для произвольной пары программ в заданной модели вычислений требуется выяснить, обладают ли эти программы одинаковым поведением. Уточняя понятия вычислительной модели, программы и ее поведения, мы получаем разнообразные вариации проблемы эквивалентности. Некоторые варианты проблемы эквивалентности программ оказываются неразрешимыми. Имеются также многочисленные примеры вычислительных моделей с разрешимой проблемой эквивалентности. Построенные разрешающие процедуры явно или опосредованно используются при решении различных задач системного программирования, к числу которых относятся задачи трансляции, оптимизации и верификации вычислительных программ, разработка методов частичных вычислений, задачи специализации и повторного использования программ и др. Именно этим обусловлен непреходящий интерес к проблеме к эквивалентности и потребность в эффективных алгоритмах ее решения.

В настоящей работе исследуется проблема эквивалентности в одном классе рекурсивных программ. Впервые вычислительные модели такого вида были использованы Дж. Маккарти [28] для теоретического обоснования и анализа функциональных программ языка ЛИСП, разработанного этим автором. Ему удалось показать, что каждая операторная программа моделируется подходящей функциональной программой. Позднее Дж. де Баккер и Д. Скотт [17] ввели общее понятие рекурсивной программы и показали, опираясь на результаты Д. Лакхэма, Д. Парка и М. Патерсона [26] о неразрешимости проблемы эквивалентности стандартных операторных программ, что проблема эквивалентности неразрешима для полиадических рекурсивных программ. Было ясно также [22], что проблема эквивалентности разрешима в классе праволинейных унарных рекурсивных программ, поскольку программы такого вида моделируются схемами Янова [1, 6, 14]. Новые результаты [13, 16, 22] позволили существенно расширить класс рекурсивных программ, обладающих разрешимой проблемой эквивалентности. В [22] путем применения «техники следов» удалось установить разрешимость эквивалентности линейных унарных рекурсивных программ, не содержащих констант. Аналогичный результат в [13] получил В. К. Сабельфельд, построив полную систему эквивалентных преобразований для линейных унарных

*) Работа выполнена при поддержке Российского фонда фундаментальных исследований (код проекта 97-01-00975).

рекурсивных схем. В [16] был рассмотрен специальный класс так называемых свободных унарных рекурсивных программ. Проблема эквивалентности для этого класса была сведена к разрешимой проблеме пустоты недетерминированных автоматов с магазинной памятью. Во всех этих случаях были получены разрешающие процедуры экспоненциальной по времени сложности. Л. П. Лисовик [7, 8, 9, 10, 11, 25] исследовал проблемы разрешимости для различных рекурсивных преобразователей. Ему удалось установить разрешимость проблемы эквивалентности для металинейных унарных рекурсивных программ с засылкой константы и перестановочными операторами. Для решения этой задачи им была разработана новая техника свéдения исследуемой проблемы к задаче разрешимости линейных диофантовых уравнений (метод «жестких множеств» и полулинейного резервуара).

Естественная аналогия между рекурсивными схемами и магазинными автоматами, позволяющая во многих случаях перекачивать результаты из одной области теории вычислений в другую, широко использовалась при исследовании проблемы эквивалентности [12, 19, 21, 24, 32, 33]. В частности, Е. Фридман [20] установил взаимную сводимость проблем эквивалентности для унарных рекурсивных программ над свободными интерпретациями и детерминированных автоматов с магазинной памятью, а затем Ж. Сенизерг [30] смог доказать разрешимость последней. Обзор основных понятий и результатов теории схем рекурсивных программ представлен в [6, 18].

Здесь следует, однако, иметь в виду, что все упомянутые исследования (за исключением [9]) проводились только для рекурсивных программ, семантика которых была основана на свободных (эрбрановских) интерпретациях базовых функций и предикатов. В обзоре по теории рекурсивных программ [18] отмечалось, что до сих пор не известно никаких результатов о разрешимости проблемы эквивалентности рекурсивных программ для других специальных классов интерпретаций.

В настоящей работе предложен новый подход к проектированию простых и эффективных алгоритмов, разрешающих проблему эквивалентности в классе линейных унарных рекурсивных программ, семантика которых определяется в терминах теории полугрупп. Ключевая идея заключается в свéдении проблемы эквивалентности рекурсивных программ к проблеме тождеств в подходящей полугруппе и последующем «экономном» применении техники следов. Следуя предложенной в [4, 5, 34] методике, в случае эффективной разрешимости проблемы тождеств удается построить алгоритмы, разрешающие проблему эквивалентности рекурсивных программ за полиномиальное время. Статья организована следующим образом. В § 1 введены основные понятия теории рекурсивных программ. В § 2 описана процедура нормализации, унифицирующая структуру рекурсивных программ. В § 3 описан общий подход к построению алгоритмов, которые разрешают проблему эквивалентности линейных унарных рекурсивных программ в одном классе семантик, базирующихся на полугруппах, которые сохраняют длину термов. В § 4 показано, каким образом следует модифицировать общий метод для получения эффективных разрешающих процедур.

## § 1. Рекурсивные программы

В этом параграфе вводятся синтаксис и процедурная семантика линейных унарных рекурсивных программ (сокращенно ЛУРП), обсуждаются простейшие свойства вычислений этих программ и формализуется проблема эквивалентности.

**1.1. Синтаксис рекурсивных программ.** Введем необходимые определения. Фиксируем два конечных алфавита, $\mathcal{B}=\{a_1,\ldots,a_N\}$ и $\mathcal{C}=\{\delta^1,\ldots,\delta^M\}$, а также бесконечный алфавит $\mathcal{G}=\{F_0,F_1,\ldots\}$. Символы алфавитов $\mathcal{B}$ и $\mathcal{C}$ будем называть *базовыми операторами* и *логическими условиями* соответственно, а символы алфавита $\mathcal{G}$ — *главными функциональными символами* или *процедурами*. Слова в алфавите $\mathcal{B}\cup\mathcal{G}$ — *термами*. В частности, пустое слово также является термом; этот терм будем обозначать $\lambda$. Пустой терм, а также всякий терм $t$, образованный базовыми операторами, будем называть *базовым термом*. Множество термов (базовых термов) условимся обозначать Term (BTerm). Через $|t|$ обозначим длину терма $t$. Запись $f \in t$, где $f \in \mathcal{B}\cup\mathcal{G}$, $t \in \mathrm{Term}$, будет обозначать факт вхождения символа $f$ в состав терма $t$. Терм $t$ назовем *линейным*, если $t$ содержит не более одного вхождения главного функционального символа из $\mathcal{G}$.

Базовые операторы из $\mathcal{B}$ соответствуют константным функциональным символам [6, 16], или встроенным процедурам. Логические условия $\delta_i$, $1 \leqslant i \leqslant M$, представляют всевозможные элементарные конъюнкции $p_1^{\sigma_1}\&\ldots p_k^{\sigma_k}$, составленные из конечного набора атомарных формул $p_1, \ldots, p_k$, соответствующих встроенным предикатам (элементарным логическим условиям). Подобное истолкование позволяет считать все логические условия из $\mathcal{C}$ попарно несовместными.

*Описанием процедуры* назовем выражение $D$ вида

$$F\colon (\delta^1, t_1), (\delta^2, t_2), \ldots, (\delta^M, t_M), \tag{1}$$

состоящее из *заголовка* $F$, $F \in \mathcal{G}$, и списка *альтернатив* $(\delta^i, t_i)$, где $\delta^i \in \mathcal{C}$, $t_i \in \mathrm{Term}$, $1 \leqslant i \leqslant M$, по одной альтернативе для каждого логического условия $\delta^i$ из $\mathcal{C}$.

Приведенное здесь определение описания процедуры соответствует описанию процедуры в терминах [6, 16]:

$$\begin{aligned} F \Leftarrow \mathbf{if}\ \delta^1\ \mathbf{then}\ T_1 \\ \mathbf{else\ if}\ \delta^2\ \mathbf{then}\ T_2 \\ \mathbf{else}\ \ldots \\ \mathbf{else\ if}\ \delta^{M-1}\ \mathbf{then}\ T_{M-1}\ \mathbf{else}\ T_M. \end{aligned}$$

*Унарной рекурсивной программой* назовем систему

$$\pi = \langle T_0, D_1, D_2, \ldots, D_n \rangle,$$

состоящую из *запроса* $T_0$, $T_0 \in \mathrm{Term}$, и совокупности описаний рекурсивных процедур $D_1, D_2, \ldots, D_n$ с попарно различными заголовками. Множество всех главных функциональных символов, входящих в программу $\pi$, обозначим $\mathcal{G}_\pi$. Для главного функционального символа $F$, являющегося заголовком описания процедуры (1), и логического условия $\delta$ из $\mathcal{C}$ запись $T_\pi(F, \delta)$ будет обозначать терм $T$ из $\delta$-альтернативы $(\delta, T)$. Программу $\pi$ будем называть *линейной*, если ее запрос $T_0$, а также все термы, фигурирующие в описаниях процедур, — линейные. Под *сложностью* $|\pi|$ *программы* $\pi$ будем понимать суммарное количество символов, входящих в состав ее запроса и всех описаний процедур.

Предположим, что описание процедуры (1) с заголовком $F$ входит в состав рекурсивной схемы $\pi$, и $T = t'Ft''$ — некоторый терм, где $t'' \in \mathrm{BTerm}$ (т. е. $F$ — самый правый, или самый внутренний, главный функциональный символ, входящий в состав $T$). Тогда терм $T' = t'T_\pi(F, \delta)t''$, будем называть *$\delta$-преемником* $T$ в $\pi$, образованным в результате вызова $\delta$-альтернативы описания процедуры с заголовком $F$.

Будем говорить, что главный функциональный символ $F \in \mathcal{G}_\pi$ *непосредственно ссылается* на главный функциональный символ $F'$ (на пустой терм $\lambda$), если $F' \in T_\pi(F, \delta)$ (соответственно, если $T_\pi(F, \delta)$ — базовый терм) для некоторого условия $\delta$. Отношение непосредственной ссылки будем обозначать $F \uparrow F'$ или $F \uparrow \lambda$. Обозначим через $\uparrow^*$ транзитивное замыкание отношения непосредственной ссылки и будем говорить, что $F$ ссылается на $F'$

(или на $\lambda$), если имеет место отношение $F\uparrow^* F'$ (или $F\uparrow^* \lambda$). Множество символов, на которые ссылается $F$, обозначим $F\uparrow$.

Главный функциональный символ $F$, $F \in \mathcal{G}_\pi$, назовем

*автореферентным*, если $F \in F\uparrow^*$;

*существенно рекурсивным*, если $F\uparrow^*$ содержит автореферентный символ;

*обладающим свойством завершаемости*, если $\lambda \in F\uparrow^*$;

*граничным*, если $F\uparrow^*$ не содержит автореферентных символов.

Для иллюстрации введенных понятий рассмотрим в качестве примера рекурсивную программу, оценивающую баланс открывающих и закрывающих скобок в тексте, представленном в виде символьного списка. В терминах [6, 16] эта программа имеет вид

$$
\begin{aligned}
&F_0(x);\\
&F_0(x) \Leftarrow \ \mathbf{if}\ x = nil\ \mathbf{then}\ F_2(x)\ \mathbf{else}\ F_1(x)\\
&F_1(x) \Leftarrow \ \mathbf{if}\ head(x) = [\ \mathbf{then}\ plus1(F_0(tail(x)))\\
&\qquad\qquad \mathbf{else\ if}\ head(x) = ]\ \mathbf{then}\ minus1(F_0(tail(x)))\ \mathbf{else}\ F_0(tail(x));\\
&F_2(x) \Leftarrow zero(x).
\end{aligned}
$$

Здесь базовыми операторами служат функции выделения заголовка списка $head(x)$, выделения хвоста списка $tail(x)$, прибавления единицы $plus1(x)$, вычитания единицы $minus1(x)$ и функция нуля $zero(x)$. Обозначим эти операторы символами $a, b, c, d, f$ соответственно. В качестве атомарных формул $p_1, p_2, p_3$ используются отношения $x = nil$, $head(x) = [$ и $head(x) = ]$. Таким образом, алфавит $\mathcal{C}$ состоит из восьми логических условий $\delta^1, \ldots, \delta^8$, соответствующих всевозможным элементарным конъюнкциям $\bar{p}_1 \,\&\, \bar{p}_2 \,\&\, \bar{p}_3, \ldots p_1 \,\&\, p_2 \,\&\, p_3$, составленным из указанных атомов. В рамках упрощенного формализма унарных рекурсивных программ, используемого в настоящей работе, приведенная выше программа будет представлена в виде

$$
\begin{aligned}
&F_0;\\
&F_0\colon (\delta^1, F_1), (\delta^2, F_1), (\delta^3, F_1), (\delta^4, F_1), (\delta^5, F_2), (\delta^6, F_2), (\delta^7, F_2), (\delta^8, F_2);\\
&F_1\colon (\delta^1, F_0 b), (\delta^2, dF_0 b), (\delta^3, cF_0 b), (\delta^4, cF_0 b),\\
&\qquad (\delta^5, F_0 b), (\delta^6, dF_0 b), (\delta^7, cF_0 b), (\delta^8, cF_0 b);\\
&F_2\colon (\delta^1, f), (\delta^2, f), (\delta^3, f), (\delta^4, f), (\delta^5, f), (\delta^6, f), (\delta^7, f), (\delta^8, f).
\end{aligned}
$$

В программе фигурируют главные функциональные символы $F_0$, $F_1$, $F_2$. При этом $F_0, F_1$ являются автореферентными процедурами, обладающие свойством завершаемости, а $F_2$ является граничной процедурой.

*Трассой* в программе $\pi$ назовем всякую последовательность термов (конечную или бесконечную)

$$T_0, T_1, \ldots, T_k, T_{k+1}, \ldots, \tag{2}$$

которая начинается запросом $T_0$ программы $\pi$ и в которой каждый очередной член $T_{k+1}$, $k \geqslant 0$, является преемником предшествующего терма $T_k$. Если трасса (2) оканчивается термом $T$, условимся называть ее *терминальной* в случае $T \in \mathrm{BTerm}$, и *тупиковой* в случае, когда запрос $T$ не имеет преемников в $\pi$, не будучи при этом базовым термом. Очевидно, что если запрос $T_0$ представлен процедурой $F$, то главный функциональный символ $F$ содержится хотя бы в одной трассе только в том случае, когда $F' = F$ или $F' \in F\uparrow$. Кроме того можно отметить, что свойство завершаемости $F$ является необходимым условием того, что главный функциональный символ $F$ может фигурировать хотя бы в одной терминальной трассе.

**1.2. Семантика рекурсивных программ.** Семантика ЛУРП описывается посредством моделей пропозициональной динамической логики [23].

*Детерминированной динамической шкалой* (или просто *шкалой*) сигнатуры $\mathscr{B}$ назовем тройку $\mathscr{F} = \langle S, s_0, R \rangle$, состоящую из:

непустого *множества состояний* $S$,

*начального состояния* $s_0$, $s_0 \in S$,

*функции преобразования* $R$: $S \times \mathscr{B} \to S$.

Шкала $\mathscr{F}$ задает интерпретацию операторов программы. Областью интерпретации операторов является множество $S$, элементы которого играют роль состояний данных. Выделенное состояние $s_0$ понимается как начальное состояние данных.

Семантика операторов описывается функцией преобразования: результатом выполнения базового оператора $b$ из $\mathscr{B}$ на состоянии памяти $s$, $s \in S$, является состояние $s' = R(s, b)$. Мы полагаем при этом, что каждый базовый оператор $b$ применим к любому состоянию данных $s$ и результат его выполнения определяется состоянием $s$ однозначно. Эти ограничения традиционно соблюдаются во всех системах функционального программирования, поэтому мы будем иметь дело только с функциональными сериальными динамическими шкалами.

Функцию преобразования $R$ можно естественным образом распространить на все базовые термы BTerm, полагая $R^*(s, \lambda) = s$ и $R^*(s, tb) = R^*(R(s, b), t)$ для всякого базового терма $t$ из BTerm, базового оператора $b$ из $\mathscr{B}$ и состояния $s$ из $S$. Будем говорить, что состояние $s''$ *достижимо* на шкале $\mathscr{F}$ из состояния $s'$ (обозначая этот факт $s' \preceq_{\mathscr{F}} s''$), если $s'' = R^*(s', t)$ для некоторого терма $t$ из BTerm. Для каждого базового оператора $t$ обозначим через $[t]_{\mathscr{F}}$ состояние $s = R^*(s_0, t)$, достижимое на $\mathscr{F}$ из начального состояния посредством $t$.

При изучении вычислений пропозициональных операторных программ нас будут интересовать только те состояния данных, которые достижимы из начального состояния $s_0$, и поэтому, не ограничивая общности, будем полагать $S = \{[t]_{\mathscr{F}}\colon t \in \text{BTerm}\}$ для всех рассматриваемых в дальнейшем шкал. Кроме того, для упрощения записи мы будем опускать обозначение шкалы $\mathscr{F}$ в выражениях $[t]_{\mathscr{F}}$ и $s' \preceq_{\mathscr{F}} s''$, если эта шкала однозначно понимается из контекста.

Подобно программе, динамическая шкала может рассматриваться как ориентированный нагруженный граф, в общем случае бесконечный. Вершины этого графа соответствуют состояниям шкалы, а расположение и пометка дуг определяются функцией преобразования.

Достижимость одного состояния шкалы из другого означает наличие маршрута, связывающего эти состояния. Однако в рамках этой работы более выгодным представляется алгебраическое истолкование некоторых типов динамических шкал.

Шкалу $\mathscr{F}_s = \langle S', s, R' \rangle$ назовем *подшкалой* шкалы $\mathscr{F} = \langle S, s_0, R \rangle$, *порожденной состоянием* $s$, если $S' = \{R^*(s, t)\colon t \in \text{BTerm}\}$ и функция преобразования $R'$ получена сужением функции $R$ на множество состояний $S'$. Будем говорить, что шкала $\mathscr{F}$ является:

*полугрупповой*, если $\mathscr{F}$ можно гомоморфно отобразить на всякую ее подшкалу $\mathscr{F}_s$,

*однородной*, если $\mathscr{F}$ изоморфна всякой своей подшкале $\mathscr{F}_s$,

*упорядоченной*, если $\preceq$ является отношением строгого частичного порядка на множестве состояний $S$,

*уравновешенной*, если для любых операторных цепочек $t_1$ и $t_2$ равенство состояний $[t_1] = [t_2]$ влечет равенство длин термов $|t_1| = |t_2|$,

*универсальной*, если для любой пары термов $t_1$ и $t_2$ равенство состояний $[t_1] = [t_2]$ влечет равенство термов $t_1 = t_2$.

Полугрупповую шкалу $\mathcal{F} = \langle S, s_0, R \rangle$ можно рассматривать как конечно порожденный моноид $\langle S, * \rangle$ с множеством образующих $\{[b]\colon b \in \mathcal{B}\}$, единицей $s_0 = [\lambda]$ и бинарной операцией $*$, такой что $[t_1] * [t_2] = [t_1 t_2]$. В таком случае универсальная шкала соответствует свободному моноиду слов в алфавите $\mathcal{B}$, упорядоченная шкала — полугруппе с неразложимой единицей (т. е. моноиду, в котором для любых термов $t_1$, $t_2$ равенство состояний $[\lambda] = [t_1 t_2]$ влечет равенство термов $t_1 = t_2 = \lambda$), а однородная шкала — правосократимой полугруппе (т. е. моноиду, в котором для любых термов $t_1$, $t_2$, $t_3$ равенство состояний $[t_2 t_1] = [t_3 t_1]$ влечет равенство $[t_2] = [t_3]$). Очевидно, что уравновешенная шкала является упорядоченной.

*Детерминированной динамической моделью* (или просто *моделью*) $M$ сигнатуры $(\mathcal{B}, \mathcal{C})$ назовем пару $\langle \mathcal{F}, \xi \rangle$, такую что

$\mathcal{F} = \langle S, s_0, R \rangle$ — шкала сигнатуры $\mathcal{B}$,

$\xi\colon S \to \mathcal{C}$ — *функция означивания*, определяющая истинностные значения логических условий в каждом состоянии шкалы.

Поскольку логические условия являются попарно несовместными, предполагается, что в каждом состоянии в точности одно из условий $\delta \in \mathcal{C}$ может быть истинным.

На модели $M$ интерпретация операторов задается посредством шкалы $\mathcal{F}$, а интерпретация логических переменных — при помощи функции означивания $\xi$. В этом случае мы будем говорить, что модель $M$ *базируется* на шкале $\mathcal{F}$. Множество всех моделей, базирующихся на шкале $\mathcal{F}$, обозначим через $\mathcal{M}_{\mathcal{F}}$.

Трассу (2) назовем *вычислением схемы $\pi$ в модели $M$*, если эта последовательность термов удовлетворяет следующим условиям.

1. Для каждого $K$, $k \geqslant 0$, если $T_k = TFt$ и $t \in \mathrm{BTerm}$, то $T_{k+1}$ является $\xi([\hat{t}]_M)$-преемником $T_k$.

2. Трасса (2) является либо терминальной, либо бесконечной, либо тупиковой. В первом случае вычисление считается успешным, и его результатом служит состояние $[T]$, соответствующее последнему терму $T$ в данной трассе. В остальных случаях вычисление не имеет результата.

В том случае, когда трасса является началом некоторого вычисления в модели $M$, мы будем говорить, что эта трасса *реализуется в модели $M$*. Очевидно, всякая программа $\pi$ имеет единственное вычисление в заданной модели $M$, которое представлено максимальной трассой, реализуемой в модели $M$. В дальнейшем трассы вида (2) будем изображать следующим образом:

$$T_0 \xrightarrow{\delta_0} T_1 \xrightarrow{\delta_1} \ldots \xrightarrow{\delta_k} T_{k+1} \xrightarrow{\delta_{k+1}} \ldots$$

**1.3. Проблема эквивалентности рекурсивных программ** Программы $\pi_1$ и $\pi_2$ назовем *эквивалентными в модели $M$*, если $\pi_1$ и $\pi_2$ либо обе не имеют успешных вычислений в $M$, либо результаты их вычислений определены и совпадают. Программы $\pi_1$ и $\pi_2$ назовем *эквивалентными на шкале $\mathcal{F}$* ($\pi_1 \sim_{\mathcal{F}} \pi_2$), если они эквивалентны во всякой модели $M$, базирующейся на шкале $\mathcal{F}$. Очевидно, отношение эквивалентности УРП обладает следующими свойствами.

Утверждение 1. *Если шкала $\mathcal{F}_1$ является гомоморфным образом шкалы $\mathcal{F}_2$, то для любой пары программ $\pi_1$, $\pi_2$ их эквивалентность $\pi_1 \sim_{\mathcal{F}_2} \pi_2$ на шкале $\mathcal{F}_2$ влечет эквивалентность $\pi_1 \sim_{\mathcal{F}_1} \pi_2$ этих программ на шкале $\mathcal{F}_1$.*

Утверждение 2. *Если программы $\pi_1$, $\pi_2$ эквивалентны на универсальной шкале, то эти программы эквивалентны на всякой шкале $\mathcal{F}$.*

## § 2. Нормализация унарных рекурсивных программ

Будем говорить, что программа

$$\pi = \langle T_0, D_0, D_1, D_2, \ldots, D_N \rangle,$$

представлена в *нормальной форме*, если ее структура удовлетворяет следующим условиям:

запрос $T_0$ является главным функциональным символом;

в множестве $\mathscr{G}_\pi$ выделен специальный функциональный символ $F_{\inf}$, описание которого имеет вид

$$F_{\inf}\colon (\delta_1, F_{\inf}a), (\delta_2, F_{\inf}a), \ldots, (\delta_M, F_{\inf}a),$$

где $a$ — некоторый базовый оператор;

все прочие главные функциональные символы $F_j$, $F_j \in \mathscr{G}_\pi$, $F_j \neq F_{\inf}$, обладают свойством завершаемости, причем для каждого условия $\delta$ терм $T_\pi(F_j, \delta)$ либо является базовым термом, либо имеет вид $tFb$, где $F$ — главный функциональный символ, а $b$ — базовый оператор.

Теорема 1. *Для любой унарной рекурсивной программы $\pi$ существует представленная в нормальной форме программа $\pi'$, эквивалентная $\pi$ на универсальной шкале.*

Доказательство. Предположим, что программа $\pi$ представлена запросом $T_0$ и описаниями процедур $D_0, D_1, D_2, \ldots, D_N$. Построим для программы $\pi$ помеченный граф несущественных ссылок $D_\pi^\uparrow$ следующего вида. Вершинами графа являются главные функциональные символы программы. Вершина $F$ считается связанной с вершиной $F'$ дугой, помеченной условием $\delta$ в том и только том случае, когда терм $T(F, \delta)$ оканчивается главным функциональным символом $F'$. Альтернативу $(\delta, T)$ в описании $F$ назовем *несущественной*, если в графе $D_\pi^\uparrow$ из вершины $F$ исходит бесконечный маршрут, состоящий из дуг, помеченных $\delta$. В противном случае альтернатива $(\delta, T)$ считается *существенной*. Очевидно, что вызов всякой несущественной альтернативы приводит вычисление программы к зацикливанию.

Укажем теперь последовательность преобразований, приводящих всякую программу $\pi$ к нормальной форме.

1. Переименуем главные функциональные символы программы так, чтобы $F_{\inf} \notin \mathscr{G}_\pi$.

2. Добавим к программе описание процедуры $D_{\inf}$

$$F_{\inf}\colon (\delta^1, F_{\inf}a), (\delta^2, F_{\inf}a), \ldots, (\delta^M, F_{\inf}a),$$

для главного функционального символа $F_{\inf}$, представляющего нигде не определенную унарную функцию.

3. Заменим терм-запрос $T_0$ запросом $F_0$, где $F_0$ — главный функциональный символ, не содержащийся в $\mathscr{G}_\pi$, добавив при этом к программе описание процедуры $D_0$

$$F_0\colon (\delta^1, T_0a), (\delta^2, T_0a), \ldots, (\delta^M, T_0a).$$

4. Ко всем альтернативам $(\delta, T)$, содержащимся в описаниях процедур $D_0, D_1, D_2, \ldots, D_N$, будем применять до тех пор, пока это возможно, следующее преобразование: если альтернатива $(\delta, T)$ в описании процедуры с заголовком $F$ содержит терм $T$, $T = T'F'$, оканчивающийся главным функциональным символом $F'$, то

в случае несущественности альтернативы $(\delta, T)$ следует заменить в ней терм $T$ на терм $F_{\inf}$;

в случае существенности альтернативы $(\delta, T)$ следует заменить в ней терм $T$ на терм $T'T_\pi(F', \delta)$.

5. Ко всем альтернативам, содержащимся в описаниях процедур программы, будем применять до тех пор, пока это возможно, следующее преобразование: если терм $T_\pi(F', \delta)$ имеет вид $tF't'b$, где $F' \in \mathcal{G}$, $t' \in \mathrm{BTerm}$, $t' \neq \lambda$, $b \in \mathcal{B}$, то, введя новый главный функциональный символ $F''$, заменим $T_\pi(F', \delta)$ на терм $tF''b$ и добавим к программе описание процедуры

$$F'': (\delta^1, F''t'), (\delta^2, F''t'), \ldots, (\delta^M, F''t')$$

с заголовком $F''$.

6. Удалим из программы описания всех процедур, заголовки $F$ которых отличны от $F_{\mathrm{inf}}$ и не обладают свойством завершаемости.

7. Во всех альтернативах описаний процедур заменим все термы, содержащие главные функциональные символы $F'$, которые не являются заголовками ни одной из процедур программы на терм $F_{\mathrm{inf}}$.

8. Удалим из программы описания всех процедур, заголовки $F$ которых не принадлежат множеству $F_0\uparrow^*$.

Каждое из перечисленных преобразований оставляет неизменным множество результативных выражений программы на универсальной шкале. Поэтому программа $\pi'$, полученная в результате применения этих преобразований, будет эквивалентна исходной программе $\pi$ на универсальной шкале.

Преобразования 1 и 2 вводят в программу специальную процедуру, обеспечивающую бесконечные безрезультатные вычисления. Преобразование 3 унифицирует запрос к программе. После применения преобразований 4 и 5 все непустые термы в альтернативах будут заканчиваться базовыми операторами. Преобразования 6–8 удаляют из программы описания процедур, не оказывающих влияния на успешные вычисления. При этом удаляются и все те процедуры, заголовки которых не обладают свойством завершаемости. Очевидно, программа $\pi'$, полученная в результате применения этих преобразований, будет представлена в нормальной форме.

Следует заметить, что при нормализации размер программы увеличивается не более чем в квадратичное число раз, Легко видеть также, что линейные программы при нормализации преобразуются в линейные программы. В дальнейшем без ограничения общности мы будем рассматривать только нормализованные рекурсивные программы. Нормальная форма позволяет унифицировать структуру рекурсивных программ и упростить тем самым их последующий анализ, о чем свидетельствует

Теорема 2. *Если унарная рекурсивная программа $\pi$ представлена в нормальной форме, то для всякой упорядоченной шкалы $\mathcal{F}$ любая трасса программы $\pi$ реализуется на некоторой модели $M$, базирующейся на шкале $\mathcal{F}$.*

Доказательство. Рассмотрим произвольную трассу программы $\pi$

$$T_0, T_1, \ldots, T_k, T_{k+1}, \ldots, T_N, \ldots,$$

в которой каждый терм $T_{i+1}$, $i \geqslant 0$, является $\delta_i$-последователем терма $T_i$. Согласно определению трассы, всякий терм $T_i$, входящий в эту трассу, либо является базовым термом $t_i$, либо имеет вид $T_i'Ft_i$, где $F$ — один из главных функциональных символов, а $t_i$ — базовый терм. Поскольку все термы, фигурирующие в альтернативах описаний процедур нормализованной программы $\pi$, оканчиваются базовыми операторами, каждый из базовых термов $t_i$, $i \geqslant 0$, содержится в качестве собственного суффикса в терме $t_{i+1}$. В случае упорядоченной шкалы $\mathcal{F}$ это приводит к тому, что для любой пары указанных термов $t_i$, $t_j$, $i \neq j$, выполняется $[t_i] \neq [t_j]$. Таким образом, выбрав функцию означивания $\xi$ такую, что $\xi([t_i]) = \delta_i$, $i \geqslant 0$, мы получим модель $M = \langle \mathrm{F}, \xi \rangle$, реализующую рассматриваемую трассу.

Из теоремы 2 в качестве следствий могут быть извлечены следующие два утверждения, которые мы будем широко использовать при проектировании алгоритмов, распознающих эквивалентность программ на полугрупповых уравновешенных и упорядоченных шкалах.

Утверждение 3. *Трассы*

$$T_0' \xrightarrow{\delta_0'} T_1'F_1't_1' \xrightarrow{\delta_1'} \ldots \xrightarrow{\delta_{i-1}'} T_i'F_i't_i' \xrightarrow{\delta_i'} T_{i+1}'F_{i+1}'t_{i+1}' \xrightarrow{\delta_{i+1}'} \ldots \xrightarrow{\delta_{k-1}'} T_k'F_k't_k'$$

$$T_0'' \xrightarrow{\delta_0''} T_1''F_1''t_1'' \xrightarrow{\delta_1''} \ldots \xrightarrow{\delta_{j-1}''} T_j''F_j''t_j'' \xrightarrow{\delta_j''} T_{j+1}''F_{j+1}''t_{j+1}'' \xrightarrow{\delta_{j+1}''} \ldots \xrightarrow{\delta_{l-1}''} T_l''F_l''t_l''$$

*в нормализованных программах $\pi_1$ и $\pi_2$ могут быть совместно реализованы на некоторой модели $M$, базирующейся на упорядоченной шкале $\mathcal{F}$ в том и только том случае, когда для всякой пары $i, j$, $0 \leqslant i \leqslant k$, $0 \leqslant j \leqslant l$, совпадение состояний $[t_i'] = [t_j'']$ влечет равенство условий $\delta_i' = \delta_j''$.*

Утверждение 4. *Трассы*

$$T_0' \xrightarrow{\delta_0'} T_1'F_1't_1' \xrightarrow{\delta_1'} \ldots \xrightarrow{\delta_{i-1}'} T_i'F_i't_i' \xrightarrow{\delta_i'} T_{i+1}'F_{i+1}'t_{i+1}' \xrightarrow{\delta_{i+1}'} \ldots \xrightarrow{\delta_{k-1}'} T_k'F_k't_k'$$

$$T_0'' \xrightarrow{\delta_0''} T_1''F_1''t_1'' \xrightarrow{\delta_1''} \ldots \xrightarrow{\delta_{i-1}''} T_i''F_i''t_i'' \xrightarrow{\delta_i''} T_{i+1}''F_{i+1}''t_{i+1}'' \xrightarrow{\delta_{i+1}''} \ldots \xrightarrow{\delta_{l-1}''} T_l''F_l''t_l''$$

*в нормализованных программах $\pi_1$ и $\pi_2$ могут быть совместно реализованы на некоторой модели $M$, базирующейся на уравновешенной шкале $\mathcal{F}$ в том и только том случае, когда для любого $i$, $0 \leqslant i \leqslant \min(k, l)$, совпадение состояний $[t_i'] = [t_i'']$ влечет равенство условий $\delta_i' = \delta_i''$.*

## § 3. Разрешимые случаи для проблемы эквивалентности линейных унарных рекурсивных программ

В этом параграфе представлен новый подход к построению алгоритмов, разрешающих проблему эквивалентности ЛУРП на уравновешенных полугрупповых шкалах. Ранее этот подход успешно применялся для анализа эквивалентности пропозициональных операторных программ [4, 5, 34]. Его основная идея состоит в следующем. Для заданной шкалы $\mathcal{F}$ сначала выбирается специальная полугруппа $W$, элементы которой используются для кодирования пар $\langle s', s''\rangle$ состояний шкалы $\mathcal{F}$. Затем на основе этого кодирования для исследуемых ЛУРП $\pi_1$ и $\pi_2$ конструируется графовая структура $\Gamma(\pi_1, \pi_2)$, представляющая все возможные пары вычислений $r(\pi_1, M)$, $r(\pi_2, M)$ программ $\pi_1$ и $\pi_2$ на моделях шкалы $\mathcal{F}$. Мы покажем, что для решения вопроса об эквивалентности программ достаточно рассмотреть лишь конечный фрагмент графа $\Gamma(\pi_1, \pi_2)$. Таким образом, проблему эквивалентности ЛУРП на шкале $\mathcal{F}$ можно свести к проблеме тождеств «$w' = w''$?» на полугруппе $W$, решение которой используется при построении структуры $\Gamma(\pi_1, \pi_2)$. Если эта проблема имеет эффективное решение, а $W$ является группой, то, как будет показано в § 4, проблема эквивалентности ЛУРП на $\mathcal{F}$ разрешима за полиномиальное время. Применение новой техники разрешения проблемы эквивалентности ЛУРП мы продемонстрируем на примерах шкал, порожденных свободными моноидами, а также свободными коммутативными и частично коммутативными моноидами.

Рассмотрим полугрупповую шкалу $\mathcal{F} = \langle S, s_0, R\rangle$. Запись $\mathcal{F} \times \mathcal{F}$ мы будем использовать для обозначения моноида, являющегося декартовым квадратом моноида, ассоциированного со шкалой $\mathcal{F}$.

Определение. Пусть $W$ — конечно порожденный моноид с бинарной операцией $\circ$ и единицей $e$, $U$ — его подмоноид, $w^+$ и $w^*$ — выделенные элементы моноида $W$. Четверку $K = \langle W, U, w^+, w^*\rangle$ назовем *критериальной системой* для $\mathcal{F}$, если $K$ и $\mathcal{F}$ удовлетворяют следующим требованиям.

Тр1. Существует такой гомоморфизм $\varphi$, отображающий $\mathcal{F} \times \mathcal{F}$ в $U$, что для всякой пары состояний $\langle s_1, s_2\rangle$ шкалы $\mathcal{F}$ выполняется соотношение

$$s_1 = s_2 \Longleftrightarrow w^+ \circ \varphi(\langle s_1, s_2\rangle) \circ w^* = e.$$

Тр2. Для всякого элемента $w$ из левого (правого) класса смежности $U \circ w^*$ ($w^+ \circ U$) уравнение $X \circ w = e$ (соответственно, $w \circ X = e$) имеет не более одного решения $X$ в классе смежности $w^+ \circ U$ ($U \circ w^*$).

Следует заметить, что требование Тр2 всегда соблюдается в том случае, когда моноид $W$ является группой.

Пусть $K = \langle W, U, w^+, w^*\rangle$ — критериальная система для полугрупповой шкалы $\mathcal{F}$, а $\pi_1$ и $\pi_2$ — нормализованные ЛУРП. Не ограничивая общности, мы будем полагать, что программы $\pi_1$ и $\pi_2$ не имеют общих главных функциональных символов. Построим ориентированный нагруженный граф $\Gamma(\pi_1, \pi_2)$. Вершинами графа являются четверки вида $(H^1, H^2, u, w)$, где $H^i \in \mathcal{G}_{\pi_i} \cup \{\lambda\}$, $i = 1, 2$, а $u$ и $w$ — элементы классов смежности $w^+ \circ U$ и $U \circ w^*$ соответственно. Вершину $(G^1, G^2, w^+, w^*)$, где $G^1$ и $G^2$ — запросы ЛУРП $\pi_1$ и $\pi_2$, назовем *корнем* графа $\Gamma(\pi_1, \pi_2)$. Множество вершин графа разобьем на классы $X_1$, $X_2$ и $X_3$, полагая

$$X_1 = \{(H^1, H^2, u, w)\colon\ w^+ \circ w \neq e,\ H^i \in \mathcal{P}_{\pi_i},\ i = 1, 2\},$$

$$X_2 = \{(H^1, H^2, u, w)\colon\ w^+ \circ w = e \text{ или } H^i \in \mathcal{P}_{\pi_i},\ H^{3-i} = \lambda,\ i \in \{1, 2\}\};$$

все прочие вершины графа отнесем к классу $X_3$.

Дуги графа $\Gamma(\pi_1, \pi_2)$ помечены парами $(\delta_1, \delta_2)$ из $\mathcal{C} \times \mathcal{C}$. Для каждой вершины $x$ графа $\Gamma(\pi_1, \pi_2)$ определим множество $\Delta_x$ допустимых пар следующим образом:

$$\Delta_x = \begin{cases} \{(\delta_1, \delta_2)\colon\ \delta_1, \delta_2 \in \mathcal{C}\}, & x \in X_1, \\ \{(\delta, \delta)\colon\ \delta \in \mathcal{C}\}, & x \in X_2, \\ \varnothing, & x \in X_3. \end{cases}$$

Каждая вершина $x$ из $X_1$ имеет $|\mathcal{C}|^2 = M^2$ исходящих дуг, каждая вершина $y$ из $X_2$ имеет $|\mathcal{C}| = M$ исходящих дуг, помеченных всевозможными допустимыми условиями из $\Delta_x$, а из вершин класса $X_3$ дуги не выходят. Дуги соединяют вершины графа $\Gamma(\pi_1, \pi_2)$ в соответствии со следующими правилами.

Пусть $x = (F_1, F_2, u, w)$ — вершина, принадлежащая множеству $X_1 \cup X_2$, и пусть $(\delta_1, \delta_2) \in \Delta_x$. Тогда дуга, помеченная $(\delta_1, \delta_2)$, ведет из $x$ в вершину $x' = (H_1, H_2, u \circ \varphi(\langle [g_1], [g_2]\rangle), \varphi(\langle h_1, h_2\rangle) \circ w)$, параметры $H_k, g_k, h_k$, $k = 1, 2$, которой удовлетворяют следующим условиям:

1. если терм $T(F_k, \delta_k)$ имеет вид $TFb$, где $T \in \mathrm{BTerm}$, $F \in \mathcal{G}_{\pi_k}$, $b \in \mathcal{B}$, то $H_k = F$, $g_k = T$, $h_k = b$;
2. если $T(F_k, \delta_k)$ — базовый терм $T$, то $H_k = h_k = \lambda$, $g_k = T$;
3. если же $F_k = \lambda$, то $H_k = g_k = h_k = \lambda$.

В дальнейшем, не оговаривая этого особо, будем полагать, что полугрупповая шкала $\mathcal{F}$ является уравновешенной, программы $\pi_1$ и $\pi_2$ нормализованы и имеют главные функциональные символы $G^1$ и $G^2$ в качестве запросов. Мы будем подразумевать также, что четверка $K = \langle W, U, w^+, w^*\rangle$ служит критериальной системой для $\mathcal{F}$.

Лемма 1. *Пусть $x_0, x_1, \ldots, x_{m+1}$, $m \geqslant 0$, — конечная последовательность вершин графа $\Gamma(\pi_1, \pi_2)$, в которой $x_0$ — корень $\Gamma(\pi_1, \pi_2)$, и для каждого $i$, $i = 1, \ldots, m+1$, вершина $x_i$ имеет вид $(F_i^1, F_i^2, u_i, w_i)$. Тогда в критериальном графе $\Gamma(\pi_1, \pi_2)$ указанная последовательность вершин образует ориентированный путь*

$$x_0 \xrightarrow{(\delta_0^1, \delta_0^2)} x_1 \xrightarrow{(\delta_1^1, \delta_1^2)} \ldots \xrightarrow{(\delta_m^1, \delta_m^2)} x_{m+1}$$

*в том и только том случае, когда в некоторой модели $M = \langle \mathcal{F}, \xi \rangle$ шкалы $\mathcal{F}$ программы $\pi_1$ и $\pi_2$ имеют вычисления*

$$G^1 \xrightarrow{\delta_0^1} T_1^1 F_1^1 t_1^1 \xrightarrow{\delta_1^1} \ldots \xrightarrow{\delta_{m-1}^1} T_m^1 F_m^1 t_m^1 \xrightarrow{\delta_m^1} T_{m+1}^1 F_{m+1}^1 t_{m+1}^1$$

$$G^2 \xrightarrow{\delta_0^2} T_1^2 F_1^2 t_1^2 \xrightarrow{\delta_1^2} \ldots \xrightarrow{\delta_{m-1}^2} T_m^2 F_m^2 t_m^2 \xrightarrow{\delta_m^2} T_{m+1}^2 F_{m+1}^2 t_{m+1}^2$$

*причем для всякого $i$, $1 \leqslant i \leqslant m+1$, выполняются соотношения*

$$u_i = w^+ \circ \varphi(\langle [T_i^1], [T_i^2] \rangle), \qquad w_i = \varphi(\langle [t_i^1], [t_i^2] \rangle) \circ w^*.$$

Доказательство проводится индукцией по $m$ с использованием теоремы 2, утверждения 4 и определения графа $\Gamma(\pi_1, \pi_2)$.

Лемма 1 устанавливает, что ориентированные пути в критериальном графе $\Gamma(\pi_1, \pi_2)$ взаимосвязаны с парами вычислений программ $\pi_1$ и $\pi_2$, совместно реализуемых на моделях шкалы $\mathcal{F}$. Эта особенность критериального графа будет широко использоваться в последующих рассуждениях.

Лемма 2. *Пусть задана некоторая последовательность логических условий $\delta_1, \delta_2, \ldots, \delta_m$ и пара главных функциональных символов $F_0^1$, $F_0^2$ из множеств $\mathcal{G}_{\pi_1}$, $\mathcal{G}_{\pi_2}$. Предположим, что в программах $\pi_k$, $k = 1, 2$, каждая из процедур $F_0^k$ порождает последовательность главных функциональных символов $F_0^k, F_1^k, \ldots, F_m^k$ такую, что $T_{\pi_k}(F_{i-1}^k, \delta_i) = T_i^k F_i^k a_i^k$ для всякого $i$, $0 \leqslant i \leqslant m$. Тогда в графе $\Gamma(\pi_1, \pi_2)$ из каждой вершины вида $x_0 = (F_0^1, F_0^2, u, w)$ выходит ориентированный путь*

$$x_0 \xrightarrow{(\delta_1, \delta_1)} x_1 \xrightarrow{(\delta_2, \delta_2)} \ldots \xrightarrow{(\delta_m, \delta_m)} x_m,$$

*ведущий в вершину $x_m = (F_m^1, F_m^2, u', w')$, такую что*

$$u' = u \circ \varphi(\langle [T^1], [^2] \rangle) \quad w' = \varphi(\langle [t^1], [t^2] \rangle) \circ w,$$

*где $t^k = a_m^k \ldots a_2^k a_1^k$, $T^k = T_1^k T_2^k \ldots T_m^k$, $k = 1, 2$.*

Доказательство этой леммы следует непосредственно из определения критериального графа $\Gamma(\pi_1, \pi_2)$. Существенную роль здесь играет то обстоятельство, что дуги в указанном пути помечены парами, состоящими из одинаковых логических условий. По этой причине указанный в лемме ориентированный путь в критериальном графе всегда существует.

Совершенно аналогично устанавливается

Лемма 3. *Пусть задана некоторая последовательность логических условий $\delta_1, \delta_2, \ldots, \delta_m$ и главный функциональный символ $F_0^1$ из множества $\mathcal{G}_{\pi_1}$. Предположим, что в программе $\pi_1$, процедура $F_0^k$ порождает последовательность главных функциональных символов $F_0^1, F_1^1, \ldots, F_m^1$ такую, что $T_{\pi_k}(F_{i-1}^k, \delta_i) = T_i F_i^k a_i$ для всякого $i$, $0 \leqslant i \leqslant m$. Тогда в графе $\Gamma(\pi_1, \pi_2)$ из каждой вершины вида $x_0 = (F_0^1, \lambda, u, w)$ выходит ориентированный путь*

$$x_0 \xrightarrow{(\delta_1, \delta_1)} x_1 \xrightarrow{(\delta_2, \delta_2)} \ldots \xrightarrow{(\delta_m, \delta_m)} x_m,$$

*ведущий в вершину $x_m = (F_m^1, \lambda, w', u')$, такую что*

$$u' = u \circ \varphi(\langle [T^1], [\lambda] \rangle) \quad w' = \varphi(\langle [t^1], [\lambda] \rangle) \circ w,$$

*где $t^k = a_m^k \ldots a_2^k a_1^k$, $T^k = T_1^k T_2^k \ldots T_m^k$, $k = 1, 2$.*

Пусть $x$, $x = (H^1, H^2, u, w)$, — вершина графа $\Gamma(\pi_1, \pi_2)$. Назовем ее *опровергающей*, если выполнено одно из двух следующих условий:

1. $H^1 = H^2 = \lambda$ и $u \circ w \neq e$;

2. один из элементов $H^1$ и $H^2$ равен $\lambda$, тогда как другой является существенно рекурсивным главным функциональным символом.

Вершину $x=(F^1, F^2, u, w)$ назовем *граничной*, если каждый из термов $F^1$, $F^2$ является либо пустым термом $\lambda$, либо граничным главным функциональным символом в соответствующей программе.

Лемма 4. *Соотношение $\pi_1 \sim_{\mathcal{F}} \pi_2$ выполнено в том и только том случае, когда ни одна опровергающая вершина не достижима из корня графа $\Gamma(\pi_1, \pi_2)$.*

Доказательство. Необходимость. Допустим противное. Тогда из корня графа $\Gamma(\pi_1, \pi_2)$ достижима либо некоторая вершина $x_1=(\lambda, \lambda, u, w)$, где $u \circ w \neq e$, либо некоторая вершина $x_2=(F^1, F^2, u, w)$, в которой один из элементов $F^1$ и $F^2$ (например, $F^2$) равен $\lambda$, тогда как другой есть имя существенно рекурсивной главной функции. В первом случае согласно лемме 1 существует модель $M$ шкалы $\mathcal{F}$, в которой программы $\pi_1$ и $\pi_2$ имеют вычисления, завершающиеся с такими результатами $s^1$ и $s^2$ соответственно, что $u \circ w = w^+ \circ \varphi(\langle s^1, s^2\rangle) \circ w^* \neq e$. Но в соответствии с требованием Тр1 отсюда следует, что $s^1 \neq s^2$. Во втором случае рассмотрим, пользуясь существенной рекурсивностью $F^1$, бесконечную последовательность логических условий $\delta_1, \ldots, \delta_m, \ldots$, порождающую бесконечную последовательность ссылок главного функционального символа $F^1$:

$$F_0^1, F_1^1, F_2^1, \ldots, F_i^1, F_{i+1}^1, \ldots,$$

такую, что $F_0^1 = F^1$, и для всякого $i$, $i \geqslant 0$, $F_{i+1}^1$ — главный функциональный символ терма $T_{\pi_1}(F_i^1, \delta_{i+1})$. Согласно лемме 3 из вершины $x_2$ выходит бесконечный путь, дуги которого помечены парами $(\delta_1, \delta_1), \ldots, (\delta_i, \delta_i), \ldots$, проходящий через вершины $(F^1, \lambda, u_1, w_1), \ldots, (F^1, \lambda, u_i, w_i), \ldots$ Тогда по лемме 1 существует модель $M$ шкалы $\mathcal{F}$, в которой программа $\pi_2$ имеет завершающееся вычисление, а $\pi_1$ — бесконечное вычисление.

Достаточность. Допустим, что программы $\pi_1$, $\pi_2$ не эквивалентны в некоторой модели $M$ шкалы $\mathcal{F}$. Согласно лемме 1 их вычисления порождают некоторый путь в графе $\Gamma(\pi_1, \pi_2)$, выходящий из корня. Если оба вычисления завершаются, но порождают различные результаты $s^1$ и $s^2$, то по той же лемме 1 этот путь оканчивается в такой вершине $x_1 = (\lambda, \lambda, u, w)$, что $u \circ w = w^+ \circ \varphi(\langle s^1, s^2\rangle) \circ w^*$. Учитывая соотношение $s^1 \neq s^2$ и требование Тр1, приходим к выводу о том, что $u \circ w \neq e$, следовательно, $x_1$ — опровергающая вершина. Если же одно из вычислений завершается с некоторым результатом, а другое — бесконечное, то это значит, что на этом пути располагается вершина $y=(F^1, F^2, u, w)$, один из элементов $F^1$ или $F^2$ — пустой терм $\lambda$, а другой — главный функциональный символ, имеющий бесконечную цепочку ссылок, т. е. существенно рекурсивный главный функциональный символ. Это означает, что $y$ — также опровергающая вершина.

Лемма 5. *Пусть один из главных функциональных символов $F^1$ или $F^2$ обладает свойством завершаемости, вершина $x=(F^1, F^2, u, w)$ достижима в графе $\Gamma(\pi_1, \pi_2)$ из корня и для всякого элемента $v$ выделенного подмоноида $U$ выполняется соотношение $u \circ v \circ w \neq e$. Тогда некоторая опровергающая вершина достижима из корня графа $\Gamma(\pi_1, \pi_2)$.*

Доказательство. Предположим, что свойством завершаемости обладает $F^2$. Тогда согласно утверждению 4 и леммам 1, 2 вершина $y=(H, \lambda, u', w')$ достижима из $x$ по некоторому ориентированному пути, дуги которого помечены парами $(\delta_1, \delta_1), \ldots, (\delta_m, \delta_m)$. Если $H$ — существенно рекурсивная главная функция, то $y$ по определению является опровергающей вершиной. В противном случае некоторая вершина $z = (\lambda, \lambda, u'', w'')$ достижима из $y$ по некоторому ориентированному пути, дуги которого помечены парами логических условий $(\delta_{m+1}, \delta_{m+1}), \ldots, (\delta_n, \delta_n)$, $m<n$. Заметим, что согласно леммам 2, 3 для некоторых термов $t^1$, $t^2$, $T^1$ и $T^2$ справедливы равенства $u'' = u \circ \varphi(\langle [T^1], [T^2]\rangle)$, $w'' = \varphi(\langle [t^1], [t^2]\rangle) \circ w$. По условию лем-

мы для всякого $v$ из $U$ (в том числе для $v = \varphi(\langle [T^1], [T^2] \rangle) \circ \varphi(\langle [t^1], [t^2] \rangle))$ выполнено соотношение $u \circ v \circ w \neq e$, поэтому $u'' \circ w'' \neq e$. Следовательно, $z$ — опровергающая вершина, достижимая из корня $\Gamma(\pi_1, \pi_2)$.

Лемма 6. *Пусть по крайней мере один из двух главных функциональных символов $F^1$ и $F^2$ обладает свойством завершаемости, и пусть вершины $x_1 = (F^1, F^2, u_1, w_1)$ и $x_2 = (F^1, F^2, u_2, w_2)$ достижимы из корня графа $\Gamma(\pi_1, \pi_2)$. Предположим также, что справедливо только одно из двух равенств $u_1 = u_2$, $w_1 = w_2$. Тогда некоторая опровергающая вершина достижима из корня графа $\Gamma(\pi_1, \pi_2)$.*

Доказательство. Допустим, что $u_1 \neq u_2$, $w_1 = w_2$ и главный функциональный символ $F^2$ обладает свойством завершаемости в программе $\pi_2$. Тогда, следуя доказательству леммы 5, мы можем прийти к выводу о том, что из $x_1$ по ориентированному пути, дуги которого помечены парами логических условий

$$(\delta_1, \delta_1), \ldots, (\delta_m, \delta_m), \tag{3}$$

достижимы либо одна из вершин вида $y_1 = (H, \lambda, u_1', w_1')$, где $H$ — существенно рекурсивный главный функциональный символ, либо одна из вершин вида $z_1 = (\lambda, \lambda, u_1'', w_1'')$.

В первом случае $y_1$ и есть искомая опровергающая вершина. Во втором случае возможны два варианта. Если $u_1'' \circ w_1'' \neq e$, то $z_1$, очевидно, также является опровергающей вершиной, достижимой из корня графа $\Gamma(\pi_1, \pi_2)$. Если же $u_1'' \circ w_1'' = e$, то по лемме 2 мы можем рассмотреть вершину $z_2 = (\lambda, \lambda, u_2'', w_2'')$, достижимую из вершины $x_2$ по ориентированному пути, дуги которого помечены теми же парами (3) . На основании леммы 2 для вершин $z_1$ и $z_2$ получаем соотношения

$$u_1'' = u_1 \circ \varphi(\langle [T^1], [T^2] \rangle), \qquad w_1'' = \varphi(\langle [t^1], [t^2] \rangle) \circ w_1,$$
$$u_2'' = u_2 \circ \varphi(\langle [T^1], [T^2] \rangle), \qquad w_2'' = \varphi(\langle [t^1], [t^2] \rangle) \circ w_2,$$

справедливые при некотором выборе базовых термов $t^1$, $t^2$, $T^1$ и $T^2$. Обозначим $v_1$ и $v_2$ элементы критериальных моноидов $u_1'' \circ \varphi(\langle [t^1], [t^2] \rangle)$ и $u_2'' \circ \varphi(\langle [t^1], [t^2] \rangle)$ соответственно. Заметим, что равенство $w_1 = w_2$ влечет $v_1 = v_2$. Ввиду того, что $u_1 \neq u_2$ и $v_1 \circ w_1 = w_1'' \circ u_1'' = e$, мы получаем, приняв во внимание требование Тр2 критериальной системы $K$, соотношение $u_2'' \circ w_2'' = v_2 \circ w_2 = v_1 \circ w_2 \neq e$. Оно свидетельствует о том, что $z_2$ — опровергающая вершина. В случае $u_1 = u_2$, $w_1 \neq w_2$ доказательство аналогично.

Лемма 7. *Пусть $L = \max(|\pi_1|, |\pi_2|) + 1$ и пусть $F^1$ и $F^2$ — главные функциональные символы, один из которых существенно рекурсивный, а другой обладает свойством завершаемости. Предположим, что по крайней мере $L$ попарно различных вершин $x_1 = (F^1, F^2, u_1, w_1)$, $\ldots$, $x_L = (F^1, F^2, u_L, w_L)$ достижимы из корня графа $\Gamma(\pi_1, \pi_2)$. Тогда некоторая опровергающая вершина также достижима из корня графа $\Gamma(\pi_1, \pi_2)$.*

Доказательство. Воспользовавшись леммой 6, мы можем ограничиться только рассмотрением случая попарно различных элементов $w_1, \ldots, w_L$. Для определенности будем считать, что символ $F^2$ обладает свойством завершаемости, а $F^1$ — существенно рекурсивный. Завершаемость символа $F^2$ означает, что имеются такие конечная последовательность условий $\delta_1^2, \ldots, \delta_m^2$ и последовательность главных функциональных символов $F_1^2, \ldots, F_m^2$, что $m < L$, $F_1^2 = F^2$, $T_{\pi_2}(F_i^2, \delta_i^2) = T_i^2 F_{i+1}^1 b_i^2$, $1 \leqslant i < m$, и $T_{\pi_2}(F_m^2, \delta_m^2) = T_m^2 \in \mathrm{BTerm}$. С другой стороны, существенная рекурсивность символа $F^1$ позволяет считать, что существуют такие конечная последовательность условий $\delta_1^1, \ldots, \delta_n^1$ и последовательность главных функциональных символов $F_1^1, \ldots, F_n^1$, что $n < L$, $F_1^1 = F^1$, $T_{\pi_1}(F_j^1, \delta_j^1) = T_j^1 F_{j+1}^1 b_j^1$, $1 \leqslant j \leqslant n$, и при этом $F_n^1$ — существенно рекурсивный главный функциональный символ в программе $\pi_1$.

Рассмотрим матрицу $\{v_{ij}\}_{j=1,L}^{i=1,m}$, элементы которой задаются равенством $v_{ij} = w^+ \circ \varphi(\langle [b_i^1 \ldots b_1^1], [b_i^2 \ldots b_1^2] \rangle) \circ w_j$.

Согласно требованию Тр2 критериальной системы список элементов $v_{i1}, v_{i2}, \ldots, v_{iL}$, образующий $i$-ю строку матрицы, содержит для каждого $i$, $1 \leqslant i \leqslant m$, не более одного элемента, равного нейтральному элементу $e$ моноида $W$. В силу неравенства $m < L$ найдется такое $k$, $1 \leqslant k \leqslant L$, что все элементы $k$-го столбца матрицы $v_{1k}, v_{2k}, \ldots, v_{mk}$ отличны $e$. Тогда по определению графа $\Gamma(\pi_1, \pi_2)$ найдется ориентированный путь, дуги которого помечены парами $(\delta_1^1, \delta_1^2), \ldots, (\delta_m^1, \delta_m^2)$, из вершины $x_k$ в вершину вида $y = (F_n^1, w', u')$. Очевидно, $y$ — опровергающая вершина.

Теорема 3. *Пусть $\mathcal{F}$ — полугрупповая уравновешенная шкала над алфавитом базовых функциональных символов $\mathcal{B}$, и $K = \langle W, U, w^+, w^* \rangle$ — такая критериальная система для $\mathcal{F}$, что в $W$ разрешима проблема тождеств слов «$w_1 = w_2$?». Тогда на шкале $\mathcal{F}$ разрешима проблема эквивалентности ЛУРП «$\pi_1 \sim_{\mathcal{F}} \pi_2$?».*

Доказательство. Основываясь на теореме 1, мы можем считать, что $\pi_1$ и $\pi_2$ — нормализованные ЛУРП. Положим $L = \max(|\pi_1|, |\pi_2|) + 1$. Для моноида $W$ разрешима проблема тождеств «$w_1 = w_2$?», поэтому любой конечный фрагмент графа $\Gamma(\pi_1, \pi_2)$ может быть построен эффективно. По лемме 4 для проверки эквивалентности программ $\pi_1$ и $\pi_2$ достаточно убедиться в том, что ни одна из опровергающих вершин не достижима из корня графа $\Gamma(\pi_1, \pi_2)$.

В процессе поиска опровергающих вершин можно принять во внимание следующие соображения. Пусть $F^1$ и $F^2$ — главные функциональные символы, фигурирующие в рассматриваемых ЛУРП.

1. Если обе процедуры, $F_1$ и $F^2$, — незавершаемые, то, очевидно, никакая опровергающая вершина не достижима из вершины $x$ вида $(F^1, F^2, w, u)$.

2. Если одна из процедур $F^1$ и $F^2$ обладает свойством завершаемости, тогда как другая — существенно рекурсивная, то в силу лемм 6 и 7 из существования $L$ различных вершин $x$ вида $(F^1, F^2, w, u)$, достижимых из корня графа $\Gamma(\pi_1, \pi_2)$, следует, что хотя бы одна опровергающая вершина также достижима из корня этого же графа.

3. В графе $\Gamma(\pi_1, \pi_2)$ из всякой граничной вершины достижимы не более $|\mathcal{C}|^L$ вершин.

Итак, для проверки эквивалентности программ $\pi_1$ и $\pi_2$ достаточно исследовать фрагмент $\Gamma(\pi_1, \pi_2)$, содержащий не более $L^3(1+|\mathcal{C}|^{L+1})$ вершин.

Покажем теперь на нескольких примерах, как можно применять теорему 2 для обоснования разрешимости проблемы эквивалентности ЛУРП на полугрупповых шкалах.

Пример 1. Рассмотрим универсальную шкалу $\mathcal{U}$ над алфавитом базовых функций $\mathcal{B} = \{a^1, \ldots, a^N\}$. В качестве критериального возьмем моноид $W$, порожденный множеством элементов $(\mathcal{U} \times \mathcal{U}) \cup \{w^+, w^*\}$ и определяющими соотношениями

$$\langle [T_1], [T_2] \rangle \circ \langle [t_1], [t_2] \rangle = \langle [T_1 t_1], [T_2 t_2] \rangle,$$
$$w^+ \circ w^* = e, \qquad w^+ \circ \langle [t], [t] \rangle = w^+, \qquad \langle [t], [t] \rangle \circ w^* = w^*.$$

Нетрудно убедиться в том, что система $K = \langle W, \mathcal{U} \times \mathcal{U}, w^+, w^* \rangle$ является критериальной для шкалы $\mathcal{U}$, если в качестве гомоморфизма $\varphi$ взять тождественное отображение множества $\mathcal{U} \times \mathcal{U}$ на себя. В результате получаем

Следствие 1 [16, 22]. *Проблема эквивалентности ЛУРП на универсальных шкалах разрешима.*

Пример 2. В теории программирования определенный интерес вызывает класс программных семантик, в рамках которых допускается, что результат последовательного выполнения некоторых пар базовых операторов (действий) $a$ и $b$ не зависит от порядка их применения. В таком случае говорят о независимости операторов $a$, $b$. В общем случае отношением независимости является произвольное отношение толерантности $I$ на множестве базовых операторов $\mathcal{B} = \{a^1, \ldots, a^N\}$. Отношение независимости является основополагающим понятием теории трасс [27] и находит широкое применение в теории сетей Петри [29] и в казуальных логиках [3, 15, 31], используемых для анализа поведения параллельных и распределенных программных систем. Отношение независимости $I$ базовых операторов индуцирует шкалу $\mathcal{F}_c^I$, ассоциированную с частично коммутативным моноидом, порожденным базовыми операторами $\mathcal{B} = \{a^1, \ldots, a^N\}$ и определяющими соотношениями $[a^i a^j] = [a^j a^i]$, $(a^i, a^j) \in I$. Для построения алгоритма, разрешающего эквивалентность ЛУРП на шкале $\mathcal{F}_c^I$, воспользуемся моноидом $W$, элементы которого порождаются всевозможными парами из $\mathcal{F}_c^I \times \mathcal{F}_c^I$, а также двумя выделенными элементами $w^+$, $w^*$. Бинарная операция $\circ$ на $W$ определяется соотношениями

$$\langle [T_1], [T_2] \rangle \circ \langle [t_1], [t_2] \rangle = \langle [T_1 t_1], [T_2 t_2] \rangle,$$
$$w^+ \circ w^* = e, \qquad w^+ \circ \langle [t], [t] \rangle = w^+, \qquad \langle [t], [t] \rangle \circ w^* = w^*.$$

В [4] было установлено, что система $K = \langle W, \mathcal{F}_c^I \times \mathcal{F}_c^I, w^+, w^* \rangle$ является критериальной для $\mathcal{F}_c^I$, и на основании теоремы 2 мы получаем

Следствие 2. *Проблема эквивалентности ЛУРП на частично коммутативных полугрупповых шкалах $\mathcal{F}_c^I$ разрешима.*

## § 4. Полиномиальная разрешимость проблемы эквивалентности ЛУРП

Теорема 3 обеспечивает разрешимость проблемы эквивалентности ЛУРП, но лишь за экспоненциальное время. Для разработки полиномиальных разрешающих процедур нужно еще более сократить пространство поиска опровергающих вершин в графе $\Gamma(\pi_1, \pi_2)$. В некоторых случаях это оказывается возможным за счет использования тех или иных особенностей рассматриваемых шкал.

Теорема 4 [2]. *Проблема эквивалентности ЛУРП на универсальных шкалах разрешима за время $O(n^3 \log n)$.*

Доказательство. Разрешимость проблемы эквивалентности вытекает из теоремы 3 (см. следствие 1). Чтобы установить указанную верхнюю оценку сложности разрешающей процедуры, обратимся к соответствующей критериальной системе $K = \langle W, \mathcal{U} \times \mathcal{U}, w^+, w^* \rangle$ и заметим, что для любых двух термов $t_1$ и $t_2$ одинаковой длины уравнение $w^+ \circ \varphi(\langle [t_1], [t_2] \rangle) \circ X \circ w^*$ имеет решение $X$ в $\mathcal{U} \times \mathcal{U}$ тогда и только тогда, когда $w^+ \circ \varphi(\langle [t_1], [t_2] \rangle) = w^+$. Отсюда на основании лемм 5 и 6 вытекает, что для всякой пары процедур, $F^1$ и $F^2$, одна из которых обладает свойством завершаемости, достижимость из корня графа $\Gamma(\pi_1, \pi_2)$ двух различных вершин вида $(F^1, F^2, w, u)$ означает, что из корня этого графа достижима некоторая опровергающая вершина. Поэтому пространство поиска опровергающих вершин в графе $\Gamma(\pi_1, \pi_2)$ ограничивается $n^2$ вершинами.

Полиномиальная разрешимость проблемы эквивалентности ЛУРП может быть также установлена в тех случаях, когда критериальный моноид $W$ является группой. Для достижения этой цели нам придется продолжить список вспомогательных лемм. Как и ранее, мы предполагаем, что полугрупповая шкала $\mathcal{F}$ — уравновешенная, $K = \langle W, U, w^+, w^* \rangle$ — критериальная система для $\mathcal{F}$, программы $\pi_1$ и $\pi_2$ нормализованы. Кроме того, мы будем теперь считать, что моноид $W$ является группой.

Лемма 8. *Пусть $x = (F^1, F^2, u_1, w_1)$ и $y = (F^1, F^2, u_2, w_2)$ — граничные вершины, достижимые из корня графа $\Gamma(\pi_1, \pi_2)$. Если ни одна опровергающая вершина не достижима ни из $x$, ни из $y$, то справедливо равенство $w_1 \circ u_1 = w_2 \circ u_2$.*

Доказательство. По определению граничных вершин и на основании леммы 2 найдется такая последовательность условий $\delta_1, \ldots, \delta_m$, что цепочки

$$x = x_0 \xrightarrow{(\delta_1, \delta_1)} x_1 \xrightarrow{(\delta_2, \delta_2)} \ldots \xrightarrow{(\delta_m, \delta_m)} x_m = z', \qquad y = y_0 \xrightarrow{(\delta_1, \delta_1)} y_1 \xrightarrow{(\delta_2, \delta_2)} \ldots \xrightarrow{(\delta_m, \delta_m)} y_m = z''$$

образуют ориентированные пути в $\Gamma(\pi_1, \pi_2)$, ведущие в вершины $z' = (\lambda, \lambda, u', w')$, $z'' = (\lambda, \lambda, u'', w'')$, и при этом соотношения

$$u' = u_1 \circ \varphi(\langle [T^1], [T^2] \rangle), \qquad w' = \varphi(\langle [t^1], [t^2] \rangle) \circ w_1,$$
$$u'' = u_2 \circ \varphi(\langle [T^1], [T^2] \rangle), \qquad w'' = \varphi(\langle [t^1], [t^2] \rangle) \circ w_2$$

выполняются для подходящих базовых термов $t^1$, $t^2$, $T^1$, $T^2$. Поскольку вершины $z'$ и $z''$ не относятся к числу опровергающих, имеем $u' \circ w' = u'' \circ w'' = e$. Следовательно, будут справедливы и равенства

$$\varphi(\langle [T^1], [T^2] \rangle) \circ \varphi(\langle [t^1], [t^2] \rangle) = (w_1 \circ u_1)^{-1} = (w_2 \circ u_2)^{-1}.$$

Лемма 9. *Пусть $L = \max(|\pi_1|, |\pi_2|)$, и пусть $F_0^1$ и $F_0^2$ — граничные процедуры в программах $\pi_1$ и $\pi_2$ соответственно. Предположим, что $y_1 = (F_0^1, F_0^2, u_1, w_1)$, $\ldots$, $y_L = (F_0^1, F_0^2, u_L, w_L)$, $y_{L+1} = (F_0^1, F_0^2, u_{L+1}, w_{L+1})$ — набор попарно различных граничных вершин, достижимых из корня графа $\Gamma(\pi_1, \pi_2)$. Предположим также, что никакая опровергающая вершина не достижима из $y_1, \ldots, y_L$. Тогда опровергающая вершина достижима из $y_{L+1}$ в том и только том случае, когда $w_{L+1} \circ u_{L+1} \neq w_1 \circ u_1$.*

Доказательство. Достаточность условия достижимости опровергающей вершины следует из леммы 8.

Обоснуем необходимое условие. Допустим, что $w_{L+1} \circ u_{L+1} = w_1 \circ u_1$. Рассмотрим произвольный ориентированный путь

$$y_{L+1} = x_0 \xrightarrow{(\delta_0^1, \delta_0^2)} x_1 \xrightarrow{(\delta_1^1, \delta_1^2)} \ldots \xrightarrow{(\delta_m^1, \delta_m^2)} x_{m+1} = z', \tag{4}$$

ведущий в вершину $z' = (\lambda, \lambda, u', w')$, полагая, что для каждого $i$, $1 \leqslant i \leqslant m$, вершина $x_i = (F_i^1, F_i^2, v_i^1, v_i^2)$ такова, что $T_{\pi_k}(F_{i-1}^k, \delta_i^k) = T_i^k F_i^k b_i^k$, $1 \leqslant i \leqslant m$, и $T_{\pi_k}(F_m^k, c_m^k) = T_{m+1}^k$, $k = 1, 2$. Ясно, что $m + 1 < L$. Далее (подобно тому, как было сделано при доказательстве леммы 7) рассмотрим матрицу $\{v_{ij}\}_{j=1,L}^{i=0,m}$, элементы которой задаются соотношениями $v_{ij} = w^+ \circ \varphi(\langle [b_i^1 \ldots b_1^1], [b_i^2 \ldots b_1^2] \rangle) \circ w_j$. В соответствии с требованием Тр2 критериальной системы в каждой строке матрицы содержится не более одного элемента $v_{ij}$, равного нейтральному элементу $e$ моноида $W$. В силу неравенства $m + 1 < L$ найдется такой номер $k$, $1 \leqslant k \leqslant L$, что $k$-й столбец $v_{0k}, v_{1k}, \ldots, v_{mk}$ данной матрицы не содержит нейтральных элементов. Тогда по определению графа $\Gamma(\pi_1, \pi_2)$ существует ориентированный путь из вершины $y_k$ в вершину $z'' = (\lambda, \lambda, w'', u'')$. Положим $w_0 = \varphi(\langle [b^1_m \ldots b^1_1], [b^2_m \ldots b^2_1] \rangle)$ и $u_0 = \varphi(\langle [\widehat{T}^1_1 \ldots \widehat{T}^1_{m+1}], \widehat{T}^2_1 \ldots \widehat{T}^2_{m+1}] \rangle)$. Вершина $z''$ не является опровергающей, а потому будут выполнены равенства $e = u'' \circ w'' = u_k \circ u_0 \circ w_0 \circ w_k$. Отсюда следует справедливость равенства $u_0 \circ w_0 = (w_k \circ u_k)^{-1}$.

Применяя лемму 8, получаем соотношения

$$u_0 \circ w_0 = (w_k \circ u_k)^{-1} = (w_1 \circ u_1)^{-1} = (w_{L+1} \circ u_{L+1})^{-1}.$$

В итоге имеем

$$u' \circ w' = u_{L+1} \circ u_0 \circ w_0 \circ w_{L+1} = u_{L+1} \circ (w_{L+1} \circ u_{L+1})^{-1} w_{L+1} = e.$$

Отсюда следует, что вершина $z'$ не является опровергающей. Мы рассматривали произвольный ориентированный путь (4) из $y_{L+1}$, а потому это означает, что никакая опровергающая вершина не достижима из $y_{L+1}$.

Теорема 5. *Пусть $\mathcal{F}$ — полугрупповая упорядоченная шкала над алфавитом базовых функциональных символов $\mathcal{B}$, и пусть $K = \langle W, U, w^+, w^* \rangle$ — такая критериальная система для $\mathcal{F}$, что в группе $W$ проблема тождеств слов «$w_1 = w_2$?» разрешима за время $\tau(m)$, где $m = \max(|w_1|, |w_2|)$. Тогда проблема эквивалентности ЛУРП «$\pi_1 \sim_{\mathcal{F}} \pi_2$?» на шкале $\mathcal{F}$ разрешима за время $C_1 L^3 \tau(C_2 L^3)$, где $L = \max(|\pi_1|, |\pi_2|)$.*

Доказательство. Теорема 3 гарантирует разрешимость проблемы эквивалентности путем построения критериального графа. Согласно лемме 9 для анализа эквивалентности программ $\pi_1$, $\pi_2$ достаточно рассмотреть не более $|\mathcal{C}|L^3$ вершин.

Пример 3. Рассмотрим шкалу $\mathcal{F}_{fc}$, ассоциированную со свободным коммутативным моноидом базовых функций $\mathcal{B} = \{a^1, \ldots, a^N\}$. Для построения подходящей критериальной системы для $\mathcal{F}_{fc}$ возьмем свободную абелеву группу $Z$ ранга $N$, порожденную некоторыми элементами $q_1, \ldots, q_N$. Нетрудно убедиться, что система $K = \langle Z, Z, e, e \rangle$ будет критериальной для $\mathcal{F}_{fc}$, если в качестве гомоморфизма использовать отображение $\varphi$, для любых базовых функций $a_i$, $a_j$ удовлетворяющее условиям $\varphi(\langle [a_i], [\lambda] \rangle) = q_i$ и $\varphi(\langle [\lambda], [a_j] \rangle) = q_j^{-1}$. Тогда, применяя экономную технику кодирования вершин критериального графа, мы извлекаем из теоремы 4

Следствие 3. *Проблема эквивалентности ЛУРП на свободно коммутативных полугрупповых шкалах $\mathcal{F}^{fc}$ разрешима за время $O(n^3 \log n)$* .

Таким образом, с использованием новой техники доказательства разрешимости проблемы эквивалентности линейных унарных рекурсивных программ нам удалось обнаружить ряд новых разрешимых случаев указанной проблемы, причем некоторые из них оказались разрешимыми за полиномиальное время. Этот подход может быть распространен на более общие виды рекурсивных программ и интерпретаций.

## СПИСОК ЛИТЕРАТУРЫ

1. Ершов А. П. Об операторных схемах Янова // Проблемы кибернетики. Вып.20. — М.: Наука, 1967. — С. 181–200.
2. Захаров В. А. Полиномиальный алгоритм разрешения проблемы эквивалентности унарных линейных рекурсивных схем программ // Труды II Международной конференции «Дискретные модели в теории управляющих систем». — М.: Диалог-МГУ, 1997. — С. 26–29.

3. Захаров В. А., Спанопуло В. В. О взаимосвязи двух семантик параллельных вычислений // Программирование. — 1997. — № 5. — С. 36–48.
4. Захаров В. А. Быстрые алгоритмы разрешения эквивалентности операторных программ на уравновешенных шкалах // Математические вопросы кибернетики. Вып.7. — М.: Наука. — 1998. — С. 303–324.
5. Захаров В. А. Быстрые алгоритмы разрешения эквивалентности пропозициональных операторных программ на упорядоченных полугрупповых шкалах // Вестник Московского университета, сер. 15, Вычислительная математика и кибернетика — 1999. — № 3. — С. 29–35.
6. Котов В. Е., Сабельфельд В. К. Теория схем программ. — М.: Наука, 1991.
7. Лисовик Л. П. О разрешимых проблемах для металинейных схем // Докл. АН УССР. — 1979. — № 2. — С. 130–133.
8. Лисовик Л. П. Металинейные рекурсивные схемы над размеченными деревьями // Программирование. — 1983. — № 5. — С. 13–22.
9. Лисовик Л. П. Металинейные схемы с засылками констант // Программирование. — 1985. — № 2. — С. 29–38.
10. Лисовик Л. П. Стандартные схемы с магазинами // Докл. АН УССР. — 1989. — № 12. — С. 23–27.
11. Лисовик Л. П. Схемы программ и преобразователи над размеченными деревьями // Математические вопросы кибернетики. Вып. 6. — М.: Наука, 1996. — С. 281–321.
12. Романовский В. Ю. Разрешимость эквивалентности линейных унарных рекурсивных схем с индивидуальными константами // Доклады АН УССР. — 1980. — № 1. — С. 72-75.
13. Сабельфельд В. К. О преобразованиях унарных линейных рекурсивных схем // Кибернетика. — 1975 — № 5. — С. 56–63.
14. Янов Ю. И. О логических схемах алгоритмов // Проблемы кибернетики. Вып. 1. — М.: Физматгиз, 1958. — С. 75–127.
15. Alur R, Peled D., Penczek W. Model-checking of causality properties // In Proceedings of the 11th IEEE Conference on Logic in Computer Science. — 1995. — P. 90–100.
16. Ashcroft E., Manna Z., Pnueli A. Decidable properties of monadic functional schemes // Journal of the ACM. — 1973. — V. 20, № 3. — P. 489–499.
17. De Bakker J. W., Scott D. A. Theory of programs. Unpublished notes // Vienna: IBM Seminar. — 1969.
18. Courcelle B. Recursive applicative program schemes // Handbook of theoretical computer science. Vol. B: Formal models and semantics. — Elsevier Sci. Publ., 1994. — P. 459–492.
19. Culik K. II. New techniques for proving the decidability of equivalence problems // Lecture Notes in Computer Science. — 1988. — V. 317. — P. 162–175.
20. Friedman E. Equivalence problems for deterministic languages and monadic recursion schemes // Journal of Computer and System Science. — 1977. — V. 14. — P. 362–399.
21. Friedman E., Greibach S. A. Monadic recursion schemes: the effect of constant // Journal of Computer and System Science. — 1979. — V. 18. — № 3. — P. 354–266.
22. Garland S. J., Luckham D. C. Program schemes, recursion schemes and formal languages // Journal of Computer and System Science. — 1973. — V. 7, № 2. — P. 119–160.
23. Harel D. Dynamic logics // Handbook of philosophical logics. — Elsevier Sci. Publ., 1984. — P. 497–604.
24. Ibarra O. H. Reversal-bounded multicounter machines and their decision problems // Journal of the ACM. — 1978. — V. 25, № 1. — P. 116–133.
25. Lisovik L. P. Hard sets method and semilinear reservoir method with applications // Lecture Notes in Computer Science. — 1996. — V. 1099. — P. 219–231.
26. Luckham D. C., Park D. M., Paterson M. S. On formalized computer programs // Journal of Computer and System Science. — 1970. — V. 4, № 3. — P. 220–249.
27. Mazurkiewicz A. Trace theory // Lecture Notes in Computer Science. — 1987. — V. 255. — P. 279–324.
28. McCarthy J. A basis for a mathematical theory of computation // Computer Programming and Formal Systems. Amsterdam, North-Holland Publ. Co. — 1963. — P. 33–70.
29. Reisig W. On the semantics of Petri nets. — Hamburg, 1984.
30. Sénizergues G. The equivalence problem for deterministic pushdown automata is decidable // Lecture Notes in Computer Science. — 1997. — V. 1256. — P. 671–681.
31. Thiagarajan P. S. A trace based extension of linear time temporal logic // In Proceedings of the 10th IEEE Conference on Logic in Computer Science. — 1994.
32. Tomita E., Seino K. The extended equivalence problem for a class of non-real-time deterministic pushdown automata // Acta Informatica. — 1995. — V. 32, № 4. — P. 395–413.
33. Valiant L. G., Paterson M. S. Deterministic one-counter automata // Journal of Computer and System Science. — 1975. — V. 10. — P. 340–350.
34. Zakharov V. A. An efficient and unified approach to the decidability of tquivalence of propositional program schemes // Lecture Notes in Computer Science. — Springer-Verlag. — 1998. — V. 1443. — P. 247–259.

Поступило в редакцию 24 IX 1998