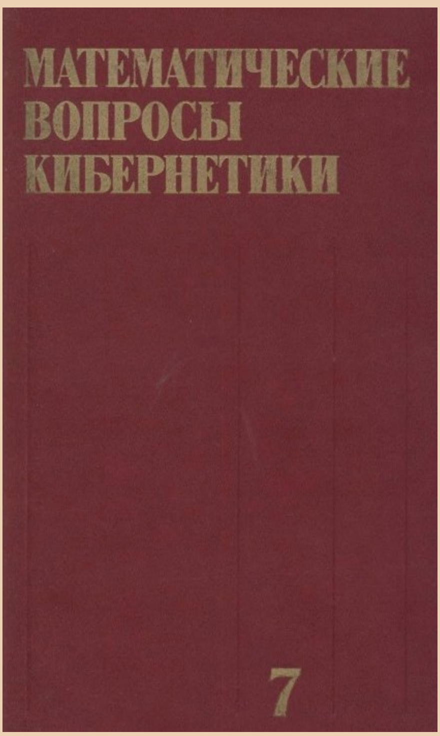

В. А. Захаров

Быстрые алгоритмы разрешения эквивалентности операторных программ на уравновешенных шкалах

БЫСТРЫЕ АЛГОРИТМЫ РАЗРЕШЕНИЯ ЭКВИВАЛЕНТНОСТИ ОПЕРАТОРНЫХ ПРОГРАММ НА УРАВНОВЕШЕННЫХ ШКАЛАХ *)

В. А. ЗАХАРОВ

(МОСКВА)

§ 1. Введение

Проблема эквивалентности программ изучается в теории вычислений уже более сорока лет. Пристальное внимание к этой задаче обусловлено ее значительным влиянием на теорию и практику программирования. Проблема эквивалентности программ явно или опосредованно возникает при решении многих задач системного программирования, к числу которых относятся задачи трансляции [1], оптимизации [18] и верификации [39, 40] вычислительных программ, разработка методов частичных вычислений [17], задачи специализации и повторного использования программ [2] и др. Всякий раз, когда возникает необходимость изменить синтаксическую составляющую вычислительной программы (текст программы), сохраняя при этом в той или иной мере ее семантическую компоненту (поведение, или функционирование), приходится иметь дело с отношением эквивалентности на множестве программ и, следовательно, с проблемой эквивалентности программ. Необходимость исследования свойства программных вычислений, наряду с проблемами синтаксического анализа и перевода, послужила источником многочисленных моделей вычислений, получивших широкое распространение и применение в современном программировании.

Самая общая формулировка проблемы эквивалентности такова. Для произвольной пары программ в заданной модели вычислений требуется выяснить, обладают ли они одинаковым поведением. Уточняя понятия вычислительной модели, программы и ее поведения, получаем многочисленные вариации проблемы эквивалентности. Впервые эта проблема была подробно исследована в [86], где в качестве вычислительной модели рассматривалась произвольная универсальная система программирования, отвечающая главной нумерации частично рекурсивных функций, а поведение программы определялось вычисляемой ею частично рекурсивной функцией. При этих условиях было установлено, что всякое свойство программ, инвариантное по отношению к их поведению (так называемое функциональное свойство), является нерекурсивным. Отсюда следует алгоритмическая неразрешимость очень многих задач семантического анализа программ, включая проблему эквивалентности в универсальных системах программирования, если под эквивалентностью программ понимать совпадение вычисляемых ими функций.

Несколькими годами позднее отрицательный результат теоремы Райса [86] был уравновешен серией положительных результатов, полученных

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

сначала в работах А. А. Ляпунова и Ю. И. Янова, а затем в статье М. Рабина и Д. Скотта [85]. В серии работ [34, 35, 60, 61] была разработана и исследована новая модель вычислений, получившая впоследствии название схем Ляпунова — Янова, в которой формализация операторных программ осуществляется на пропозициональном уровне абстракции, а поведение формализованной таким образом программы полностью определяется совокупностью допустимых трасс вычислений. Было показано, что эквивалентность программ в такой модели разрешима, а соответствующая эквациональная теория программ конечно аксиоматизируема. В [85] были разработаны основы теории конечных автоматов и был поставлен ряд алгоритмических задач анализа поведения конечных автоматов-акцепторов различных типов (детерминированных, недетерминированных, многоленточных, двухсторонних и др.). В частности, была установлена разрешимость проблемы эквивалентности одноленточных конечных автоматов, неразрешимость проблемы включения для многоленточных автоматов, а также была поставлена проблема эквивалентности детерминированных многоленточных автоматов, остававшаяся впоследствии открытой на протяжении тридцати лет. В [87] была выявлена тесная взаимосвязь между моделями вычислений схем Янова и конечных автоматов. В [6] теория схем Янова была представлена в более простом изложении и при этом были явно обозначены возможности ее применения при решении прикладных задач программирования. Таким образом, в середине 60-х годов сложилось новое направление в теории программирования — теория схем программ, основным объектом исследования которой становятся формальные модели вычислений.

За прошедшие три десятилетия в рамках теории вычислений были построены и изучены многие новые модели программ. Обзор основных концепций и результатов в этой области представлен в [1, 4, 7, 8, 20, 66]. Наиболее интенсивно исследования проводились по следующим направлениям.

1. Изучение проблем эквивалентности, эквивалентных преобразований, включения, пустоты и др. для схем программ со структурированной памятью [16, 20, 38, 41, 42, 56, 58, 79, 80, 82, 83, 90].
2. Разработка моделей рекурсивных вычислений и исследование эквивалентности рекурсивных программ [15, 29–33, 50–54, 57, 63, 64, 71, 90].
3. Изучение проблемы эквивалентности дискретных преобразователей (автоматов над полугруппами) [3–5, 21–28, 37, 59].
4. Развитие теории модели схем программ Янова [9–13, 43–52]
5. Исследование проблемы эквивалентности детерминированных многоленточных автоматов [19, 65, 68, 76].
6. Исследование проблемы эквивалентности детерминированных автоматов с магазинной памятью [36, 55, 67, 88, 91, 93–95, 97–99].
7. Исследование проблемы эквивалентности простых операторных программ [62, 72, 73, 78, 81, 96].
8. Изучение вопросов трансляции программ из одних вычислительных моделей в другие [20, 71, 84, 92].

Помимо этого, элементы теории схем программ были положены в основу нового класса неклассических логик — динамических логик [74, 75]. Проводились также попытки распространить элементы этой теории на параллельные и распределенные программы.

В построенных моделях вычислений весьма интенсивно изучалась проблема эквивалентности. Было найдено немало примеров моделей программ, обладающих нерекурсивным отношением эквивалентности, см. [5, 22, 23, 38, 41, 44, 73, 80–83, 85]. В то же время во многих случаях удалось доказать разрешимость проблемы эквивалентности. Из достижений последнего десятилетия в этом направлении можно выделить серию результатов Л. П. Лисовика [31–33], в которых разработан общий подход к построению алгоритмов, разрешающих эквивалентность рекурсивных преобразователей различных типов, работы В. К. Сабельфельда [58, 89], построившего алгоритм для разрешения эквивалентности в широком классе так называемых «сквозных» схем операторных программ со структурированной памятью, и работы [76, 91], содержащие окончательное положительное решение алгоритмических проблем эквивалентности для многоленточных детерминированных автоматов и детерминированных автоматов с магазинной памятью.

Вместе с тем при всем многообразии процедур, разрешающих эквивалентность программ в тех или иных моделях, сложностной аспект проблемы эквивалентности программ изучен весьма незначительно. Одна из причин состоит в том, что даже для простейшей вычислительной модели схем Янова проблема эквивалентности схем оказывается *NP*-полной задачей: анализ логических условий схем требует сравнивать булевы формулы, т. е. в сущности решать задачу выполнимости пропозициональных формул. Здесь нужно, однако, иметь в виду, что сложность алгоритма, разрешающего эквивалентность программ, складывается из двух компонент. Одна имеет чисто комбинаторную природу и возникает в результате анализа формул, задающих логические условия в управляющем графе программы (системе переходов). Если сигнатура программ не ограничена и допустимо произвольное число разнообразных элементарных логических условий, то задача сравнительного анализа структуры управления программ будет *NP*-полной [69, 70] в подавляющем большинстве вычислительных моделей. Следует отметить, что эта задача имеет две особенности. С одной стороны, она присуща почти всем известным моделям программ и практически не зависит от индивидуальных особенностей семантик этих моделей. С другой стороны, если многообразие элементарных логических условий фиксировано и конечно, а размер исследуемых программ не ограничен, то эту задачу можно решить за полиномиальное время. В таком случае говорят об униформной проблеме эквивалентности программ; вычислительная сложность этой проблемы целиком определяется второй компонентой сложности, связанной с анализом вычислений программ в конкретной семантике. Эта компонента зависит только от индивидуальных особенностей исследуемой вычислительной модели; с этой точки зрения ее можно рассматривать как «собственно вычислительную» сложностную характеристику модели (в отличие от «комбинаторной» сложности). Ограничения, налагаемые на синтаксис вычислительной модели при униформизации проблемы эквивалентности, не оказывают заметного воздействия на вычислительную мощность программ [38, 41, 62], но позволяют лучше оценить влияние тех или иных особенностей модели на вычисления, порождаемые программой. Например, как было отмечено, проблема эквивалентности схем Янова является в общем случае *NP*-полной задачей. Однако униформная проблема эквивалентности схем Янова сводится за линейное время к проблеме эквивалентности конечных автоматов и решается за время $Cn \log n$ [77], где константа C экспоненциально зависит от количества базовых логических условий, используемых в схемах.

В настоящей работе изучается униформная проблема эквивалентности пропозициональных операторных программ, обобщающих схемы Ляпунова — Янова [34, 61] и дискретные преобразователи Глушкова — Летичевского [4]. Униформность проблемы эквивалентности означает, что мы рассматриваем операторные программы конечной фиксированной сигнатуры, не налагая при этом ограничений на размеры самих программ. В § 2 приводится подробное описание синтаксиса и семантики пропозициональных операторных программ. Синтаксически рассматриваемые программы совпадают со схемами Янова; семантика вычислительной модели этих программ определяется на основе семантики пропозициональной динамической логики [74]. В § 3 содержится краткий обзор основных результатов, устанавливающих разрешимость проблемы эквивалентности конечных автоматов относительно некоторых типов полугрупп. Эти результаты интерпретируются в терминах модели пропозициональных операторных программ. В § 4 исследуется проблема эквивалентности программ в специальном классе уравновешенных моделей. Отличительная особенность этих моделей состоит в том, что если две конечные последовательности операторов, $y'_1, \ldots, y'_n$ и $y''_1, \ldots, y''_m$, имеют различную длину, т. е. $n \neq m$, то результаты их вычисления при одних и тех же начальных данных будут различны. Для уравновешенных моделей программ предложен общий подход к построению процедур, разрешающих эквивалентность программ. Методика построения разрешающих алгоритмов основана на сведении проблемы эквивалентности к проблеме тождества в специальных полугруппах. В каждом случае выбор полугруппы зависит от индивидуальных особенностей исследуемой уравновешенной модели программ. В § 5 представлены примеры использования разработанной методики для построения полиномиальных по времени алгоритмов, разрешающих проблему эквивалентности в некоторых подклассах уравновешенных моделей: для программ с перестановочными, частично перестановочными и условно эквивалентными операторами. Намечены перспективы дальнейших исследований в этом направлении с целью построения «быстрых» алгоритмов, разрешающих эквивалентность в других классах моделей пропозициональных операторных программ, в том числе рекурсивных.

§ 2. Пропозициональные операторные программы

В этом параграфе вводятся синтаксис и семантика пропозициональных операторных программ, обсуждаются простейшие свойства вычислений этих программ и формализуется проблема эквивалентности.

2.1. Синтаксис пропозициональных операторных программ. Фиксируем два конечных алфавита: $\mathcal{A} = \{a^1, \ldots, a^N\}$ и $\mathcal{P} = \{p_1, \ldots, p_M\}$. Символы алфавита $\mathcal{A}$ будем называть *операторами*, подразумевая, что ими обозначаются элементарные операторы программ, наподобие операторов присваивания вида $x := y + 2$ или операторов вызова процедуры вида **call** Factorial(n, x). Конечные последовательности операторов будем называть *операторными цепочками* (о. ц.). Множество всех операторных цепочек обозначим $\mathcal{A}^*$ и будем использовать символ λ для обозначения пустой о. ц. Длину о. ц. h обозначим $|h|$, а конкатенацию (последовательное соединение) операторных цепочек h и g обозначим hg. Естественно, предполагается, что $|\lambda| = 0$ и $\lambda h = h\lambda = h$.

Символы алфавита $\mathcal{P}$ будем называть *логическими переменными*, полагая при этом, что ими обозначаются элементарные свойства и отношения на множестве данных, с которым имеют дело программы. Каждая логическая переменная может принимать одно из двух истинностных значений: 0 или 1. Двоичный набор $\delta = \langle d_1, \ldots, d_M \rangle$ длины M, где $M = |\mathcal{P}|$, образованный истинностными значениями всех логических переменных, назовем *логическим условием*. Множество всех таких наборов обозначим $\mathcal{C}$.

Пропозициональная операторная программа сигнатуры $(\mathcal{A}, \mathcal{P})$ задается конечной системой переходов $\pi = \langle V$, **вход**, **выход**, **тупик**, B, $T \rangle$, где

V — конечное множество *вершин-преобразователей*;

вход — *начальная вершина*, **вход** $\notin V$;

выход — *заключительная вершина*, **выход** $\notin V$;

тупик — *вершина пустого цикла*, **тупик** $\notin V$;

B: $V \to \mathcal{A}$ — *функция привязки*, ставящая в соответствие каждому преобразователю программы некоторый оператор;

T: $(V \cup \{\textbf{вход}\}) \times \mathcal{C} \to (V \cup \{\textbf{выход}, \textbf{тупик}\})$ — *функция переходов*.

Пропозициональную операторную программу можно представить в виде конечного ориентированного графа с множеством вершин $V \cup \{\textbf{вход}, \textbf{выход}, \textbf{тупик}\}$ и множеством дуг $E = \{(v, T(v, \delta)) \colon v \in V \cup \{\textbf{вход}\},\ \delta \in \mathcal{C}\}$.

В графе особо выделены три вершины **вход**, **выход**, **тупик**; всем прочим вершинам графа приписаны операторы. Каждой дуге графа приписано некоторое условие, причем дугам, исходящим из одной вершины, сопоставлены попарно различные условия. *Размером* $|\pi|$ программы π назовем число $|V|$ ее вершин-преобразователей.

Будем говорить, что вершина v'' программы π *доступна* для вершины v', если $v'' = v'$ или $v'' = T(\ldots T(T(v', \delta_1), \delta_2), \ldots, \delta_n)$ для некоторой конечной последовательности условий $\delta_1, \delta_2, \ldots, \delta_n$, $n \geqslant 1$. С точки зрения теоретико-графового представления программы доступность вершины v'' для вершины v' означает, что в программе имеется маршрут из v' в v''. Конечный маршрут, исходящий из начальной вершины, будем называть *трассой*. Программу π назовем *редуцированной*, если всякий ее преобразователь доступен для начальной вершины, а заключительная вершина **выход** доступна для всякого преобразователя.

2.2. Динамические шкалы. Семантика пропозициональных операторных программ описывается посредством шкал и моделей пропозициональной динамической логики [74].

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

непустого множества *состояний* S,

начального состояния s_0, $s_0 \in S$,

функции преобразования R: $S \times \mathcal{A} \to S$.

Шкала $\mathcal{F}$ 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 оператора a из $\mathcal{A}$ на состоянии данных s, $s \in S$, является состояние $s' = R(s, a)$. Мы полагаем при этом, что каждый элементарный оператор a выполним на любом состоянии данных s и результат его выполнения определяется состоянием s однозначно. Эти ограничения традиционно соблюдаются во всех системах операторного программирования, поэтому мы будем иметь дело только с функциональными сериальными динамическими шкалами.

Функцию преобразования R можно распространить естественным образом на всё множество операторных цепочек $\mathcal{A}^*$, полагая $R^*(s, \lambda) = s$ и $R^*(s, ha) = R(R^*(s, h), a)$ для всякой о. ц. h из $\mathcal{A}^*$, оператора a из $\mathcal{A}$ и состояния s из S. Будем говорить, что состояние s'' *достижимо* на шкале $\mathcal{F}$ из состояния s' (обозначая этот факт $s' \preceq_{\mathcal{F}} s''$), если $s'' = R^*(s', h)$ для некоторой о. ц. h из $\mathcal{A}^*$. Для каждой о. ц. h обозначим $[h]_{\mathcal{F}}$ состояние $s = R^*(s_0, h)$, достижимое на $\mathcal{F}$ из начального состояния посредством h. При изучении вычислений пропозициональных операторных программ нас будут интересовать только те состояния данных, которые достижимы из начального состояния s_0. Поэтому, не ограничивая общности, будем полагать $S = \{[h]_{\mathcal{F}}\colon h \in \mathcal{A}^*\}$ для всех рассматриваемых в дальнейшем шкал. Кроме того, для упрощения записи мы будем опускать обозначение шкалы $\mathcal{F}$ в выражениях $[h]_{\mathcal{F}}$ и $s' \preceq_{\mathcal{F}} s''$, если шкала однозначно понимается из контекста.

Подобно программе, динамическую шкалу можно рассматривать 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 настоящей работы более выгодным представляется алгебраическое истолкование некоторых типов динамических шкал.

Шкалу $\mathcal{F}_s = \langle S', s, R'\rangle$ назовем *подшкалой* шкалы $\mathcal{F} = \langle S, s_0, R\rangle$, порожденной состоянием s, если $S' = \{R^*(s, h)\colon h \in \mathcal{A}^*\}$ и функция преобразования R' получена сужением функции R на множество состояний S'. Будем говорить, что шкала $\mathcal{F}$ является:

полугрупповой, если ее можно гомоморфно отобразить на всякую свою подшкалу $\mathcal{F}_s$;

однородной, если она изоморфна всякой своей подшкале $\mathcal{F}_s$;

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

уравновешенной, если для любых операторных цепочек h и g из $[h]=[g]$ следует $|h|=|g|$;

универсальной, если для любых операторных цепочек h и g из $[h]=[g]$ следует $h=g$.

Полугрупповую шкалу $\mathcal{F}=\langle S, s_0, R\rangle$ можно рассматривать как конечно порожденный моноид $\langle S, *\rangle$ с множеством образующих $\{[a]\colon a\in\mathcal{A}\}$, единицей $s_0=[\lambda]$ и бинарной операцией $*$, где $[h]*[g]=[hg]$. В таком случае универсальная шкала соответствует свободному моноиду слов в алфавите $\mathcal{A}$, упорядоченная шкала — полугруппе с неразложимой единицей (т. е. $[\lambda]=[gh]$ влечет $g=h=\lambda$), а однородная шкала — моноиду, в котором выполняется закон левого сокращения (т. е. $[gh']=[gh'']$ влечет $[h']=[h'']$). Очевидно, что уравновешенная шкала является упорядоченной. В настоящей работе мы в основном будем иметь дело с уравновешенными полугрупповыми шкалами.

2.3. Динамические модели. *Детерминированной динамической моделью* (или просто *моделью*) M сигнатуры $(\mathcal{A},\mathcal{P})$ назовем пару $\langle\mathcal{F},\xi\rangle$, где

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

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

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

Пусть $\pi=\langle V,$ **вход**, **выход**, **тупик**, $B, T\rangle$ — некоторая пропозициональная операторная программа, а $M=\langle\mathcal{F},\xi\rangle$ — модель, базирующаяся на шкале $\mathcal{F}=\langle S, s_0, R\rangle$. Тогда *вычислением* программы π на модели M называется последовательность четверок (конечная или бесконечная)

$$r=(v_0,\delta_0,s_0,\lambda),(v_1,\delta_1,s_1,a_1),\ldots,(v_m,\delta_m,s_m,a_m),\ldots, \tag{1}$$

удовлетворяющая следующим требованиям:

1) $v_0=$ **вход**, $\delta_0\in\mathcal{C}$, s_0 — начальное состояние, λ — пустая о. ц.;

2) каждая четверка (v_i,δ_i,s_i,a_i), $i\geqslant 1$, состоит из вершины-преобразователя v_i (состояния программы), логического условия δ_i (набора истинностных значений элементарных отношений), состояния данных s_i и оператора a_i;

3) при $i=1,2,\ldots$ выполнены соотношения $v_{i+1}=T(v_i,\delta_i)$, $\delta_i=\xi(s_i)$, $s_{i+1}=R(s_i,a_{i+1})$ и $a_{i+1}=B(v_i)$;

4) последовательность (1) конечна в том и только том случае, когда для ее последнего члена (v_m,δ_m,s_m,a_m) справедливо включение $T(v_m,\delta_m)\in\{$**вход**, **тупик**$\}$.

Введенное здесь понятие вычисления в сущности соответствует полному пошаговому протоколу функционирования программы π в семантике M при начальных данных s_0. Ввиду детерминированности рассматриваемых программ и шкал непосредственно из определения вытекает, что каждая программа π имеет единственное вычисление (1) на заданной модели M; будем обозначать его $r(\pi, M)$. Длину последовательности (1) обозначим $|r(\pi, M)|$, полагая $|r(\pi, M)|=\infty$ в случае бесконечного вычисления. Подпоследовательность, состоящую из первых n элементов вычисления $r(\pi, M)$, назовем *n-префиксом* этого вычисления и будем обозначать $r(\pi, M)|_n$.

Если последовательность (1) оканчивается четверкой $(v_m, \delta_m, s_m, a_m)$, где $T(v_m, \delta_m) = \mathbf{вход}$ $(T(v_m, \delta_m) = \mathbf{тупик})$, то вычисление $r(\pi, M)$ будем называть *терминальным* (*тупиковым*). *Результатом терминального вычисления* назовем состояние данных s_m, достигнутое на последнем шаге, и обозначим его $[r(\pi, M)]$. Тупиковые и бесконечные вычисления считаем безрезультатными, полагая значение $[r(\pi, M)]$ для них неопределенным.

Для заданного вычисления (1) программы π на модели M и трассы

$$\alpha = \mathbf{вход} \xrightarrow{\delta_0} v_1 \xrightarrow{\delta_1} v_2 \xrightarrow{\delta_2} \dots \xrightarrow{\delta_{m-1}} v_m \xrightarrow{\delta_m} v_{m+1}, \qquad (2)$$

длины m, $1 \leqslant m \leqslant |r(\pi, M)|$, в программе π будем говорить, что вычисление $r(\pi, M)$ *реализует* трассу (2) на модели M, а трасса (2) *порождает* о. ц. $h = a_1 \dots a_m = B(v_1)B(v_2)\dots B(v_m)$. Легко видеть, что трассу (2) однозначно определяет m-префикс вычисления (1), на котором она реализуется, и при этом $s_m = [h]$. Будем также говорить, что трасса (2) реализуема на множестве моделей $\mathcal{M}$ (на шкале $\mathcal{F}$), если она реализуема на некоторой модели M из $\mathcal{M}$ (из $\mathcal{M}_{\mathcal{F}}$). Трассы α_1 и α_2 в программах π_1 и π_2 соответственно назовем *совместными* на множестве моделей $\mathcal{M}$ (на шкале $\mathcal{F}$), если эти трассы одновременно реализуемы на некоторой модели M из $\mathcal{M}$ (из $\mathcal{M}_{\mathcal{F}}$).

Отметим некоторые простейшие свойства вычислений, которые потребуются в дальнейшем при анализе эквивалентности программ.

Утверждение 1. *Пусть модели $M_1 = \langle \mathcal{F}, \xi_1 \rangle$, $M_2 = \langle \mathcal{F}, \xi_2 \rangle$ базируются на одной и той же шкале $\mathcal{F}$. Пусть последовательность* (1) *является m-префиксом вычисления $r(\pi, M_1)$ некоторой программы π на M_1, причем для каждого i, $0 \leqslant i \leqslant m$, выполняется $\xi_1(s_i) = \xi_2(s_i)$. Тогда $r(\pi, M_1)|_m = r(\pi, M_2)|_m$.*

Доказательство следует непосредственно из определения вычисления.

Утверждение 2. *Трасса* (2) *в программе π реализуема на модели $M = \langle \mathcal{F}, \xi \rangle$ в том и только том случае, когда для каждого i, $0 \leqslant i < m$, выполнено соотношение $\delta_i = \xi([B(v_1)B(v_2)\dots B(v_i)])$.*

Доказательство следует из определения вычисления, утверждения 1 и свойства реализуемости трассы на модели.

Утверждение 3. *Трасса* (2) *в программе π реализуема на шкале $\mathcal{F}$ в том и только том случае, когда для каждой пары i, j, $0 \leqslant i < j < m$, из равенства состояний $[B(v_1)B(v_2)\dots B(v_i)] = [B(v_1)B(v_2)\dots B(v_j)]$ следует равенство условий $\delta_i = \delta_j$.*

Для доказательства достаточно рассмотреть такую модель $M = \langle \mathcal{F}, \xi \rangle$, что $\xi([B(v_1)B(v_2)\dots B(v_i)]) = \delta_i$ для каждого i, $0 \leqslant i < m$, и воспользоваться утверждением 2.

Следствие 1. *Всякая трасса в программе π реализуема на упорядоченной шкале $\mathcal{F}$.*

Для доказательства здесь нужно воспользоваться тем фактом, что на упорядоченной шкале $\mathcal{F}$ для всякой трассы (2) при $i \neq j$ выполнено соотношение $[B(v_1)B(v_2)\dots B(v_i)] \neq [B(v_1)B(v_2)\dots B(v_j)]$.

Утверждение 4. *Трассы*

$$\alpha_1 = \mathbf{вход} \xrightarrow{\delta_1^0} v_1^1 \xrightarrow{\delta_1^1} v_1^2 \xrightarrow{\delta_1^2} \dots \xrightarrow{\delta_1^{m-1}} v_1^m \xrightarrow{\delta_1^m} v_1^{m+1},$$

$$\alpha_2 = \mathbf{вход} \xrightarrow{\delta_2^0} v_2^1 \xrightarrow{\delta_2^1} v_2^2 \xrightarrow{\delta_2^2} \dots \xrightarrow{\delta_2^{n-1}} v_2^n \xrightarrow{\delta_2^n} v_2^{n+1}$$

в программах π_1 и π_2 соответственно являются совместными на шкале $\mathcal{F}$ тогда и только тогда, когда обе трассы реализуемы на этой шкале и для всякой пары i, j, $0 \leqslant i < m$, $0 \leqslant j < n$, из равенства состояний $[B(v_1^1)B(v_1^2)\dots B(v_1^i)] = [B(v_2^1)B(v_2^2)\dots B(v_2^j)]$ следует равенство условий $\delta_1^i = \delta_2^j$.

Для доказательства достаточно рассмотреть модель $M = \langle \mathcal{F}, \xi \rangle$ с такой функцией означивания ξ, что $\xi([B(v_1^1)B(v_1^2)\dots B(v_1^i)]) = \delta_1^i$ и $\xi([B(v_2^1)B(v_2^2)\dots B(v_2^j)]) = \delta_2^j$ для всех i, j, $0 \leqslant i < m$, $0 \leqslant j < n$, и далее воспользоваться утверждением 2.

Следствие 2. *Трассы α_1 и α_2 в программах π_1 и π_2 соответственно являются совместными на уравновешенной шкале $\mathcal{F}$ в том и только том случае, когда для всякого i, $0 \leqslant i < \min(m, n)$, из равенства состояний $[B(v_1^1)B(v_1^2)\dots B(v_1^i)] = [B(v_2^1)B(v_2^2)\dots B(v_2^i)]$ следует равенство условий $\delta_1^i = \delta_2^i$.*

2.4. Проблема эквивалентности программ. Пусть π' и π'' — некоторые программы, M — произвольная модель, $\mathcal{M}$ — некоторое множество моделей, $\mathcal{F}$ — какая-либо шкала. Программы π' и π'' назовем:

эквивалентными на M ($\pi' \sim_M \pi''$), если $[r(\pi', M)] = [r(\pi'', M)]$, т. е. вычисления $r(\pi', M)$ и $r(\pi'', M)$ либо оба безрезультатны, либо оба являются терминальными вычислениями, имеющими одно и то же состояние s в качестве результата;

эквивалентными на $\mathcal{M}$ ($\pi' \sim_{\mathcal{M}} \pi''$), если соотношение $\pi' \sim_M \pi''$ выполнено для каждой модели M из $\mathcal{M}$;

эквивалентными на $\mathcal{F}$ ($\pi' \sim_{\mathcal{F}} \pi''$), если $\pi' \sim_{\mathcal{M}_{\mathcal{F}}} \pi''$.

Для заданного множества моделей $\mathcal{M}$ (заданной шкалы $\mathcal{F}$) фиксированной конечной сигнатуры $(\mathcal{A}, \mathcal{P})$ *проблема эквивалентности* относительно $\mathcal{M}$ (относительно $\mathcal{F}$) состоит в том, чтобы для произвольной пары программ π_1 и π_2 распознать выполнимость отношения $\pi' \sim_{\mathcal{M}} \pi''$ (соответственно, $\pi' \sim_{\mathcal{F}} \pi''$). В случае, когда исследуются алгоритмические аспекты проблемы эквивалентности, подразумевают, что множество моделей $\mathcal{M}$ (шкала $\mathcal{F}$) специфицированы некоторым эффективным способом (например, посредством аксиом пропозициональной динамической логики [74, 75], при помощи полугрупповых определяющих соотношений и т. п.).

Отметим некоторые простейшие свойства отношений эквивалентности пропозициональных динамических программ.

Утверждение 5. *Пусть каждая модель M' из множества моделей $\mathcal{M}_1$ является гомоморфным образом некоторой модели M'' из множества $\mathcal{M}_2$. Тогда для любых программ π_1 и π_2 из эквивалентности $\pi_1 \sim_{\mathcal{M}_2} \pi_2$ следует $\pi_1 \sim_{\mathcal{M}_1} \pi_2$.*

Следствие 3. *Пусть $\mathcal{U}$ — универсальная шкала. Тогда для любых программ π_1 и π_2 из эквивалентности $\pi_1 \sim_{\mathcal{U}} \pi_2$ следует эквивалентность $\pi_1 \sim_{\mathcal{M}} \pi_2$ для всякого множества моделей $\mathcal{M}$.*

Это очевидное утверждение играет ключевую роль в применении результатов теории схем программ [8, 20, 44, 45, 80] в практическом программировании. Семантика программ реальных универсальных систем операторного программирования соответствует множеству моделей $\mathcal{M}_1$, имеющих очень сложную структуру. Поэтому подавляющее большинство задач семантического анализа программ не допускает эффективного алгоритмического решения [86]. Один из методов преодоления этой принципиальной трудности таков. Выделив некоторые наиболее существенные свойства программных операторов и отношений на множестве данных, мы получаем возможность абстрагироваться от прочих особенностей и деталей семантики множества $\mathcal{M}_1$ и перейти к множеству моделей $\mathcal{M}_2$, имеющих более простое устройство и позволяющих получить алгоритмическое решение интересующих нас задач семантического анализа. Утверждение 5 гарантирует, что все инвариантные относительно функциональной эквивалентности позитивные свойства программ, выявленные при анализе этих программ в семантике множества $\mathcal{M}_2$, сохранятся и в семантике множества $\mathcal{M}_1$. В таком случае говорят, что семантика множества программ $\mathcal{M}_2$ *аппроксимирует* семантику множества $\mathcal{M}_1$. Исследование отношения аппроксимируемости семантик операторных программ проводилось в [11, 12, 44, 50].

Утверждение 6. *Для любой программы π существует такая редуцированная программа π', что $|\pi'| \leqslant |\pi|$ и $\pi \sim_{\mathscr{U}} \pi'$.*

Доказательство. Пусть $\pi = \langle V$, **вход**, **выход**, **тупик**, $B, T\rangle$. Обозначим через U_1 множество всех преобразователей π, для которых недоступна заключительная вершина **выход**, а через U_2 — множество всех преобразователей, недоступных для начальной вершины. Рассмотрим редуцированную программу $\pi' = \langle V'$, **вход**, **выход**, **тупик**, $B, T'\rangle$, где $V' = V \setminus (U_1 \cup U_2)$ и

$$T'(v, a) = \begin{cases} T(v, a), & \text{если } T(v, a) \in V', \\ \textbf{тупик} & \text{в противном случае,} \end{cases} \qquad v \in V' \cup \{\textbf{вход}\}, \quad a \in \mathscr{A}.$$

Очевидно, что программы π и π' обладают одинаковым множеством терминальных трасс и поэтому эквивалентны на любой модели $M = \langle U, \xi\rangle$.

§ 3. Разрешимые случаи проблемы эквивалентности программ

Как было отмечено в § 1, первый случай разрешимой проблемы эквивалентности пропозициональных операторных программ обнаружил Ю. И. Янов [60, 61]. Он установил разрешимость эквивалентности операторных программ в одном классе моделей, базирующихся на универсальных шкалах, и построил полную систему правил эквивалентных преобразований программ. Последующие результаты в этом направлении были установлены В. А. Непомнящим, М. А. Тайцлиным и А. А. Летичевским. Мы приведем соответствующие формулировки в терминах рассматриваемой здесь модели вычислений пропозициональных операторных программ.

Один из первых результатов о разрешимости отношения эквивалентности программ, существенно выходящий за пределы схем Янова, был установлен В. А. Непомнящим.

Теорема 1 [37]. *Пусть $\mathscr{F}$ — однородная уравновешенная шкала, на которой разрешима проблема тождества слов «верно ли, что $[h_1] = [h_2]$?». Для равновеликих операторных цепочек h_1 и h_2 положим $\rho(h_1, h_2) = \min(n\colon (\exists g_1, g_2)\ (|g_1| = |g_2| = n) \wedge ([h_1 g_1] = [h_2 g_2]))$. Тогда проблема эквивалентности пропозициональных операторных программ на шкале $\mathscr{F}$ разрешима, если существует такая монотонная неубывающая общерекурсивная функция $f(n)$, что для любых равновеликих операторных цепочек h_1 и h_2 выполнено условие*

$$\rho(h_1,h_2)<\infty \Longrightarrow (\exists g, g_1', g_2')\big(([h_1]=[gg_1'])\wedge([h_2]=[gh_2'])\wedge(|g_1'|=|g_2'|\leqslant f(\rho(h_1,h_2)))\big).$$

Последнее неравенство означает, что длина частных g_1' и g_2' от деления элементов h_1 и h_2 на их наибольший общий делитель рекурсивно ограничена длиной частных g_1, g_2 от деления наибольшего общего кратного $h_1 g_1 = h_2 g_2$ на h_1 и h_2. Предложенный в теореме разрешающий алгоритм имеет сверхэкспоненциальную временну́ю сложность.

Аналогичный результат был независимо установлен А. А. Летичевским для коммутативных полугрупповых шкал.

Теорема 2 [23, 25]. *Если шкала $\mathscr{F}$ соответствует свободной коммутативной полугруппе, то проблема эквивалентности программ на $\mathscr{F}$ разрешима.*

Развивая результаты [23], М. А. Тайцлин в [59] исследовал проблему эквивалентности программ на шкалах, соответствующих коммутативным полугруппам, и установил следующий общий результат.

Теорема 3 [59]. *Если однородная шкала $\mathscr{F}$ соответствует сократимой коммутативной полугруппе, то для разрешимости проблемы эквивалентности программ на шкале $\mathscr{F}$ необходимо и достаточно, чтобы проблема эквивалентности программ была разрешима на шкале, соответствующей максимальной подгруппе полугруппы $\mathscr{F}$.*

При этом, как показал Ю. Ш. Гуревич, проблема эквивалентности программ на $\mathcal{F}$ неразрешима, если $\mathcal{F}$ соответствует абелевой группе по меньшей мере второго ранга.

В [4, 24, 26] была разработана новая техника доказательства разрешимости эквивалентности программ — метод устранения несущественных ветвлений. С его помощью А. А. Летичевский доказал следующую теорему.

Теорема 4 [4]. *Если для однородной упорядоченной шкалы $\mathcal{F}$ разрешима проблема тождества слов «верно ли, что $[h_1]=[h_2]$?», то на $\mathcal{F}$ разрешима и проблема эквивалентности программ.*

Предложенный в [4] алгоритм опирается на рекурсивное устранение несущественных ветвлений с целью приведения пары эквивалентных программ к общему виду и поэтому имеет экспоненциальную сложность.

Теорема 5 [24]. *Проблема эквивалентности программ на шкале $\mathcal{F}$ разрешима в тех случаях, когда шкала $\mathcal{F}$ соответствует абелевой группе ранга 1, свободной группе или свободному произведению подгрупп, каждая из которых либо циклична, либо конечна.*

И, наконец, в [24, 26] был установлен наиболее общий критерий разрешимости проблемы эквивалентности программ на полугрупповых шкалах, обобщающий ранее полученный критерий разрешимости Тайцлина для коммутативных шкал.

Теорема 6 [24, 26]. *Пусть однородная шкала $\mathcal{F}$ обладает следующими свойствами:*

на $\mathcal{F}$ разрешима проблема тождества слов «верно ли, что $[h_1]=[h_2]$?»;

для любых операторных цепочек g, h_1, h_2 выполнено соотношение $[h_1 g]=[h_2 g]$ (закон правого сокращения);

имеется алгоритм, для любых операторных цепочек g и g' распознающий, существуют ли такие операторные цепочки h' и h'', что $[h'h'']=[h''h']=[\lambda]$ и $gh'=g'$ ($h''g=g'$).

Тогда проблема эквивалентности на шкале $\mathcal{F}$ разрешима в том и только том случае, когда она разрешима на шкале $\mathcal{F}'$, соответствующей максимальной подгруппе $\mathcal{F}$.

Проблема эквивалентности программ исследовалась также для классов шкал, соответствующих полугруппам с правым нулем (см. [5, 21, 71]) и группам специального вида [28]. В [27] отмечалось, что все построенные разрешающие алгоритмы обладают чрезмерно большой трудоемкостью, и высказывалось предположение, что в некоторых практически важных случаях можно построить существенно более быстрые процедуры распознавания эквивалентности программ. В настоящей работе мы смогли подтвердить эту гипотезу, разработав общий подход к построению полиномиальных по быстродействию алгоритмов, разрешающих эквивалентность пропозициональных операторных программ на некоторых шкалах. Ограничимся здесь рассмотрением уравновешенных полугрупповых шкал.

§ 4. Уравновешенные полугрупповые шкалы

В этом параграфе представлен новый подход к построению эффективных алгоритмов, разрешающих проблему эквивалентности пропозициональных детерминированных программ на уравновешенных полугрупповых шкалах путем сведения ее к проблеме тождества слов в специальных полугруппах. Мы опишем алгоритм сведéния, оценим его сложность и обоснуем корректность. Мы также покажем на ряде примеров, что проблема эквивалентности пропозициональных детерминированных программ конечной фиксированной сигнатуры решается в отдельных случаях за полиномиальное время.

Введем вспомогательные понятия, в терминах которых будут описаны разрешающие алгоритмы.

Пусть $\mathcal{F} = \langle S, s_0, R\rangle$ — полугрупповая уравновешенная шкала. Рассмотрим $\mathcal{F}$ как моноид $\langle S, *\rangle$ и обозначим через $\mathcal{F} \times \mathcal{F}$ прямое произведение моноидов. В этом произведении выделим подмножество $E = \{\langle [g'], [h']\rangle\colon |g'| = |h'|\}$. Шкала $\mathcal{F}$ уравновешенная, поэтому E образует подполугруппу моноида $\mathcal{F} \times \mathcal{F}$, порожденную множеством образующих $\{\langle [a], [b]\rangle\colon a, b \in \mathcal{A}\}$.

Рассмотрим некоторую конечно порожденную полугруппу W с бинарной операцией $\circ$ и единицей e. Пусть в W выделена подполугруппа U и пара элементов w^+, w^*. Пусть k_0 — натуральное число. Четверку $K = \langle W, U, w^+, w^*\rangle$ назовем *k_0-критериальной системой* для уравновешенной полугрупповой шкалы $\mathcal{F}$, если K и $\mathcal{F}$ удовлетворяют следующим требованиям.

С1. Существует такой гомоморфизм φ полугруппы E в полугруппу U, что соотношение $[h] = [g] \iff w^+ \circ \varphi(\langle [g], [h]\rangle) \circ w^* = e$ выполнено для всех операторных цепочек g и h из $\mathcal{A}^*$.

С2. Для каждого элемента w из класса смежности $U \circ w^*$ существует не более k_0 левых обратных элементов из класса смежности $w^+ \circ U$, т. е. уравнение $y \circ w = e$ имеет не более k_0 различных решений y, представимых в виде $y = w^+ \circ u$, где $u \in U$.

Заметим, что условие С2 выполнено при $k_0 = 1$, если W — группа, и при $k_0 = m$, если класс смежности $w^+ \circ U$ содержит не более m элементов.

Теорема 7. *Пусть $\mathcal{F}$ — уравновешенная полугрупповая шкала сигнатуры $(\mathcal{A}, \mathcal{P})$, имеющая k_0-критериальную систему $K = \langle W, U, w^+, w^*\rangle$, $k_0 > 0$. Тогда если в полугруппе W проблема тождества слов «верно ли, что $w_1 = w_2$?» разрешима за время $t(m)$, где $m = \max(|w_1|, |w_2|)$, то проблема эквивалентности «верно ли, что $\pi_1 \sim_{\mathcal{F}} \pi_2$?» пропозициональных операторных программ на шкале $\mathcal{F}$ разрешима за время $c_1 n^2 (t(c_2 n^2) + \log n)$, где $n = \max(|\pi_1|, |\pi_2|)$, а константы c_1 и c_2 зависят только от k_0, $|\mathcal{A}|$, $|\mathcal{P}|$ и гомоморфизма φ.*

Доказательство. Вначале опишем алгоритм, разрешающий эквивалентность программ на шкале $\mathcal{F}$ при помощи критериальной системы K, а затем оценим его сложность и обоснуем корректность.

Для заданной пары пропозициональных детерминированных программ $\pi_i = \langle V_i, \textbf{вход}, \textbf{выход}, B_i, T_i\rangle$, $i = 1, 2$, рассмотрим помеченный ориентированный граф Γ, вершинами которого являются всевозможные тройки вида $(\textbf{вход}, \textbf{вход}, w^+)$ или (v_1, v_2, w), где $v_i \in V_i \cup \{\textbf{выход}, \textbf{тупик}\}$, $i = 1, 2$, а w — элемент из класса смежности $w^+ \circ U$. Вершину $(\textbf{вход}, \textbf{вход}, w^+)$ назовем *корнем* графа Γ. Множество вершин графа разбито на три попарно непересекающихся подмножества, X_1, X_2 и X_3, где

$$X_1 = \{(v_1, v_2, w)\colon w \circ w^* \neq e, \ v_1 \in V_1, \ v_2 \in V_2\},$$

$$X_2 = \{(v_1, v_2, w)\colon w \circ w^* = e, \ v_1 \in V_1, \ v_2 \in V_2\} \cup \{(\textbf{вход}, \textbf{вход}, w^+)\};$$

а все остальные вершины относятся к X_3.

Каждая дуга графа Γ помечена парой условий (δ_1, δ_2) из $\mathcal{C} \times \mathcal{C}$, причем дуги, исходящие из одной вершины, помечены различными парами условий. Для каждой вершины x графа Γ определяется множество пометок Δ_x:

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

Каждая вершина из подмножества X_1 имеет $4^{|\mathcal{P}|}$ исходящих дуг, каждая вершина из подмножества X_2 имеет $2^{|\mathcal{P}|}$ исходящих дуг, а вершины из множества X_3 не имеют ни одной исходящей дуги. Отношение инцидентности вершин и дуг графа Γ определяется следующим образом.

Для каждой вершины $x = (v_1, v_2, w)$ из множества $X_1 \cup X_2$ и пары условий (δ_1, δ_2) из множества пар Δ_x, соответствующего вершине x, дуга, помеченная (δ_1, δ_2), ведет из x в вершину $x' = (v_1', v_2', w')$, где $v_i' = T_i(v_i, \delta_i)$, $i = 1, 2$, а элемент w' из класса смежности $w^+ \circ U$ равен либо $w \circ w^*$ (если v_1' или v_2' не является преобразователем), либо $w \circ \varphi(\langle[B_1(v_1')], [B_2(v_2')]\rangle)$ — если v_1' и v_2' являются преобразователями программ π_1 и π_2 соответственно.

Вершину $x = (v_1, v_2, w)$ графа Г будем называть *опровергающей*, если $\{v_1, v_2\} \cap \{\textbf{выход}, \textbf{тупик}\} \neq \varnothing$ и $v_1 \neq v_2$, или если $v_1 = v_2 = \textbf{выход}$ и $w \neq e$.

Алгоритм анализа эквивалентности программ на шкале $\mathcal{F}$ проверяет вершины графа Г и объявляет схемы π_1 и π_2 эквивалентными на $\mathcal{F}$ в том и только том случае, когда выполнены следующие два условия.

R1. Ни одна опровергающая вершина не достижима по ориентированному маршруту из корня Г.

R2. Для любых преобразователей v_1 и v_2, $v_1 \in V_1$, $v_2 \in V_2$, не более k_0 различных вершин вида (v_1, v_2, w) достижимы по ориентированным маршрутам из корня Г.

Очевидно, что для проверки этих условий требуется рассмотреть не более $k_0 |V_1| |V_2| + 3$ вершин графа Г. Поэтому предложенный алгоритм всегда завершает работу с тем или иным результатом. Отсюда следует также, что для каждой из рассмотренных вершин $x = (v_1, v_2, w)$ элемент w полугруппы W имеет вид $w = w^+ \cdot \varphi(\langle[g], [h]\rangle)$, где $|g| = |h| \leqslant k_0 |V_1| |V_2|$. Согласно описанию графа Г, переход из вершины $x = (v_1, v_2, w)$ в вершину $x' = (v_1', v_2', w')$ по дуге с пометкой (δ_1, δ_2) возможен в том и только том случае, когда $v_1' = T_1(v_1, \delta_1)$, $v_2' = T_2(v_2, \delta_2)$ и справедливо одно из равенств $w' = w \circ \varphi(\langle[B_1(v_1')], [B_2(v_2')]\rangle)$ или $w' = w \circ w^*$ в зависимости от вида v_1' и v_2'. Сравнение преобразователей программ осуществимо за время $O(\log(|V_1| |V_2|))$, сравнение элементов полугруппы W — за время $t(c_2 |V_1| |V_2|)$, где константа c_2 зависит от k_0 и гомоморфного отображения φ полугруппы E в полугруппу U. Совмещая оценку числа рассматриваемых вершин и оценку трудоемкости проведения одной дуги между парой вершин, получаем искомую оценку сложности предложенного алгоритма.

Обоснование корректности алгоритма опирается на четыре вспомогательных утверждения.

Лемма 1. *Последовательность $x_0, x_1, \ldots, x_m, x_{m+1}$, $m \geqslant 0$, вершин графа Г, где x_0 — корень графа Г и $x_i = (v_1^i, v_2^i, w_i)$ для $i = 1, \ldots, m+1$, образует маршрут $x_0 \xrightarrow{(\delta_1^0, \delta_2^0)} x_1 \xrightarrow{(\delta_1^1, \delta_2^1)} \ldots \xrightarrow{(\delta_1^{m-1}, \delta_2^{m-1})} x_m \xrightarrow{(\delta_1^m, \delta_2^m)} x_{m+1}$ в том и только том случае, когда трассы*

$$\alpha_1 = \textbf{вход} \xrightarrow{\delta_1^0} v_1^1 \xrightarrow{\delta_1^1} v_1^2 \xrightarrow{\delta_1^2} \ldots \xrightarrow{\delta_1^{m-1}} v_1^m \xrightarrow{\delta_1^m} v_1^{m+1},$$
$$\alpha_2 = \textbf{вход} \xrightarrow{\delta_2^0} v_2^1 \xrightarrow{\delta_2^1} v_2^2 \xrightarrow{\delta_2^2} \ldots \xrightarrow{\delta_2^{m-1}} v_2^m \xrightarrow{\delta_2^m} v_2^{m+1}$$

программ π_1 и π_2 совместны на модели $\mathcal{F}$, причем $w_i = w^+ \circ \varphi(\langle h_1^i, h_2^i\rangle)$, где $h_1^i = B_1(v_1^1)B_1(v_1^2)\ldots B_1(v_1^i)$, $h_2^i = B_2(v_2^1)B_1(v_2^2)\ldots B_1(v_2^i)$, $i = 0, 1, \ldots, m$.

Доказательство проводится индукцией по m с учетом структуры графа Г, условия C1 определения критериальной системы и следствия 2.

Лемма 2. *Пусть $\delta^0, \delta^1, \ldots, \delta^m$ — последовательность условий из $\mathcal{C}$, $1 \leqslant i \leqslant m$. Рассмотрим пару маршрутов*

$$\beta_1 = v_1^0 \xrightarrow{\delta^0} v_1^1 \xrightarrow{\delta^1} v_1^2 \xrightarrow{\delta^2} \ldots \xrightarrow{\delta^{m-1}} v_1^m \xrightarrow{\delta^m} v_1^{m+1},$$
$$\beta_2 = v_2^0 \xrightarrow{\delta^0} v_2^1 \xrightarrow{\delta^1} v_2^2 \xrightarrow{\delta^2} \ldots \xrightarrow{\delta^{m-1}} v_2^m \xrightarrow{\delta^m} v_2^{m+1}$$

в программах π_1 и π_2 соответственно. Тогда для каждого элемента w из класса смежности $w^+ \circ U$ в графе Г из вершины $x_0 = (v_1^0, v_2^0, w_0)$ исходит маршрут $x_0 \xrightarrow{(\delta^0, \delta^0)} x_1 \xrightarrow{(\delta^1, \delta^1)} \ldots \xrightarrow{(\delta^{m-1}, \delta^{m-1})} x_m \xrightarrow{(\delta^m, \delta^m)} x_{m+1}$, проходящий через вершины $x_i = (v_1^i, v_2^i, w_0 \circ \varphi(\langle[h_1^i], [h_2^i]\rangle))$, $i = 1, \ldots, m$, где $h_1^i = B_1(v_1^1)B_1(v_1^2)\ldots B_1(v_1^i)$, $h_2^i = B_2(v_2^1)B_2(v_2^2)\ldots B_2(v_2^i)$.

Доказательство проводится индукцией по m с учетом структуры графа Γ, а также того факта, что для всякого условия δ из каждой вершины (v_1, v_2, w), принадлежащей $X_1 \cup X_2$, при $v_1 \in V_1$, $v_2 \in V_2$ исходит дуга, помеченная парой (δ, δ).

Лемма 3. *Если π_1 и π_2 — редуцированные программы, то $\pi_1 \sim_{\mathcal{F}} \pi_2$ в том и только том случае, когда Γ удовлетворяет требованию* R1.

Доказательство. Воспользуемся леммой 1 и условием C1 из определения критериальной системы.

Необходимость. Пусть R1 не выполнено и опровергающая вершина $x = (v_1, v_2, w)$ достижима из корня. Рассмотрим случай $v_1 = v_2 =$ **выход** и $w \neq e$. Тогда согласно лемме 1 в программах π_1 и π_2 имеется пара совместных трасс, оканчивающихся в выходах этих программ:

$$\alpha_1 = \textbf{вход} \xrightarrow{\delta_1^0} v_1^1 \xrightarrow{\delta_1^1} v_1^2 \xrightarrow{\delta_1^2} \ldots \xrightarrow{\delta_1^{m-1}} v_1^m \xrightarrow{\delta_1^m} \textbf{выход},$$
$$\alpha_2 = \textbf{вход} \xrightarrow{\delta_2^0} v_2^1 \xrightarrow{\delta_2^1} v_2^2 \xrightarrow{\delta_2^2} \ldots \xrightarrow{\delta_2^{m-1}} v_2^m \xrightarrow{\delta_2^m} \textbf{выход}.$$

Обе трассы реализуются на некоторой модели M, базирующейся на шкале $\mathcal{F}$, и поэтому из определения вычисления программ на модели следует, что $[r(\pi_1, M)] = [h_1]$, $[r(\pi_2, M)] = [h_2]$, где $h_1 = B_1(v_1^1)B_1(v_1^2)\ldots B_1(v_1^m)$, $h_2 = B_2(v_2^1)B_1(v_2^2)\ldots B_1(v_2^m)$. С другой стороны, из леммы 1 также следует, что $w = w^+ \circ \varphi(\langle [h_1], [h_2] \rangle) \circ w^*$. Принимая во внимание условие C1 и соотношение $w \neq e$, приходим к выводу о том, что $[r(\pi_1, M)] \neq [r(\pi_2, M)]$.

Обратимся теперь к случаю $v_1 \in V_1$, $v_2 =$ **выход**. Согласно лемме 1 в программах π_1 и π_2 имеется пара совместных на шкале $\mathcal{F}$ трасс α_1 и α_2 одинаковой длины, причем маршрут α_1 оканчивается в преобразователе v_1, а α_2 — в выходе π_2. Программа π_1 редуцированная, поэтому выход π_1 доступен из преобразователя v_1 по некоторому маршруту β_1. Ввиду уравновешенности $\mathcal{F}$, трассы $\alpha_1\beta_1$ и α_2 будут также совместны на рассматриваемой шкале. Это означает, что для некоторой модели M, базирующейся на $\mathcal{F}$, оба вычисления, $r(\pi_1, M)$ и $r(\pi_2, M)$, будут терминальными, причем справедливы отношения $|r(\pi_1, M)| = |\alpha_1\beta_1| > |\alpha_2| = |r(\pi_2, M)|$. Принимая во внимание уравновешенность $\mathcal{F}$, приходим к выводу о том, что $[r(\pi_1, M)] \neq [r(\pi_2, M)]$.

Остальные случаи достижимости опровергающих вершин в Γ рассматриваются аналогично.

Достаточность. Предположим, что на некоторой модели M из $\mathcal{M}_{\mathcal{F}}$ терминальные вычисления

$$r(\pi_1, M) = (\textbf{вход}, \delta_1^0, s_1^0, \lambda), (v_1^1, \delta_1^1, s_1^1, a_1^1), \ldots, (v_1^n, \delta_1^n, s_1^n, a_1^n),$$
$$r(\pi_2, M) = (\textbf{вход}, \delta_2^0, s_2^0, \lambda), (v_2^1, \delta_2^1, s_2^1, a_2^1), \ldots, (v_2^m, \delta_2^m, s_2^m, a_2^m)$$

оканчиваются с различными результатами: s_1^m и s_2^n. Если длина одного из вычислений, — например, $r(\pi_1, M)$, — больше длины другого (в данном случае $r(\pi_2, M)$), то согласно лемме 1 и с учетом совместности трасс

$$\alpha_1 = \textbf{вход} \xrightarrow{\delta_1^0} v_1^1 \xrightarrow{\delta_1^1} v_1^2 \xrightarrow{\delta_1^2} \ldots \xrightarrow{\delta_1^{m-1}} v_1^m \xrightarrow{\delta_1^m} v_1^{m+1},$$
$$\alpha_2 = \textbf{вход} \xrightarrow{\delta_2^0} v_2^1 \xrightarrow{\delta_2^1} v_2^2 \xrightarrow{\delta_2^2} \ldots \xrightarrow{\delta_2^{m-1}} v_2^m \xrightarrow{\delta_2^m} \textbf{выход},$$

где $v_1^{m+1} \neq$ **выход**, приходим к выводу о том, что из корня графа Γ достижима опровергающая вершина $x = (v_1^{m+1}, \textbf{выход}, w)$. Если же оба вычисления имеют одинаковую длину, т. е. $m = n$, то на основании той же леммы и условия C1 приходим к заключению о том, что из корня Γ достижима опровергающая вершина $x = (\textbf{выход}, \textbf{выход}, w)$, где $w = w^+ \circ \varphi(\langle s_1^m, s_2^n \rangle) \circ w^*$.

Варианты других пар вычислений $r(\pi_1, M)$ и $r(\pi_2, M)$, имеющих различные результаты, рассматриваются аналогично.

Лемма 4. *Для редуцированных программ* π_1 *и* π_2 *из* R1 *следует* R2.

Доказательство. Пусть имеется такая пара преобразователей, v_1 из V_1 и v_2 из V_2, что из корня графа Г достижимы по крайней мере k_0+1 различных вершин: $(v_1, v_2, w_1), \ldots, (v_1, v_2, w_{k_0+1})$. Программа π_1 — редуцированная, поэтому ее выход доступен из преобразователя v_1, т. е. для некоторой последовательности условий $\delta^0, \delta^1, \ldots, \delta^m$ справедливо равенство $T(\ldots T(T(v_1, \delta_1), \delta_2), \ldots, \delta_m) = \textbf{выход}$. Рассмотрим пару маршрутов

$$\beta_1 = v_1 \xrightarrow{\delta^0} v_1^1 \xrightarrow{\delta^1} v_1^2 \xrightarrow{\delta^2} \ldots \xrightarrow{\delta^{m-1}} v_1^m \xrightarrow{\delta^m} \textbf{выход},$$
$$\beta_2 = v_2 \xrightarrow{\delta^0} v_2^1 \xrightarrow{\delta^1} v_2^2 \xrightarrow{\delta^2} \ldots \xrightarrow{\delta^{m-1}} v_2^m \xrightarrow{\delta^m} v_2^{m+1}.$$

Согласно лемме 2 из каждой вершины $x_i = (v_1, v_2, w_i)$, $i = 1, \ldots, k_0+1$, графа Г (значит, и из корня этого графа) достижима вершина $x_i' = (\textbf{выход}, v_2^{m+1}, w_i')$. При $v_2^{m+1} \neq \textbf{выход}$ каждая вершина x_i' является опровергающей. Если же $v_2^{m+1} = \textbf{выход}$, то по лемме 2 для каждого i, $i = 1, \ldots, k_0+1$, выполнено соотношение $w_i' = w_i \circ \varphi(\langle h_1, h_2 \rangle) \circ w^*$, где $h_1 = B_1(v_1)B_1(v_1^1)\ldots B_1(v_1^m)$ и $h_2 = B_2(v_2)B_1(v_2^1)\ldots B_2(v_2^m)$. С учетом свойства C2 критериальной системы заключаем, что по крайней мере для одного j, $1 \leqslant j \leqslant k_0+1$, выполнено неравенство $w_j' \neq e$. Значит, и в этом случае из корня графа Г достижима опровергающая вершина $x_j' = (\textbf{выход}, \textbf{выход}, w_j')$.

Таким образом, нарушение условия R2 влечет за собой невыполнение условия R1.

Леммы 3 и 4 гарантируют корректность предложенного алгоритма в случае, когда программы π_1 и π_2 редуцированы. Для завершения доказательства теоремы нужно воспользоваться утверждением 6 и следствием 3. Теорема 1 полностью доказана.

§ 5. Примеры быстрых алгоритмов разрешения эквивалентности программ

Обратимся к примерам применения теоремы 1 для построения эффективных процедур распознавания эквивалентности пропозициональных детерминированных программ на уравновешенных полугрупповых шкалах.

5.1. Универсальные шкалы. Рассмотрим универсальную шкалу $\mathcal{U}$ сигнатуры $\mathscr{A}$; очевидно, это уравновешенная полугрупповая шкала, соответствующая свободному моноиду с множеством образующих $\mathscr{A}$. Вычислительная модель программ на шкале $\mathcal{U}$ соответствует модели вычислений схем Янова с универсальным сдвигом [6, 60, 61]. Покажем, что разрешимость проблемы эквивалентности схем Янова — простое следствие теоремы 1.

Теорема 8. *Проблема эквивалентности программ на универсальной шкале разрешима за время* $O(n^2 \log n)$.

Доказательство. Пусть W — полугруппа, образованная элементами 0 и 1 булевой алгебры с операцией конъюнкции $\wedge$. Тогда $K = \langle W, W, 1, 1 \rangle$ является 1-критериальной системой для шкалы $\mathcal{F}$. Чтобы убедиться в этом, достаточно рассмотреть гомоморфное отображение $\varphi\colon E \to W$, где

$$\varphi(\langle [a], [b] \rangle) = \begin{cases} 1, & a = b, \\ 0, & a \neq b, \end{cases}$$

для любых операторов a и b. Применяя методику распознавания эквивалентности программ, описанную в теореме 1, получаем алгоритм, разрешающий проблему эквивалентности программ на универсальной шкале за время $O(n^2 \log n)$. Этот пример имеет демонстрационный характер и призван свидетельствовать, что случай классических схем Янова органически вписывается в общую методику доказательства быстрой разрешимости проблемы эквивалентности операторных программ.

5.2. Коммутативные шкалы. Рассмотрим шкалу $\mathcal{F}_{fc}$ сигнатуры $\mathcal{A}$, соответствующую свободной коммутативной полугруппе с множеством образующих $\mathcal{A} = \{a^1, \ldots, a^N\}$. Равенство $[h] = [g]$ на шкале $\mathcal{F}_{fc}$ выполнено в том и только том случае, когда для каждого оператора a из $\mathcal{A}$ число его вхождений в операторную цепочку h совпадает с числом его же вхождений в цепочку g. Шкала $\mathcal{F}_{fc}$ описывает семантику операторных программ, все элементарные операторы a^i которых имеют вид $x_i := f_i(x_i)$, а логические условия — $P(x_1, \ldots, x_N)$. Разрешимость проблемы эквивалентности программ на свободной коммутативной полугруппе была установлена независимо в [22, 37, 59]. В [14, 52] был предложен полиномиальный алгоритм распознавания эквивалентности программ на $\mathcal{F}_{fc}$. Возможность построения разрешающего алгоритма следует из теоремы 1.

Теорема 9 [52]. *Проблема эквивалентности программ на свободной коммутативной шкале разрешима за время $O(n^2 \log n)$.*

Доказательство. Рассмотрим свободную абелеву группу Z ранга N, порожденную элементами $q_1, \ldots, q_N$, где $N = |\mathcal{A}|$. Тогда $K = \langle Z, Z, e, e\rangle$ образует 1-критериальную систему для $\mathcal{F}_{fc}$, если воспользоваться гомоморфизмом $\varphi(\langle [a^i], [a^j]\rangle) = q_i \circ q_j^{-1}$ для каждой пары операторов a^i и a^j из $\mathcal{A}$. Каждое слово w из Z однозначно представляется двоичным вектором длины $O(\log|w|)$, поэтому непосредственно из теоремы 1 вытекает верхняя оценка $O(n^2 \log n)$ сложности распознавания эквивалентности программ на $\mathcal{F}_{fc}$.

5.3. Частично коммутативные шкалы. Пусть I — симметричное рефлексивное отношение перестановочности на множестве операторов $\mathcal{A}$. Рассмотрим шкалу $\mathcal{F}_c^I$, где $I \subseteq \mathcal{A} \times \mathcal{A}$, соответствующую полугруппе с множеством образующих $\mathcal{A} = \{a^1, \ldots, a^N\}$ и множеством определяющих соотношений $\{a^i a^j = a^j a^i \colon (a^i a^j) \in I\}$. При помощи таких шкал описываются семантики операторных программ, элементарные операторы a^i которых имеют вид $x_i := f_i(\bar{y}_i)$, а логические условия — $P(x_1, \ldots, x_N)$. Для перестановочности операторов $x_i := f_i(\bar{y}_i)$ и $x_j := f_j(\bar{y}_j)$ достаточно одновременного выполнения условий $x_i \neq x_j$, $x_i \notin \bar{y}_j$ и $x_j \notin \bar{y}_i$. Всевозможные пары перестановочных операторов образуют множество I. В частном случае (при $I = \mathcal{A} \times \mathcal{A}$) шкала $\mathcal{F}_c^I$ совпадает со свободной коммутативной шкалой $\mathcal{F}_{fc}$. Шкалу $\mathcal{F}_c^I$ будем называть *I-коммутативной шкалой*. Разрешимость проблемы эквивалентности на шкалах такого вида следует из результатов работ [4, 23]. Воспользовавшись теоремой 1, мы покажем, что эквивалентность программ на $\mathcal{F}_c^I$ разрешима за полиномиальное время.

Для построения критериальной системы введем некоторые вспомогательные понятия и рассмотрим ряд характерных свойств I-коммутативных шкал. Очевидно, $\mathcal{F}_c^I$ является уравновешенной полугрупповой шкалой, на которой действуют законы левого и правого сокращения, т. е. из $[h' h_1 h''] = [h' h_2 h'']$ следует $[h_1] = [h_2]$. Непосредственно из определения $\mathcal{F}_c^I$ вытекает справедливость следующих высказываний.

Утверждение 7. *При $[h_1] = [h_2]$ число вхождений каждого оператора a, $a \in \mathcal{A}$, в о. ц. h_1 равно числу его вхождений в о. ц. h_2.*

Утверждение 8. *Если операторные цепочки h_1' и h_2' состоят из операторов множества $\mathcal{A}'$, а цепочки h_1'' и h_2'' — из операторов множества $\mathcal{A}''$, причем $\mathcal{A}' \cap \mathcal{A}'' = \varnothing$, то*

$$[h_1' h_1''] = [h_2' h_2''] \iff (([h_1'] = [h_2']) \wedge ([h_1''] = [h_2''])).$$

Утверждение 9. *Если в о. ц. g первое вхождение оператора a предшествует первому вхождению оператора b, а в о. ц. h первое вхождение b предшествует первому вхождению a, и при этом $[g] = [h]$, то $(a, b) \in I$.*

Пару операторных цепочек (g, h) назовем *I-ортогональной*, если выполнены следующие условия:

1) $|g|=|h|$;

2) ни один оператор a из $\mathcal{A}$ не входит одновременно в g и в h;

3) любой оператор цепочки g перестановочен с любым оператором цепочки h.

Пусть g' и h' — операторные цепочки одинаковой длины. Будем говорить, что тройка о. ц. (H, h, g) является *левым* (*правым*) *ортогональным разложением* пары (g', h'), если операторные цепочки h и g образуют I-ортогональную пару и выполнены равенства $[h']=[Hh]$ и $[g']=[Hg]$ (соответственно, $[h']=[hH]$ и $[g']=[gH]$).

Из утверждения 8 и определения ортогональной пары следует

Утверждение 10. *Если обе тройки, (H_1, h_1, g_1) и (H_2, h_2, g_2), образуют левое (или правое) разложение пары операторных цепочек (g', h'), то $[H_1]=[H_2]$, $[h_1]=[h_2]$ и $[g_1]=[g_2]$.*

Лемма 5. *Пусть операторные цепочки h', h'', g' и g'' таковы, что $|h'|=|g'|$ и $|h''|=|g''|$. Тогда равенство $[h'h'']=[g'g'']$ имеет место в том и только том случае, когда найдутся такие цепочки H, h, g, G, что тройка (H, h, g) образует левое ортогональное разложение пары (h', g'), а тройка (G, g, h) — правое ортогональное разложение пары (h'', g'').*

Доказательство. Достаточность. Из определения ортогональных разложений следует, что $[h'h'']=[HhgG]$ и $[g'g'']=[HghG]$. Пара (g, h) является I-ортогональной, поэтому любой оператор из g перестановочен с любым оператором из h. Отсюда $[hg]=[gh]$. Следовательно, $[HhgG]=[HghG]$.

Необходимость. Сначала индукцией по $l=|h'|$ покажем, что пара (h', g') имеет левое ортогональное разложение (H, h_1, g_1).

Если $|h'|=|g'|=0$, то $H=h_1=g_1=\lambda$.

Предположим, что утверждение леммы справедливо при $l \leqslant n$. Пусть $|h'|=|g'|=n+1$. Тогда по предположению индукции $h'=HH_1a$, $g'=HG_1b$, причем пара (H_1, G_1) является I-ортогональной. С учетом правила левого сокращения для $\mathcal{F}_c^I$ имеем $[H_1ah'']=[G_1bg'']$. Исследуем различные варианты операторов a и b.

I. Допустим, что оператор a содержится в G_1. Тогда цепочка G_1 представима в виде G_3aG_4, где G_3 — наибольший префикс G_1, не содержащий вхождений a. По предположению индукции цепочки H_1 и G_1 образуют ортогональную пару, поэтому оператор a перестановочен со всеми операторами цепочки H_1. В результате получаем $[H_1a]=[aH_1]$ и $[aH_1h'']=[G_3aG_4bg'']$. В силу утверждения 7 из последнего равенства следует, что всякий оператор a' цепочки G_3 содержится в цепочке H_1h''. Таким образом, для каждого оператора a' из G_3 его первое вхождение в цепочку G_3aG_4bg'' предшествует первому вхождению оператора a, тогда как, напротив, в цепочке aH_1h'' первое вхождение a предшествует первому вхождению a'. Согласно утверждению 9 это означает, что оператор a перестановочен со всеми операторами цепочки G_3, поэтому справедливо равенство $[G_3a]=[aG_3]$. Учитывая равенства $[H_1ah'']=[aH_1h'']$ и $[G_1bg'']=[aG_3G_4bg'']$, а также левую сократимость $\mathcal{F}_c^I$, мы можем применить индуктивное предположение по отношению к операторным цепочкам H_1, h'', G_3G_4b и g'' и получить искомый результат.

Аналогичные рассуждения применимы и в случае, когда оператор b содержится в H_1.

II. Допустим, что оператор a не имеет вхождений в о. ц. G_1, а оператор b — в H_1. Помня о том, что пара (G_1, H_1) является I-ортогональной и $[H_1ah'']=[G_1bg'']$, мы можем заключить, воспользовавшись утверждением 7, что все операторы из G_1b входят в операторную цепочку H_1ah'', причем только в ее суффикс ah'', а все операторы из H_1a — в цепочку G_1bg'', причем только в суффикс bg''. Тогда из утверждения 9 следует, что оператор a перестановочен со всеми операторами из $G_1 b$, а оператор b — со всеми операторами из $H_1 a$. Если $a \neq b$, то $(G_1 b, H_1 a)$ — ортогональная пара. Если $a = b$, то $[H_1 a h''] = [aH_1 h'']$, $[G_1 b g''] = [aG_1 g'']$, и снова можно применить индуктивное предположение по отношению к H_1, h'', G_1 и g''.

Правомерность индуктивного перехода обоснована.

Совершенно аналогично индукцией по $l = |h''|$ можно показать, что пара (h'', g'') имеет правое ортогональное разложение (G, h_2, g_2).

Обратимся к равенству $[h'h''] = [g'g'']$. Учитывая левое ортогональное разложение пары (h', g') и правое ортогональное разложение пары (h'', g''), получаем $[Hh_1 h_2 G] = [Hg_1 g_2 G]$. В полугруппе $\mathcal{F}_c^I$ действуют законы левого и правого сокращения, поэтому $[h_1 h_2] = [g_1 g_2]$. Пары (h_1, g_1) и (h_2, g_2) являются I-ортогональными, поэтому из утверждения 7 следует, что цепочки h_1 и g_2 состоят из операторов одного и того же множества $\mathcal{A}'$, а цепочки h_2 и g_1 — из операторов одного и того же множества $\mathcal{A}''$. При этом $\mathcal{A}' \cap \mathcal{A}'' = \varnothing$ и $\mathcal{A}' \times \mathcal{A}'' \subseteq I$. Значит, цепочки g_1 и g_2 перестановочны между собой и выполняется равенство $[h_1 h_2] = [g_1 g_2] = [g_2 g_1]$. Тогда из утверждения 8 следует, что $[h_1] = [g_2]$ и $[h_2] = [g_1]$. Лемма 5 доказана.

Теорема 10. *Для всякого множества перестановочности I проблема эквивалентности программ на шкале $\mathcal{F}_c^I$ разрешима за время $O(n^3 \log n)$.*

Доказательство. Пусть E — подполугруппа в $\mathcal{F}_c^I \times \mathcal{F}_c^I$, указанная в определении критериальной системы. Рассмотрим полугруппу W, порожденную множеством элементов $\{w^+, w^*\} \cup \{\langle [a], [b] \rangle \colon a, b \in \mathcal{A}\}$ при помощи определяющих соотношений

$$\langle [g_1],[h_1] \rangle \circ \langle [g_2],[h_2] \rangle = \langle [g_1 g_2],[h_1 h_2] \rangle,\ w^+ \circ \langle [a],[a] \rangle = w^+,\ \langle [a],[a] \rangle \circ w^* = w^*,\ w^+ \circ w^* = e.$$

Покажем, что система $K = \langle W, E, w^+, w^* \rangle$ является 1-критериальной для шкалы $\mathcal{F}_c^I$. Как нетрудно заметить, определяющие соотношения на полугруппе W устроены так, что равенство $w^+ \circ \langle [g], [h] \rangle \circ w^* = e$ справедливо в том и только том случае, когда $[g] = [h]$. Поэтому система K обладает свойством C1, если в качестве гомоморфизма φ взять тождественное отображение полугруппы E на себя. Выполнимость условия C2 следует из леммы 5, утверждения 10 и определяющих соотношений на W.

5.4. Шкалы условной эквивалентности. Рассмотрим следующие три оператора:

$$a_1\colon\quad x := 2 * y, \qquad a_2\colon\quad z := 2 * x, \qquad a_3\colon\quad z := 4 * y.$$

Очевидно, что последовательное выполнение операторов a_1 и a_2 дает точно такой же эффект, что и последовательное выполнение операторов a_1 и a_3, т. е. для данных операторов выполнено тождество $[a_1 a_2] = [a_1 a_3]$. Обозначим через J некоторое множество упорядоченных троек операторов, $J \subseteq \mathcal{A} \times \mathcal{A} \times \mathcal{A}$. Будем говорить, что *операторы a и b условно эквивалентны относительно оператора c*, если $(a, b, c) \in J$. Рассмотрим шкалу условно эквивалентных операторов $\mathcal{F}^J$, соответствующую полугруппе операторов с множеством определяющих соотношений $\{ca = cb\colon (a, b, c) \in J\}$. Очевидно, что $\mathcal{F}^J$ — полугрупповая уравновешенная шкала. Отметим следующее свойство полугруппы условно эквивалентных операторов.

Утверждение 11. *Пусть g_1, g_2, h_1 и h_2 — такие операторные цепочки, что $|g_1| = |h_1|$ и $|g_2| = |h_2|$. Тогда $[g_1 g_2] = [h_1 h_2] \Longrightarrow [g_1] = [h_1]$.*

Доказательство. При $[g_1 g_2] = [h_1 h_2]$ существует конечная последовательность основных тождеств $t_1, \ldots, t_n$, определяющих полугруппу $\mathcal{F}^J$, применение которых преобразует цепочку $g_1 g_2$ в $h_1 h_2$. Заметим, что в ходе этого преобразования каждое основное тождество t_i вида $ca = cb$, вносящее изменение в суффикс длины $|g_2|$, не изменяет префикс длины $|g_1|$. В свою очередь, возможность применения тождеств вида $ca = cb$ к префиксам длины $|g_1|$ не зависит от последних $|g_2|$ операторов цепочки и не изменяет их. Таким образом, последовательность применений тождеств $t_1, \ldots, t_n$ однозначно разбивается на подпоследовательности $t_{i_1}, \ldots, t_{i_k}$ и $t_{j_1}, \ldots, t_{j_m}$, причем первая состоит в точности из всех тех применений тождеств, которые изменяют префиксы длины $|g_1|$, а вторая — из применений тождеств, вносящих изменения в суффиксы длины $|g_2|$. Тогда последовательное применение тождеств $t_{i_1}, \ldots, t_{i_m}$ преобразует g_1 в h_1. Следовательно, $[g_1]=[h_1]$.

Следствие 4. *В полугруппе $\mathcal{F}^J$ действует закон правого сокращения, но не выполняется закон левого сокращения.*

Это означает, что для шкалы $\mathcal{F}^J$ не выполняются условия теорем Летичевского [4, 24, 26] о разрешимости эквивалентности пропозициональных программ на полугрупповых шкалах. Тем не менее можно, применяя теорему 1, построить полиномиальный алгоритм, разрешающий эквивалентность программ на $\mathcal{F}^J$. Чтобы сформировать критериальную систему для данной шкалы, рассмотрим вспомогательный помеченный граф Γ_J следующего вида. Вершинами Γ_J служат операторы из $\mathcal{A}$. Ребра графа также помечены операторами. Каждой тройке (a, b, c) из J ставится в соответствие ребро графа Γ_J, соединяющее вершины a и b и помеченное оператором c. Пусть A — некоторое множество операторов. Обозначим через $\Gamma_J(A)$ подграф, полученный из Γ_J путем удаления всех ребер, пометки которых не содержатся в A. Для заданного множества операторов A и некоторого выделенного оператора a обозначим через $V_J(A, a)$ множество вершин-операторов Γ_J, содержащихся в той же компоненте связности подграфа $\Gamma_J(A)$, что и вершина-оператор a. Иными словами, $V_J(A, a)$ — множество всех операторов, достижимых в графе Γ_J из вершины a по ребрам, помеченным операторами из A. Очевидно, что для любых операторов a и b и множества операторов A выполнено либо $V_J(A, a)=V_J(A, b)$, либо $V_J(A, a)\cap V_J(A, b)=\varnothing$.

Утверждение 12. *Пусть A — множество всех операторов, которыми оканчиваются операторные цепочки, условно эквивалентные заданной цепочке g. Тогда для любых операторов a и b имеет место эквивалентность $[ga]=[gb] \Longleftrightarrow V_J(A, a)=V_J(A, b)$.*

Доказательство. Согласно определению полугруппы $\mathcal{F}^J$ операторные цепочки ga и gb условно эквивалентны в том и только том случае, когда существует конечная последовательность определяющих полугруппу $\mathcal{F}^J$ тождеств $t_1, t_2, \ldots, t_n$, применение которых преобразует ga в gb. Выделим из нее подпоследовательность $t_{j_1}, t_{j_2}, \ldots, t_{j_m}$ тождеств вида $ca=cb$, применение которых изменяет последний оператор в цепочках. Таким образом, получаем последовательность операторных цепочек

$$g_1c_1a_1,\ g_1c_1a_2,\ \ldots,\ g_ic_ia_{2i-1},\ g_ic_ia_{2i},\ \ldots,\ g_mc_ma_{2m-1},\ g_mc_ma_{2m},$$

в которой $a=a_1$, $b=a_{2m}$ и выполнены соотношения

$$[g]=[g_ic_i], \qquad a_{2i}=a_{2i+1}, \quad (a_{2i-1}, a_{2i}, c_i)\in J, \qquad i=1, \ldots, m.$$

Включение $c_i \in A$ справедливо при всех i, $1 \leqslant i \leqslant m$, поэтому можно заметить, обратившись к определению Γ_J, что существование такой последовательности равносильно наличию в подграфе $\Gamma_J(A)$ маршрута из вершины, помеченной оператором a, в вершину, помеченную b. Последнее, очевидно, возможно в том и только том случае, когда $V_J(A, a)=V_J(A, b)$. Утверждение доказано.

Для шкалы условно эквивалентных операторов сформируем критериальную систему $K=\langle W, E, w^+, w^*\rangle$, где E — подполугруппа моноида $\mathcal{F}^J \times \mathcal{F}^J$, порожденная множеством пар $\{\langle [a], [b]\rangle\colon a, b\in \mathcal{A}\}$, а полугруппа W образуется за счет обогащения E дополнительным множеством образующих $\{w_A\colon A\in 2^{\mathcal{A}}\}\cup\{w^+, w^*\}$.

Определяющие соотношения для полугруппы W таковы:

$\langle[g_1],[h_1]\rangle \circ \langle[g_2],[h_2]\rangle = \langle[g_1g_2],[h_1h_2]\rangle$,
$w^+ \circ \langle[a],[a]\rangle = w_{\{a\}}$ для любого $a \in \mathscr{A}$,
$w^+ \circ \langle[a],[b]\rangle = w_{\varnothing}$ для любых $a, b \in \mathscr{A}$, $a \neq b$,
$w^+ \circ w^* = e$,
$w_A \circ \langle[a],[b]\rangle = w_B$ для любых $a, b \in A$, где $A \neq \varnothing$, $B = V_J(A, a) \cap V_J(A, b)$,
$w_A \circ w^* = e$ для любого $A \neq \varnothing$,
$w_{\varnothing} \circ w = w_{\varnothing}$ для любого элемента $w \in W$.

Лемма 6. *Пусть g и h — равновеликие упорядоченные о. ц. Тогда $[g]=[h]$ в том и только том случае, когда $w^+ \circ \langle[g],[h]\rangle = w_A$, где A — множество всех операторов, которыми оканчиваются цепочки, условно эквивалентные g.*

Доказательство проводится индукцией по длине цепочки g с использованием утверждения 12 и определяющих соотношений для W.

Теорема 11. *Для всякого множества операторных троек J проблема эквивалентности программ на шкале $\mathscr{F}^J$ разрешима за время $O(n^2 \log n)$.*

Доказательство. Убедимся, что построенная система K является $2^{|\mathscr{A}|}$-критериальной системой для шкалы $\mathscr{F}^J$. Легко видеть, что определяющие соотношения для W устроены так, что для любых равновеликих операторных цепочек g и h существует единственное множество операторов A, $A \in \mathscr{A}$, для которого $w^+ \circ \langle[g],[h]\rangle = w_A$. Поэтому $w^+ \circ U \subseteq \{w_A : A \in 2^{\mathscr{A}}\} \cup \{w^+\}$, причем решением уравнения $w \circ w^* = e$ могут быть только элементы вида w_A, где $A \neq \varnothing$, а также выделенный элемент w^+. Значит, для K выполнено свойство С2 критериальной системы при $k_0 = 2^{|\mathscr{A}|}$. Чтобы проверить свойство С1, достаточно обратиться к лемме 6 и заметить, что для любого непустого множества операторов A выполнено неравенство $w_A \neq w_{\varnothing}$. Теорема доказана.

Предложенную методику с некоторыми изменениями различной степени сложности можно распространить для исследования эквивалентности операторных программ на упорядоченных полугрупповых шкалах, а также для анализа поведения рекурсивных программ на уравновешенных и упорядоченных шкалах. В частности, таким образом удалось построить полиномиальный алгоритм распознавания эквивалентности монадических линейных рекурсивных программ на универсальных и коммутативных шкалах [15].

Пользуясь случаем, автор выражает признательность участникам научно-исследовательского семинара «Проблемы теоретического программирования», проводимого на факультете ВМиК МГУ, за внимание, проявленное к настоящей работе, и полезные замечания, высказанные при ее обсуждении.

СПИСОК ЛИТЕРАТУРЫ

1. Ахо А., Ульман Д. Теория синтаксического анализа, перевода и компиляции. — М.: Мир, 1978.
2. Бульонков М. А., Кочетов Д. В. Схема эффективной специализации императивных программ // Программирование. — 1995. — № 5. — С. 33–45.
3. Глушков В. М. Теория автоматов и вопросы проектирования структур цифровых машин // Кибернетика. — 1965. — № 1. — С. 1–9.
4. Глушков В. М., Летичевский А. А. Теория дискретных преобразователей // Избранные вопросы алгебры и логики. — Новосибирск: Наука, 1973. — С. 5–39.
5. Годлевский А. Б. Об одном случае специальной проблемы функциональной эквивалентности дискретных преобразователей // Кибернетика. — 1974. — № 3. — С. 32–36.

6. Ершов А. П. Об операторных схемах Янова // Проблемы кибернетики. Вып. 20. — М.: Наука, 1968. — С. 181–200.
7. Ершов А. П. Современное состояние теории схем программ // Проблемы кибернетики. Вып. 27. — М.: Наука, 1973 — С. 87–110.
8. Ершов А. П. Введение в теоретическое программирование. — М.: Наука, 1977.
9. Захаров В. А. Автоматные модели программ // Докл. АН СССР. — 1989. — Т. 309, № 1. — С. 24–27.
10. Захаров В. А. Формальные модели и свободные схемы программ // Программирование. — 1992. — № 2. — С. 10–24.
11. Захаров В. А. Об одном критерии сравнения операторных формальных моделей // Программирование. — 1993. — № 4. — С. 12–25.
12. Захаров В. А. Об отношении аппроксимируемости семантик операторных программ // Вестник МГУ. Вычислительная математика и кибернетика. — 1994. — № 3. — С. 54–60.
13. Захаров В. А. О свободных схемах в формальных моделях программ // Математические вопросы кибернетики. Вып. 5. — М.: Наука, 1994. — С. 208–239.
14. Захаров В. А., Подловченко Р. И. Полиномиальный алгоритм разрешения эквивалентности схем программ // Тезисы докладов XI Междунар. конф. «Проблемы теоретической кибернетики». — М., 1996. — С. 68–69.
15. 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.
16. Иткин В. Э. Логико-термальная эквивалентность схем программ // Кибернетика. — 1972. — № 1. — С. 5–27.
17. Касьянов В. П. Смешанные вычисления и оптимизация программ // Кибернетика. — 1980. — № 2. — С. 51–54.
18. Касьянов В. П. Оптимизирующие преобразования программ. — М.: Наука, 1988.
19. Кинбер Е. Б. Об одном классе многоленточных автоматов с разрешимой проблемой эквивалентности // Программирование. — 1983. — № 3. — С. 3–16.
20. Котов В. Е., Сабельфельд В. К. Теория схем программ. — М.: Наука, 1991.
21. Летичевский А. А. Эквивалентность автоматов с заключительным состоянием относительно свободной полугруппы с правым нулем // Докл. АН СССР. — 1968. — Т. 182, № 5. — С. 248–252.
22. Летичевский А. А. Функциональная эквивалентность дискретных преобразователей. I // Кибернетика. — 1969. — № 2. — С. 5–15.
23. Летичевский А. А. Функциональная эквивалентность дискретных преобразователей. II // Кибернетика. — 1970. — № 2. — С. 14–28.
24. Летичевский А. А. Эквивалентность автоматов относительно полугрупп // Теоретическая кибернетика. Вып. 6. — Киев, 1970. — С. 3–71.
25. Летичевский А. А. Функциональная эквивалентность дискретных преобразователей. III // Кибернетика. — 1972. — № 1. — С. 1–4.
26. Летичевский А. А. Эквивалентность автоматов относительно полугрупп с сокращением // Проблемы кибернетики. Вып. 27. — М.: Наука, 1973. — С. 195–212.
27. Летичевский А. А. Практические методы распознавания эквивалентности дискретных преобразователей и схем программ // Кибернетика. — 1973. — № 4. — С. 15–26.
28. Летичевский А. А., Смикун Л. Б. О группах с разрешимой проблемой эквивалентности // Тезисы 4-й Всесоюзной конференции по математической логике. — М., 1976. — С. 77.
29. Лисовик Л. П. О разрешимых проблемах для металинейных схем // Докл. АН УССР. — 1979. — № 2. — С. 130–133.
30. Лисовик Л. П. Металинейные рекурсивные схемы над размеченными деревьями // Программирование. — 1983. — № 5. — С. 13–22.
31. Лисовик Л. П. Металинейные схемы с засылками констант // Программирование. — 1985. — № 2. — С. 29–38.
32. Лисовик Л. П. Стандартные схемы с магазинами // Докл. АН УССР. — 1989. — № 12. — С. 23–27.
33. Лисовик Л. П. Схемы программ и преобразователи над размеченными деревьями // Математические вопросы кибернетики. Вып. 6. — М.: Наука, 1996. — С. 281–320.
34. Ляпунов А. А. О логических схемах программ // Проблемы кибернетики. Вып. 1. — М.: Физматгиз, 1958. — С. 46–74.
35. Ляпунов А. А., Янов Ю. И. О логических схемах программ // Труды конференции «Пути развития советского математического машиностроения и приборостроения». Ч. 3. — М., 1956. — С. 5–8.
36. Мейтус В. Ю. Проблема эквивалентности строгих детерминированных автоматов, действующих в реальном времени // Кибернетика. — 1989. — № 5. — С. 14–25.
37. Непомнящий В. А. Об одном методе распознавания эквивалентности схем программ и дискретных преобразователей // Труды всесоюзной конференции по программированию (ВПК-2). Вып. К. — Новосибирск: ВЦ СО АН СССР. — 1970. — С. 49–63.

38. Непомнящий В. А. О проблеме пустоты для схем программ. I, II // Программирование. — 1976. — № 4. — С. 43–51; 1977. — № 4. — С. 3–13.
39. Непомнящий В. А., Рякин О. М. Прикладные методы верификации программ. — М.: Радио и связь, 1988.
40. Непомнящий В. А., Сабельфельд В. К. Трансформационный синтез корректных программ // Прикладная информатика. — 1986. — Т. 2. — С. 19–38.
41. Петросян Г. Н. Об одном базисе операторов и предикатов с неразрешимой проблемой пустоты // Кибернетика. — 1974. — № 5. — С. 23–28.
42. Петросян Г. Н. Проблема включения на подпамяти для операторных схем и случай ее разрешения // Системное и теоретическое программирование. — Новосибирск: ВЦ СО АН СССР. — 1974. — С. 130–151.
43. Подловченко Р. И. Иерархия моделей программ // Программирование. — 1981. — № 2. — С. 3–14.
44. Подловченко Р. И. Полугрупповые модели программ // Программирование. — 1981. — № 4. — С. 3–13.
45. Подловченко Р. И. Моделирование программ схемами и построение систем преобразований схем // Кибернетика. — 1982. — № 6.
46. Подловченко Р. И. О проблеме эквивалентных преобразований программ // Программирование. — 1986. — № 6. — С. 3–14.
47. Подловченко Р. И. Проблема эквивалентности в коммутативных s-моделях // Программирование. — 1987. — № 5. — С. 6–17.
48. Подловченко Р. И. Схемы программ с монотонными операторами // Программирование. — 1988. — № 6. — С. 3–12.
49. Подловченко Р. И. Разрешимость эквивалентности в множестве схем программ с монотонными и частично перестановочными операторами // Программирование. — 1990. — № 5. — С. 3–12.
50. Подловченко Р. И. Рекурсивные программы и иерархия их моделей // Программирование. — 1991. — № 6. — С. 44–51.
51. Подловченко Р. И. Разрешимость проблемы эквивалентных преобразований в специальных автоматных моделях рекурсивных программ // Докл. РАН. — 1994. — Т. 337, № 5. — С. 577–580.
52. Подловченко Р. И., Захаров В. А. Быстрые алгоритмы распознавания эквивалентности в моделях операторных программ с коммутирующими операторами // Сборник «Компьютерные аспекты в научных исследованиях и учебном процессе». — М.: Изд-во МГУ, 1996. — С. 3–8.
53. Подловченко Р. И., Захаров В. А. Полиномиальный по сложности алгоритм, распознающий коммутативную эквивалентность схем программ // Докл. РАН. Информатика. — 1998. — Т. 362, № 6.
54. Романовский В. Ю. Разрешимость эквивалентности линейных унарных рекурсивных схем с индивидуальными константами // Докл. АН УССР. — 1980. — № 1. — С. 72–75.
55. Романовский В. Ю. Проблема эквивалентности детерминированных МП-автоматов, действующих в реальном времени // Кибернетика. — 1986. — № 2. — С. 13–23.
56. Сабельфельд В. К. Полиномиальная оценка сложности распознавания логико-термальной эквивалентности // Докл. АН СССР. — 1979. — Т. 249, № 4. — С. 793–796.
57. Сабельфельд В. К. О проблеме древесной пустоты для рекурсивных схем программ // Проблемы системного и теоретического программирования. — Новосибирск, 1984. — С. 157–166.
58. Сабельфельд В. К. Новый класс схем с разрешимой функциональной эквивалентностью // Информатика: инструментальные средства. — Новосибирск, 1988. — С. 109–126.
59. Тайцлин М. А. Эквивалентность автоматов относительно коммутативной полугруппы // Алгебра и логика. — 1968. — Т. 8, № 5. — С. 553–600.
60. Янов Ю. И. О равносильности и преобразованиях схем программ // Докл. АН СССР. — 1957. — Т. 113, № 1. — С. 39–42.
61. Янов Ю. И. О логических схемах алгоритмов // Проблемы кибернетики. Вып. 1. — М.: Физматгиз, 1958. — С. 75–127.
62. Янов Ю. И. О вычислениях в одном классе программ // Проблемы кибернетики. Вып. 32. — М.: Наука, 1977. — С. 237–245.
63. Ashcroft E., Manna Z., Pnueli A. A decidable properties of monadic functional schemes // Journal of the ACM. — 1973. — V. 20, № 3. — P. 489–499.
64. De Bekker J. W., Skott D. A. Theory of programs. Unpublished notes // Vienna: IBM Seminar. — 1969.
65. Bird R. The equivalence problem for deterministic two-tape automata // Journal of Computer and System Science. — 1973. — V. 7, № 4. — P. 218–236.
66. Culik K. II New techniques for proving the decidability of equivalence problems // Lecture Notes in Computer Science. — 1988. — V. 317. — P. 162–175.
67. Culik K. II, Karhumaki J. Synchronizable deterministic pushdown automata and the decidability of their equivalence // Acta Informatica. — 1986. — V. 23, № 5. — P. 597–605.

68. Culik K. II, Linna M. The equivalence problem for n-tape finite automata with simple cycles // Lecture Notes in Computer Science. — 1987. — V. 287. — P. 15–25.
69. Fortune S., Hopcroft J. E., Schmidt E. Complexity of equivalence and containment for free single variable program schemes // Cornell University, Computer Science Department / Technical Report TR 77-310. — 1977. — P. 14.
70. Garey M. R., Johnson D. S. A guide to the theory of NP-completeness. — San-Francisco: W. H. Freeman & Co. Publ., 1979.
71. Garlend 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.
72. Gurari E. M. Decidable problems for the reinforced programs // Journal of the ACM. — 1985. — V. 32, № 2. — P. 466–483.
73. Gurari E. M., Ibarra O. H. The complexity of equivalence problem for simple programs // Journal of the ACM. — 1981. — V. 28, № 3. — P. 535–560.
74. Harel D. Dynamic logics // Handbook of Philosophical Logics. — Elsevier Sci. Publ., 1984. — P. 497–604.
75. Harel D., Sherman R. Propositional dynamic logic of flowcharts // Lecture Notes in Computer Science. — 1982. — V. 158. — P. 195–206.
76. Harju T., Karhumaki J. The equivalence of multi-tape finite automata // Theoretical Computer Science. — 1991. — V. 78, № 2. — P. 347–355.
77. Hopcroft J. E., Karp R. M. A linear algorithm for testing equivalence of finite automata // Cornell University, Computer Science Department / Technical Report TR 71-114. — 1971. — P. 5.
78. Ibarra O. H. Reversal-bounded multicounter machines and their decision problems // Journal of the ACM. — 1978. — V. 25, № 1. — P. 116–133.
79. Itkin V. E., Zwinogrodski Z. On program schemata equivalence // Journal of Computer and System Science. — 1972. — V. 6, № 1. — P. 88–101.
80. 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.
81. Meyer A. R., Ritchi D. M. Computational complexity and program structure // IBM Techn. J. — May 1967. — Res. Papers RC-1817.
82. Paterson M. S. Program schemata // Machine Intelligence. — 1968. — V. 3. — P. 19–31.
83. Paterson M. S. Decision problems in computational models // SIGPLAN Notices. — 1972. — V. 7 — P. 74–82.
84. Paterson M. S., Hewitt C. T. Comparative schematology // Proc. of the ACM Conf. on Concurrent Systems and Parallel Computation. — New York, 1970. — P. 119–127.
85. Rabin M. O., Scott D. Finite automata and their decision problems // IBM Jornal of Research and Development. — 1959. — V. 3, № 2. — P. 114–125.
86. Rice H. G. Classes of recurcively enumerable sets and their decision problems // Transactions of American Mathematical Society. — 1953. — V. 74, № 2.
87. Rutledge J. D. On Ianov's program schemata // Journal of the ACM. — 1964. — V. 11, № 1. — P. 1–9.
88. Salomaa K. Decidability of equivalence for deterministic synchronized tree automata // Lecture Notes in Computer Science. — 1995. — V. 915. — P. 140–162.
89. Sabelfeld V. K. Tree equivalence of linear recursive schemata is polynomial-time decidable // Information Processing Letters. — 1981. — V. 13, № 4. — P. 147–153.
90. Sabelfeld V. K. An algorithm deciding functional equivalence in a new class of program schemes // Theoretical Computer Science. — 1990. — V. 71, № 2. — P. 265–279.
91. Senizergues G. The equivalence problem for deterministic pushdown automata is decidable // Lecture Notes in Computer Science. — 1997. — V. 1256. — P. 671–681.
92. Strong H. R. Translating recursive equations into flow-charts // Journal of Computer and System Science. — 1971. — V. 5, № 3. — P. 254–285.
93. Tomita E., Seino K. A weaker condition for the equivalence of a pair of DPDAs to be decidable // Theoretical Computer Science. — 1985. — V. 46, № 2–3. — P. 175–195.
94. Tomita E., Seino K. A direct branching algorithm for checking the equivalence of two deterministic pushdown transducers one of which is real time strict // Theoretical Computer Science. — 1989. — V. 64, № 1. — P. 39–53.
95. 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.
96. Tsichritzis D. The equivalence problem of a simple programs // Journal of the ACM. — 1970. — V. 17, № 4.
97. Valiant L. G. Decision procedures for families of deterministic pushdown automata // Univ. of Warwick Computer Center. Rep. № 7. — 1973.
98. Valiant L. G. The equivalence problem for deterministic finite-turn pushdown automata // Information and Control. — 1974. — V. 25, № 2. — P. 123–133.
99. Valiant L. G., Paterson M. S. Deterministic one-counter automata // Journal of Computer and System Science. — 1975. — V. 10. — P. 340–350.

Поступило в редакцию 10 III 1998