УДК 519.71

В.А. ЗАХАРОВ

ПРОВЕРКА ЭКВИВАЛЕНТНОСТИ ПРОГРАММ С ПОМОЩЬЮ ДВУХЛЕНТОЧНЫХ АВТОМАТОВ

Ключевые слова: *программа, эквивалентность, алгоритмическая разрешимость, многоленточный автомат, вычислительная сложность.*

ВВЕДЕНИЕ

Проблема эквивалентности программ состоит в том, чтобы для произвольной пары программ выяснить, имеют ли они одинаковое поведение. Поведение программ может определяться в зависимости от исследуемого класса программ. В частности, поведение вычислительной программы может быть охарактеризовано функцией преобразования входных данных программы в выходные данные. В этом случае программы считаются эквивалентными, если они вычисляют одну и ту же функцию. Проблема функциональной эквивалентности программ возникает во многих задачах системного и прикладного программирования. К сожалению, в универсальных системах программирования эта проблема алгоритмически неразрешима. Поэтому для получения эффективно проверяемых достаточных условий функциональной эквивалентности программ приходится обращаться к абстрактным моделям программ. Интерес представляют такие модели программ, в которых разрешимое отношение эквивалентности программ аппроксимирует отношение функциональной эквивалентности.

Начиная с основополагающих работ [1, 2], был предложен широкий спектр формальных моделей программ, ориентированных на изучение проблемы функциональной эквивалентности, — стандартные схемы программ [2, 3], дискретные преобразователи [5, 6], схемы рекурсивных программ [7], алгебраические модели программ [8] и др. На основе этих моделей в 60–70-е годы прошлого века была предпринята попытка разработать математические методы оптимизации компьютерных программ и микропроцессоров [9]. На первом этапе исследований планировалось выделить модели программ с разрешимой проблемой эквивалентности. Изучение этой задачи проводилось достаточно интенсивно. Было установлено, что во всех указанных моделях программ проблема эквивалентности в общем случае является алгоритмически неразрешимой [3, 8, 10, 11]. Были также выделены многочисленные и обширные классы программ, в которых проблема эквивалентности оказалась разрешимой (например, [6, 12–24]). Подробный обзор и анализ большинства указанных результатов можно найти в монографии [25].

Для решения проблемы эквивалентности программ был предложен ряд оригинальных подходов: метод сетевой разметки для стандартных схем [21, 22], метод устранения несущественных ветвлений [4, 14], метод жестких множеств [26], метод параллельного стекинга [23, 27]. Кроме того, было установлено, что разрешающие процедуры, созданные на основе этих методов, имеют чрезвычайно высокую вычислительную сложность и непригодны для практического использования. По мнению автора настоящей статьи, одним из возможных объяснений этому явлению могло быть желание исследователей добиться как можно большей универсальности предложенных ими подходов к проверке эквивалентности программ. По мере того, как расширялся класс программ, к которым этот метод мог быть применен, и совершенствовалась разрешающая процедура, она становилась все менее чувствительной к специфическим особенностям анализируемых программ. В результате одни и те же средства применялись как для сложных, так и простых случаев при разрешении проблемы эквивалентности программ, и это зачастую приводило к неоправданно большим вычислительным затратам. Необходимость в создании практичных и быстрых (полиномиальных по времени) алгоритмов проверки эквивалентности программ, учитывающих в большей мере индивидуальные алгебраические свойства базовых компонентов программы, отмечалась в ряде работ [9, 28].

Существенным продвижением в построении быстрых алгоритмов проверки эквивалентности программ стала статья [29], в которой доказана разрешимость за полиномиальное время проблемы эквивалентности программ с коммутативными операторами (существование разрешающего алгоритма в этой модели программ было установлено ранее в статьях [6, 13]). Эти результаты привели к разработке нового метода [30–32], позволяющего создавать полиномиальные по времени алгоритмы проверки эквивалентности последовательных программ, семантика операторов которых удовлетворяет определенным алгебраическим соотношениям. Предложенный метод был распространен и на другие модели программ [33–37]. Кроме того, в статьях [38, 39] с помощью метода следов удалось построить разрешающие алгоритмы (в том числе полиномиальной сложности) в одном классе алгебраических моделей программ, обладающих следующей характерной особенностью: любые две конечные эквивалентные последовательности программных операторов имеют одинаковую длину. Были обнаружены также простые модели программ, в которых задача проверки эквивалентности программ (в конечном алфавите операторов и предикатов) является PSPACE-полной [40].

Полученные результаты послужили предпосылкой для более глубокого исследования сложностных аспектов проблемы эквивалентности программ. В результате был предложен новый способ описания семантик пропозициональных моделей последовательных программ с использованием двухленточных автоматов и на его основе создан общий метод решения проблемы эквивалентности программ путем ее сведения к задаче проверки пустоты для двухленточных автоматов, а также установлены достаточные условия разрешимости этой задачи за полиномиальное время.

В настоящей статье показано, каким образом двухленточные автоматы можно применять для описания семантик некоторых пропозициональных моделей последовательных программ, а также для проектирования процедур проверки эквивалентности программ в этих моделях.

ПРОПОЗИЦИОНАЛЬНАЯ МОДЕЛЬ ПРОГРАММ

Определим синтаксис и семантику рассматриваемого класса программ. Пусть заданы два конечных алфавита: $A=\{a_1,\dots,a_r\}$ и $P=\{p_1,\dots,p_k\}$. Элементы множества алфавита A называются операторами. Они соответствуют программным командам. Конечную последовательность операторов будем называть A-цепочкой. Множество всех A-цепочек обозначим A^*. Пустую A-цепочку обозначим λ; запись h_1h_2 будет обозначать конкатенацию A-цепочек h_1 и h_2. Элементами множества P являются предикаты. В любом состоянии данных каждый предикат может принимать одно из двух значений: 0 или 1. Двоичные наборы $\langle\delta_1,\dots,\delta_k\rangle$ значений предикатов назовем условиями. Множество всех условий обозначим C; условия из C будем обозначать Δ_1, Δ_2, ...

Программа представляется конечной размеченной системой переходов $\pi=\langle V, entry, exit, T, B\rangle$, в которой V — непустое конечное множество точек программы, $entry$ — точка входа, $exit$ — точка выхода, $T:(V\setminus\{exit\})\times C\to V$ — функция переходов, $B:(V\setminus\{exit\})\to A$ — функция привязки. Функция переходов задает поток управления программы, а функция привязки ассоциирует оператор с каждой точкой программы, отличной от точки выхода. Размер $|\pi|$ программы π — это количество точек программы.

Трассой в программе π называется любая последовательность пар $tr=(v_1,\Delta_1)$, (v_2,Δ_2), ... , удовлетворяющая для каждого $i, i\ge 1$, соотношениям $v_i\in V$, $\Delta_i\in C$ и $v_{i+1}=T(v_i,\Delta_i)$. Если $v_1=entry$, то трасса tr называется начальной. Если начальная трасса бесконечна или завершается такой парой (v_n,Δ_n), что $T(v_n,\Delta_n)=exit$, то трасса называется полной. Для конечной трассы tr запись tr^i будет обозначать префикс tr длины i, а $B(tr)$ будет обозначать A-цепочку $B(v_1), B(v_2), \dots, B(v_n)$ операторов, приписанных точкам этой трассы. Точка v программы π называется мертвой, если она не содержится ни в одной полной трассе этой программы.

Семантика программ определяется на основе моделей Крипке, используемых в пропозициональных динамических логиках программ [41]. Динамической шкалой называется тройка $F = \langle S, s_0, R \rangle$, в которой S — непустое множество состояний данных, s_0 — начальное состояние ($s_0 \in S$) и $R: S \times A \to S$ — интерпретация операторов, вычисляющая то состояние данных $R(s, a)$, в которое оператор a преобразует состояние данных s. Функцию R можно распространить на множество операторных цепочек, введя функцию R^*, удовлетворяющую соотношениям $R^*(s, \lambda) = s$, $R^*(s, ha) = R(R^*(s, h), a)$. Состояние данных s'' называется достижимым из состояния s' (обозначается $s' \prec_F s''$), если $s'' = R(s', h)$ для некоторой цепочки $h \in A^*$. Если отношение достижимости $\prec_F$ является отношением частичного порядка на S, то шкала F называется упорядоченной. Запись $[h]_F$ используется для обозначения состояния данных $R^*(s_0, h)$.

Динамическая модель Крипке — это пара $M = (F, \xi)$, где F — шкала Крипке, а $\xi: S \to C$ — оценка (истинности предикатов). Модель Крипке вида (F, ξ) будем называть F-моделью. Вычислением программы $\pi = \langle V, entry, exit, T, B \rangle$ на динамической модели Крипке $M = (F, \xi)$ называется максимальная последовательность пар

$$comp(\pi, M) = (v_0, s_0), (v_1, s_1), \ \ldots \ , (v_i, s_i), \ (v_{i+1}, s_{i+1}), \ \ldots,$$

удовлетворяющая следующим условиям:

1) s_0 — начальное состояние шкалы F и $v_0 = entry$;

2) $v_{i+1} = T(v_i, \xi(s_i))$ и $s_{i+1} = R(s_i, B(v_i))$ для всех i, $i \geq 0$.

Если вычисление является бесконечным, то считается, что оно безрезультатно зацикливается. В противном случае последовательность $comp(\pi, M)$ оканчивается парой $(exit, s_n)$, и тогда вычисление завершается с результатом s_n. Запись $[comp(\pi, M)]$ будет обозначать результат этого вычисления. Если вычисление зацикливается, то его результат считается неопределенным. Очевидно, что каждое вычисление $comp(\pi, M)$ на модели M прокладывает в программе π полную трассу

$$tr(\pi, M) = (v_0, \xi(s_0)), (v_1, \xi(s_1)), \ldots, (v_i, \xi(s_i)), \ \ldots$$

Если вычисление $comp(\pi, M)$ завершается, то $[comp(\pi, M)] = [B(tr(\pi, M))]$. Программы π' и π'' называются эквивалентными на шкале F ($\pi' \approx_F \pi''$), если равенство $[comp(\pi', M)] = [comp(\pi'', M)]$ выполняется для любой F-модели M. Проблема эквивалентности программ на динамической шкале F состоит в том, чтобы для произвольной пары программ π', π'' проверить выполнимость отношения $\pi' \approx_F \pi''$.

ДВУХЛЕНТОЧНЫЕ МАШИНЫ

Для исследования алгоритмической разрешимости и сложности проблемы эквивалентности программ необходимо рассмотреть эффективный способ описания динамических шкал. Каждая шкала F индуцирует отношение эквивалентности на множестве A^*: цепочки h_1 и h_2 считаются эквивалентными на шкале F, если $[h_1]_F = [h_2]_F$. Известно [42], что отношения на множестве конечных слов можно задавать многоленточными автоматами подобно тому, как языки задаются с помощью одноленточных автоматов. В частности, некоторые отношения эквивалентности на множестве конечных слов можно задавать с помощью двухленточных автоматов. Поэтому для описания некоторых классов динамических шкал целесообразно воспользоваться двухленточными детерминированными автоматами (машинами).

Двухленточной детерминированной машиной (2-DM) называется система $D = \langle \Sigma, Q_1, Q_2, q_0, Q_{accept}, \varphi \rangle$, состоящая из входного алфавита Σ, двух непересекающихся множеств внутренних состояний: Q_1 и Q_2, начального состояния q_0, $q_0 \in Q_1 \cup Q_2$, множества допускающих состояний Q_{accept}, $Q_{accept} \subseteq Q_1 \cup Q_2$, и функции переходов $\varphi: (Q_1 \cup Q_2) \times \Sigma \to Q_1 \cup Q_2$. Машина D прочитывает пару слов w_1 и w_2, записанных на лентах 1 и 2. Когда машина D пребывает во внутреннем состоянии q, $q \in Q_\sigma$, $\sigma \in \{1, 2\}$, она прочитывает очередную букву x (если таковая есть) слова w_σ, размещенного на ленте σ, и переходит в состояние $q' = \varphi(q, x)$.

Пара слов $\langle w_1, w_2 \rangle$ допускается машиной D, если она прочитывает оба слова w_1 и w_2 и оказывается по прочтении этих слов в допускающем состоянии. Более формально поведение машины D определяется следующим образом.

Прогоном 2-DM D называется конечная или бесконечная последовательность пар $\alpha = (q_1, x_1), (q_2, x_2), \ldots, (q_i, x_i), \ \ldots \ , x_i \in \Sigma$, удовлетворяющая соотношениям $q_i \in Q_1 \cup Q_2$ и $q_{i+1} = \varphi(q_i, x_i)$ для всех i, $i \geq 1$. Если эта последовательность оканчивается парой (q_n, x_n), то считают, что прогон α начинается в состоянии q_1 и достигает состояния $q_{n+1} = \varphi(q_n, x)$. Для $\sigma \in \{1, 2\}$ рассмотрим подпоследовательность $\alpha_\sigma = (q_{i_1}, x_{i_1}), (q_{i_2}, x_{i_2}), \ldots$, состоящую из всех тех пар (q_{i_j}, x_{i_j}) последовательности α, которые удовлетворяют условию $q_{i_j} \in Q_\sigma$. Тогда σ-проекцией прогона α будем называть слово $\alpha[\sigma] = x_{i_1} x_{i_2} \ldots$ Функцию переходов 2-DM D можно распространить на множество пар слов $\Sigma^* \times \Sigma^*$ следующим образом: $\varphi^*(q, w_1, w_2) = q'$, если существует прогон α машины D, который начинается в состоянии q, достигает состояния q', при этом $\alpha[1] = w_1$, $\alpha[2] = w_2$. В противном случае значение $\varphi^*(q, w_1, w_2)$ не определено. Каждая 2-DM $D = \langle \Sigma, Q_1, Q_2, q_0, Q_{accept}, \varphi \rangle$ порождает бинарное отношение $E_D = \{\langle w_1, w_2 \rangle : \varphi^*(q_0, w_1, w_2) \in Q_{accept}\}$. Будем говорить, что 2-DM D описывает шкалу F, если для любой пары A^*-цепочек h_1 и h_2 имеет место соотношение $[h_1]_F = [h_2]_F \Leftrightarrow \langle h_1, h_2 \rangle \in E_D$.

Теорема 1. Динамическая шкала F может быть описана некоторой 2-DM тогда и только тогда, когда F упорядочена.

Доказательство. ($\Rightarrow$) Для любой шкалы F отношение достижимости $\prec_F$ рефлексивно и транзитивно. Если бы оно не было антисимметричным, то существовала бы такая пара A-цепочек h', h'', что $h'' \neq \lambda$, $[h' h'']_F = [h']_F$. Поскольку шкалу F описывает 2-DM D, то в этом случае существовала бы пара допускающих состояний $q' = \varphi^*(q_0, h', h')$ и $q'' = \varphi^*(q_0, h', h' h'')$. Но это означало бы, что $q' \in Q_2$ и, следовательно, значение $\varphi^*(q_1, h' h'', h')$ не определено вопреки тому, что $[h' h'']_F = [h']_F$.

($\Leftarrow$) Для упорядоченной шкалы $F = \langle S, s_0, R \rangle$ рассмотрим 2-DM $D = \langle A, Q_1, Q_2, q_0, Q_{accept}, \varphi \rangle$, где $Q_1 = \{(s', s''): s', s'' \in S, \ s' \prec_F s''\}$, $Q_2 = (S \times S) \setminus Q_1$, $q_0 = (s_0, s_0)$, $Q_{accept} = \{(s, s): s \in S\}$ и функция переходов определена соотношениями $\varphi((s', s''), a) = (R(s', a), s'')$, если $(s', s'') \in Q_1$, и $\varphi((s', s''), a) = (s', R(s'', a))$, если $(s', s'') \in Q_2$. Для любой пары A-цепочек h_1, h_2 либо есть такой префикс h'_1 цепочки h_1, что $\varphi^*(q_0, h'_1, h_2)$ определено, либо есть такой префикс h'_2 цепочки h_2, что $\varphi^*(q_0, h_1, h'_2)$ определено. Предположим, что $[h_1]_F = [h_2]_F$, $h_1 = h'_1 a_1 a_2 \ldots a_k$ и $\varphi^*(q_0, h'_1, h_2) = q'$. Отношение достижимости $\prec_F$ является частичным порядком, и поэтому $[h'_1 a_1 a_2 \ldots a_i]_F \prec_F [h_2]_F$ выполняется для всех i, $1 \leq i \leq k$. Применяя индукцию по i, можно убедиться, что $\varphi^*(q_0, h'_1 a_1 a_2 \ldots \ldots a_i, h_2) = ([h'_1 a_1 a_2 \ldots a_i]_F, [h_2]_F) \in Q_1$ для всех i, $1 \leq i \leq k$. Поэтому значение $\varphi^*(q_0, h_1, h_2)$ определено и равно $([h_1]_F, [h_2]_F)$. Как видно из описания D, достигнутое состояние $\varphi^*(q_0, h_1, h_2)$ является допускающим. Теорема доказана.

КОМБИНИРОВАННЫЕ ДВУХЛЕНТОЧНЫЕ МАШИНЫ

Цель настоящей статьи — создание общего теоретико-автоматного метода проверки эквивалентности программ, семантика операторов которых определяется упорядоченными динамическими шкалами. В рамках этого подхода программы можно рассматривать как конечные автоматы, порождающие трассы. Поэтому отношение эквивалентности программ $\pi' \approx_F \pi''$ целесообразно определить в терминах программных трасс.

Для заданной шкалы F начальная трасса $tr = (v_1, \Delta_1), (v_2, \Delta_2), \ldots$ в программе π называется F-непротиворечивой, если для любой пары i, j из равенства $\lfloor B(tr^i) \rfloor_F = \lfloor B(tr^j) \rfloor_F$ следует $\Delta_{i+1} = \Delta_{j+1}$. Очевидна справедливость следующих утверждений.

Утверждение 1. Трасса tr в программе π является F-непротиворечивой тогда и только тогда, когда она является префиксом трассы $tr(\pi, M)$ для некоторой F-модели M.

Утверждение 2. Шкала F является упорядоченной тогда и только тогда, когда каждая начальная трасса в любой программе является F-непротиворечивой.

Таким образом, F-непротиворечивые трассы — это в точности те программные трассы, которые прокладываются вычислениями на F-моделях.

F-непротиворечивые трассы $tr_1 = (v'_1, \Delta'_1), (v'_2, \Delta'_2), \dots$ и $tr_2 = (v''_1, \Delta''_1)$, (v''_2, Δ''_2), ... в программах π_1 и π_2 называются F-совместными, если для любой пары i, j из равенства $[B(tr_1^i)]_F = [B(tr_2^j)]_F$ следует $\Delta'_{i+1} = \Delta''_{j+1}$.

Утверждение 3. Трассы tr_1 и tr_2 в программах π_1 и π_2 соответственно являются F-совместными тогда и только тогда, когда они являются префиксами трасс $tr(\pi_1, M)$ и $tr(\pi_2, M)$ для некоторой F-модели M.

Из этих утверждений следует теорема.

Теорема 2. Программы π_1 и π_2 неэквивалентны на шкале F тогда и только тогда, когда в этих программах существует пара F-совместных полных трасс tr_1 и tr_2, удовлетворяющих одному из двух условий: либо в точности одна из этих трасс конечна, либо обе трассы конечны и при этом $[B_1(tr_1)]_F = [B_2(tr_2)]_F$.

Теоретико-автоматный подход предусматривает представление анализируемых программ и их спецификаций в виде конечных автоматов (см. [43, 44]). Благодаря этому верификация программы сводится к конструированию различных композиций автоматов и проверке их свойств. Двухленточные машины можно использовать не только для описания динамических шкал, но и для решения с их помощью проблемы эквивалентности программ. Для заданной пары программ π_1 и π_2, а также 2-DM D, описывающей упорядоченную шкалу F, строится комбинированная 2-DM $K(\pi_1, \pi_2, D)$, состоящая из трех взаимодействующих компонентов: π_1, π_2 и D. На вход этой машины подается пара конечных последовательностей вида $tr_\sigma = (v_1^\sigma, \Delta_1^\sigma), (v_2^\sigma, \Delta_2^\sigma), \dots, \sigma = 1, 2$. Каждая компонента π_σ проверяет, является ли tr_σ трассой в программе π_σ. Компоненте D отводится роль синхронизатора, позволяющего прочитывать до конца только пары F-совместных трасс tr_1 и tr_2. Комбинированная 2-DM $K(\pi_1, \pi_2, D)$ допускает только такие пары F-совместных трассы в программах π_1 и π_2, которые удовлетворяют требованиям теоремы 2. Таким образом, соотношение $\pi_1 \approx_F \pi_2$ будет выполняться в том и только том случае, когда комбинированная 2-DM $K(\pi_1, \pi_2, D)$ порождает пустое бинарное отношение.

Дадим строгое определение 2-DM $K(\pi_1, \pi_2, D)$. Пусть заданы программы $\pi_\sigma = \langle V_\sigma, entry, exit, T_\sigma, B_\sigma \rangle$, $\sigma = 1, 2$, и машина $D = \langle A, Q_1, Q_2, q_0, Q_{accept}, \varphi \rangle$, описывающая упорядоченную динамическую шкалу $F = \langle S, s_0, R \rangle$. Введем вспомогательный символ $\perp$, $\perp \notin C$. Тогда $K(\pi_1, \pi_2, D) = \langle \hat{\Sigma}, \hat{Q}_1, \hat{Q}_2, \hat{q}_0, \hat{Q}_{accept}, \hat{\varphi} \rangle$, где

- $\hat{\Sigma} = (V_1 \times C) \cup (V_2 \times C)$,
- $\hat{Q}_\sigma = V_1 \times V_2 \times Q_\sigma \times (C \cup \{\perp\})$, $\sigma = 1, 2$,
- $\hat{q}_0 = (entry, entry, q_0, \perp)$,
- $\hat{\varphi}$ — частичная функция переходов, значения $\hat{\varphi}(\langle v_1, v_2, q, z \rangle, (v, \Delta))$ которой в случае $q \in Q_1$ определяются следующими соотношениями:

$\hat{\varphi}(\langle v_1, v_2, q, \perp \rangle, (v, \Delta)) = \langle T_1(v_1, \Delta), v_2, \varphi(q, B_1(v_1)), \perp \rangle$, если $v = v_1$, $q \notin F$ и v_1 не является ни выходом *exit*, ни мертвой точкой;

$\hat{\varphi}(\langle v_1, v_2, q, \perp \rangle, (v, \Delta)) = \langle T_1(v_1, \Delta), v_2, \varphi(q, B_1(v_1)), \Delta \rangle$, если $v = v_1$, $q \in F$ и v_1 не является ни выходом *exit*, ни мертвой точкой;

$\hat{\varphi}(\langle v_1, v_2, q, \perp \rangle, (v, \Delta))$ не определено во всех остальных случаях;

$\hat{\varphi}(\langle v_1, v_2, q, \Delta' \rangle, (v, \Delta)) = \langle T_1(v_1, \Delta), v_2, \varphi(q, B_1(v_1)), \perp \rangle$, если $v = v_1$, $q \notin F$ и v_1 не является ни выходом *exit*, ни мертвой точкой и $\Delta = \Delta'$;

$\hat{\varphi}(\langle v_1, v_2, q, \Delta' \rangle, (v, \Delta)) = \langle T_1(v_1, \Delta), v_2, \varphi(q, B_1(v_1)), \perp \rangle$, если $v = v_1$, $q \in F$, v_1 не является ни выходом *exit*, ни мертвой точкой и $\Delta = \Delta'$;

$\hat{\varphi}(\langle v_1, v_2, q, \perp \rangle, (v, \Delta))$ не определено во всех остальных случаях.

При $q \in Q_2$ значение $\hat{\varphi}(\langle v_1, v_2, q, z\rangle, (v, \Delta))$ определяется аналогично.

Множество $\hat{Q}_{accept}$ состоит из всех четверок $\langle v_1, v_2, q, z\rangle$, удовлетворяющих одному из следующих требований:

1) $v_1 = exit$, $v_2 = exit$ и $q \notin Q_{accept}$;

2) $v_i = exit$, $v_{3-i} \neq exit$ и $q \in Q_i$, где $i = 1, 2$;

3) v_i является мертвой точкой, v_{3-i} таковой не является и при этом $q \in Q_i$, где $i = 1, 2$.

Во избежание терминологических коллизий условимся состояния $\langle v_1, v_2, q, z\rangle$ комбинированных машин называть мета-состояниями. Функция переходов $\hat{\varphi}$ выступает в двоякой роли. С одной стороны, $\hat{\varphi}$ обеспечивает прочтение машиной $K(\pi_1, \pi_2, D)$ только таких пар последовательностей вида $tr = (v_1, \Delta_1), (v_2, \Delta_2), \ldots$, которые являются F-совместными трассами в программах π_1 и π_2. В случае нарушения требований F-совместности значение функции $\hat{\varphi}$ становится неопределенным. С другой стороны, машина $K(\pi_1, \pi_2, D)$ проверяет, одинаковы ли результаты тех вычислений, которые прокладывают эти трассы в программах π_1 и π_2. Значение $\hat{\varphi}$ становится неопределенным (и, следовательно, прогон машины прекращается), как только становится ясно, совпадают или не совпадают эти результаты. В последнем случае комбинированная машина допускает эту пару трасс. Более формально эти свойства комбинированной машины установлены в следующих леммах.

Лемма 1. Предположим, что прогон $\hat{\alpha}$ комбинированной машины $K(\pi_1, \pi_2, D)$ начинается в мета-состоянии $\langle v_1, v_2, q, z\rangle$ и достигает мета-состояния $\langle v_1', v_2', q', z'\rangle$. Тогда проекции $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$ этого прогона являются трассами в программах π_1 и π_2, ведущими из точек v_1 и v_2 в точки v_1' и v_2' соответственно.

Доказательство проводится индукцией по длине прогона $\hat{\alpha}$.

Лемма 2. Для любого начального прогона $\hat{\alpha}$ комбинированной машины $K(\pi_1, \pi_2, D)$ проекции $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$ являются F-совместными трассами в программах π_1 и π_2.

Доказательство. Согласно лемме 1 проекции $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$ являются начальными трассами в π_1 и π_2. Чтобы убедиться в F-совместности этих трасс, ограничимся рассмотрением случая, когда $\hat{\alpha}[1] = tr_1, (v'_1, \Delta'_1)$, $\hat{\alpha}[2] = tr_2, (v'_2, \Delta'_2)$ и при этом $[B_1(tr_1)]_F = [B_2(tr_2)]_F$. Поскольку 2-DM D описывает шкалу F, то справедливо, что $\varphi^*(q_0, B_1(tr_1), B_2(tr_2)) = q' \in Q_{accept}$. Для определенности будем считать, что $q' \in Q_1$. Предположим, что $\hat{\varphi}^*(\hat{q}_0, tr_1, tr_2) = \langle v_1', v_2', q', z\rangle$. По определению $\hat{\varphi}$ верны равенства $\hat{\varphi}^*(\hat{q}_0, \hat{\alpha}[1], tr_2) = \hat{\varphi}(\langle v_1', v_2', q', z\rangle, (v_1', \Delta_1')) = \langle T_1(v_1', \Delta_1'), v_2', \varphi(q', B_1(v_1')), \Delta_1'\rangle$. Так как D описывает шкалу F и $[B_2(tr_2)]_F \prec [B_1(\alpha[1])]_F$, то состояние $q'' = \varphi(q', B_1(v_1'))$ принадлежит множеству Q_2. Значит, по определению функции переходов $\hat{\varphi}$ имеет место равенство $\hat{\varphi}^*(\hat{q}_0, \alpha[1], \alpha[2]) = \hat{\varphi}(\langle T_1(v_1', \Delta_1'), v_2', q'', \Delta_1'\rangle, (v_2', \Delta_2'))$. Значение в его левой части определено, а значит, определено и значение в его правой части. Это возможно лишь в случае $\Delta_1' = \Delta_2'$, что свидетельствует о F-совместности трасс $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$. Лемма доказана.

Начальный прогон $\hat{\alpha}$ машины $K(\pi_1, \pi_2, D)$ назовем максимальным, если $\hat{\alpha}$ либо является бесконечным, либо является конечным, но не может быть продолжен.

Лемма 3. Для любой пары F-совместных трасс tr_1 и tr_2 в программах π_1 и π_2 существует такой максимальный прогон $\hat{\alpha}$ машины $K(\pi_1, \pi_2, D)$, что $\hat{\alpha}[1]$ является префиксом tr_1 и $\hat{\alpha}[2]$ является префиксом tr_2.

Доказательство. Справедливость утверждения следует непосредственно из определения комбинированной машины $K(\pi_1, \pi_2, D)$.

Теорема 3. Если 2-DM D описывает шкалу F, то $\pi_1 \approx_F \pi_2$ тогда и только тогда, когда комбинированная машина $K(\pi_1, \pi_2, D)$ удовлетворяет следующим требованиям.

А. $K(\pi_1, \pi_2, D)$ порождает пустое бинарное отношение $E_{K(\pi_1, \pi_2, D)}$.

В. Для каждого бесконечного прогона $\hat{\alpha}$ машины $K(\pi_1, \pi_2, D)$ обе проекции $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$ являются бесконечными трассами.

Доказательство. Далее для краткости обозначения комбинированной машины $K(\pi_1, \pi_2, D)$ будем использовать запись $\hat{K}$.

($\Rightarrow$) Покажем, что из невыполнимости любого из требований А или В следует неэквивалентность программ π_1 и π_2.

Требование А. Предположим, что $E_{\hat{K}} \neq \varnothing$. Тогда некоторый начальный прогон $\hat{\alpha}$ приводит $\hat{K}$ к допускающему мета-состоянию $\hat{q} = \langle v_1, v_2, q, z \rangle \in \hat{Q}_{accept}$. Как следует из определения множества допускающих состояний $\hat{Q}_{accept}$, возможны три случая.

1. Если $v_1 = exit$, $v_2 = exit$ и $q \notin Q_{accept}$, то согласно леммам 1 и 2 проекции $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$ являются полными F-совместными трассами в программах π_1 и π_2, завершающимися в точке *exit*. Так как 2-DM D описывает шкалу F и $q \notin Q_{accept}$, то справедливо неравенство $[B_1(\hat{\alpha}[1])]_F \neq [B_2(\hat{\alpha}[2])]_F$ следовательно, по теореме 2 программы π_1 и π_2 неэквивалентны на шкале F.

2. Предположим, что $v_1 = exit$, $v_2 \neq exit$ и $q \in Q_1$. Тогда по лемме 1 проекция $\hat{\alpha}[1]$ — это полная трасса в π_1, завершающаяся в точке *exit*, $\hat{\alpha}[2]$ — это начальная трасса в π_2, ведущая в точку v_2, при этом $\varphi^*(q_0, B_1(\hat{\alpha}[1]), B_2(\hat{\alpha}[2])) = q$. По лемме 2 трассы $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$ являются F-совместными. Из утверждения 3 следует, что $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$ — префиксы трасс $tr(\pi_1, M)$ и $tr(\pi_2, M)$ для некоторой F-модели M. Если трасса $tr(\pi_2, M)$ бесконечна, то $[comp(\pi_1, M)]_F \neq [comp(\pi_2, M)]_F$. Если трасса $tr(\pi_2, M)$ конечна, то она представима в виде $tr(\pi_2, M) = \hat{\alpha}[2]\,tr$ для некоторой трассы *tr*, ведущей из точки v_2 в *exit*. Поскольку $\varphi^*(q_0, B_1(\hat{\alpha}[1]), B_2(\hat{\alpha}[2]\,tr)) = \varphi^*(q, \lambda, tr)$ и $q \in Q_1$, значение $\varphi^*(q_0, B_1(\hat{\alpha}[1]), B_2(\hat{\alpha}[2]\,tr))$ не определено. Учитывая, что 2-DM D описывает шкалу F, приходим к выводу, что $[comp(\pi_1, M)]_F = [B_1(\hat{\alpha}[1])]_F \neq [B_2(\hat{\alpha}[2]\,tr)]_F = [comp(\pi_2, M)]_F$. Отсюда следует, что программы π_1 и π_2 неэквивалентны на шкале F.

3. Предположим, что v_1 является мертвой точкой, v_2 таковой не является и $q \in Q_1$. Согласно леммам 1 и 2 проекции $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$ — это F-совместные трассы в программах π_1 и π_2, ведущие в точки v_1 и v_2 соответственно, при этом $\varphi^*(q_0, B_1(\hat{\alpha}[1]), B_2(\hat{\alpha}[2])) = q$. Рассмотрим произвольную трассу *tr*, ведущую в программе π_2 из точки v_2 в точку *exit*. Так как для любого $i, 1 \leq i \leq |tr|$, справедливо равенство $\varphi^*(q_0, B_1(\hat{\alpha}[1]), B_2(\hat{\alpha}[2]\,tr^i)) = \varphi^*(q, \lambda, B_2(tr^i))$ и $q \in Q_1$, то значения $\varphi^*(q_0, B_1(\hat{\alpha}[1]), B_2(\hat{\alpha}[2]\,tr^i))$ не определены. Учитывая, что 2-DM D описывает шкалу F, это означает, что $\varphi^*(q_0, B_1(\hat{\alpha}[1]), B_2(\hat{\alpha}[2]\,tr^i))$ для любого $i, 1 \leq i \leq |tr|$. Отсюда согласно утверждению 2 следует, что $\hat{\alpha}[1]$ и $\hat{\alpha}[2]\,tr$ — F-совместные трассы. Согласно утверждению 3 $\hat{\alpha}[1]$ и $\hat{\alpha}[2]\,tr$ — префиксы трасс $tr(\pi_1, M)$ и $tr(\pi_2, M)$ для некоторой F-модели M. Но поскольку $\hat{\alpha}[1]$ ведет в мертвую точку, трасса $tr(\pi_1, M)$ не может завершаться в точке *exit*. Значит, вычисление $comp(\pi_1, M)$ бесконечно и, следовательно, программы π_1 и π_2 неэквивалентны на шкале F.

Требование В. Предположим, что существует такой бесконечный начальный прогон $\hat{\alpha}$ комбинированной машины $\hat{K}$, одна из проекций которого (например, $\hat{\alpha}[1]$) является бесконечной трассой, а другая (в данном случае $\hat{\alpha}[2]$) — конечная трасса, ведущая в точку v_2. Тогда по определению $\hat{K}$ существует такое $i_0, i_0 > 0$, что для каждого $j, j \geq i_0$, имеет место включение $\varphi^*(q_0, B_1(\hat{\alpha}[1]^j), B_2(\hat{\alpha}[2])) = q_j \in Q_1$. Заметим, что $q_j \notin Q_{accept}$, поскольку в противном случае для любого оператора $a, a \in A$, выполнялось бы равенство $\lfloor B_1(\hat{\alpha}[1]^j)a \rfloor_F = \lfloor B_2(\hat{\alpha}[2]a \rfloor_F$. Учитывая, что 2-DM D описывает шкалу F, это равенство привело бы к неопределенному значению $\varphi^*(q_0, B_1(\hat{\alpha}[1]^{j+2}), B_2(\hat{\alpha}[2]))$.

Далее рассмотрим два случая.

1. Точка v_2 является мертвой. Для произвольного j_0, $j_0 \geq i_0$, рассмотрим точку v_1, в которую в программе π_1 ведет трасса $\alpha[1]^{j_0}$. Как следует из определения функции переходов $\hat{\varphi}$, точка v_1 не может быть мертвой, поскольку в противном случае такой бесконечный прогон $\hat{\alpha}$ был бы невозможен. Поэтому в про-

грамме π_1 есть трасса *tr*, ведущая из точки v_1 в *exit*. Заметим, что трассы $\alpha[1]^{j_0}\, tr$ и $\alpha[2]$ являются F-совместными. Действительно, в противном случае равенство $\lfloor B_1(\hat{\alpha}[1]^{j_0}\, tr^k)\rfloor_F = \lfloor B_2(\hat{\alpha}[2]^m)\rfloor_F$ выполнялось бы для некоторых k, m, а это привело бы к тому, что оба значения $\varphi^*(q_0, B_1(\hat{\alpha}[1]^{j_0}), B_2(\hat{\alpha}[2]))$ и $\varphi^*(q_0, B_1(\hat{\alpha}[1]^{j_0}\, tr^k), B_2(\hat{\alpha}[2]^m))$ были бы определены вопреки тому, что D является детерминированной машиной. Согласно утверждению 3 F-совместность трасс $\alpha[1]^{j_0}\, tr$ и $\alpha[2]$ гарантирует существование такой F-модели M, для которой $\alpha[1]^{j_0}\, tr = tr(\pi_1, M)$, а $\alpha[2]$ является префиксом $tr(\pi_2, M)$. Поскольку v_2 — это мертвая точка, полная трасса $tr(\pi_2, M)$ не может вести в точку *exit* и поэтому является бесконечной трассой. Таким образом, вычисление $comp(\pi_1, M)$ оказывается конечным, а вычисление $comp(\pi_2, M)$ — бесконечным. Это означает, что программы π_1 и π_2 неэквивалентны на шкале F.

2. Точка v_2 не является мертвой. Рассмотрим в программе π_2 произвольную трассу *tr*, ведущую из точки v_2 в точку *exit*. Как было отмечено ранее, для любого $j, j \ge i_0$, выполняется включение $\varphi^*(q_0, B_1(\hat{\alpha}[1]^j), B_2(\hat{\alpha}[2])) = q_j \in Q_1$. Поэтому для любой пары $j, m,\ j \ge i_0, 0 < m < |tr|$, значение $\varphi^*(q_0, B_1(\hat{\alpha}[1]^j), B_2(\hat{\alpha}[2]\, tr^m))$ не определено. Исходя из того, что 2-DM D описывает шкалу F, неравенство $\lfloor B_1(\hat{\alpha}[1]^j)\rfloor_F \ne \lfloor B_2(\hat{\alpha}[2]\, tr^m)\rfloor_F$ выполняется для каждой такой пары j, m. Следовательно, $\hat{\alpha}[1]$ и $\hat{\alpha}[2]\, tr$ — полные F-совместные трассы в программах π_1 и π_2, причем одна из них является бесконечной, а другая — конечной. Согласно теореме 2 это означает, что программы π_1 и π_2 неэквивалентны на шкале F.

($\Leftarrow$) Рассмотрим произвольную пару полных F-совместных трасс tr_1 и tr_2 в программах π_1 и π_2. Согласно лемме 3 комбинированная машина $\hat{K}$ имеет максимальный прогон $\hat{\alpha}$, проекции $\hat{\alpha}[1]$ и $\hat{\alpha}[2]$ которого являются префиксами трасс tr_1 и tr_2 соответственно. Возможны два варианта устройства прогона $\hat{\alpha}$.

1. Максимальный прогон $\hat{\alpha}$ является конечным. Тогда он достигает такого мета-состояния $\hat{q} = \langle v_1, v_2, q, z\rangle$, для которого значение функции переходов $\hat{\varphi}(\hat{q}, (v, \Delta))$ не определено ни для одной пары (v, Δ). Поскольку требование A выполнено, мета-состояние $\hat{q}$ не может быть допускающим. Поэтому существуют лишь две причины, не позволяющие продолжить прогон $\hat{\alpha}$: либо $v_1 = exit$, $v_2 = exit$ и $q \in Q_{accept}$, либо обе точки v_1 и v_2 являются мертвыми.

Так как 2-DM D описывает шкалу F, в первом случае выполняется равенство $[B_1(tr_1)]_F = [B_2(tr_2)]_F$, а во втором случае обе трассы tr_1 и tr_2 являются бесконечными.

2. Максимальный прогон $\hat{\alpha}$ является бесконечным. Тогда требование B гарантирует, что обе проекции $tr_1 = \alpha[1]$ и $tr_2 = \alpha[1]$ являются бесконечными трассами.

Таким образом, для любой пары полных F-совместных трасс tr_1 и tr_2 в программах π_1 и π_2 либо обе трассы оказываются бесконечными, либо $[B_1(tr_1)]_F = [B_2(tr_2)]_F$. Согласно теореме 2 это означает, что $\pi_1 \approx_F \pi_2$.

Теорема доказана.

Основное достоинство теоремы 3 состоит в том, что задача проверки эквивалентности программ $\pi_1 \approx_F \pi_2$ сводится к проверке пустоты 2-DM $K(\pi_1, \pi_2, D)$, в структуре которой отражаются индивидуальные особенности семантики программных операторов. Выделив классы комбинированных машин, для которых проблема пустоты решается эффективно, можно установить и те алгебраические свойства операторов, которые целесообразно учитывать при разработке быстрых алгоритмов проверки эквивалентности и минимизации программ. Эта задача требует проведения отдельного исследования, результаты которого будут опубликованы в последующих работах. Приведенное ниже простое следствие из теоремы 3 показывает, что это направление исследований является перспективным.

Теорема 4. Если динамическая шкала F может быть описана 2-DM с конечным числом состояний (двухленточным детерминированным конечным автоматом), то задача проверки эквивалентности программ $\pi_1 \approx_F \pi_2$ принадлежит классу сложности NLOGSPACE.

Доказательство. Если 2-DM D, описывающая шкалу F, имеет n состояний, то комбинированная машина $K(\pi_1, \pi_2, D)$ имеет $O(n|\pi_1||\pi_2|)$ мета-состояний. Проверка требований A и B из теоремы 3 сводится, таким образом, к NLOGSPACE-полной задаче проверки достижимости вершин в конечном ориентированном графе.

Существует целый ряд шкал, отражающих алгебраические свойства реальных программных операторов и удовлетворяющих условиям теоремы 4. Полугруппы операторов являются частным случаем динамических шкал: нейтральный элемент служит начальным состоянием данных, а интерпретация операторов определяется равенством $R(s,a) = s \circ a$. Тогда условиям теоремы 4 удовлетворяют, например, полугрупповые шкалы, заданные системами тождеств следующего вида:

1) $ab = b$ (в случае, когда из этих тождеств не следует равенства $aa = a$),
2) $ab = ac$,
3) $ba = ca$.

Здесь a, b, c — программные операторы.

Возможности применения нового метода, сводящего решение проблемы эквивалентности программ к проверке пустоты двухленточных машин специального вида, не исчерпываются этим простым примером. В последующей работе будет показано, что комбинированные машины $K(\pi_1, \pi_2, D)$ обладают целым рядом замечательных свойств. Опираясь на эти свойства, во многих случаях удается конструировать быстрые алгоритмы проверки эквивалентности программ, семантика операторов которых описывается с помощью упорядоченных динамических шкал.

СПИСОК ЛИТЕРАТУРЫ

1. Ляпунов А.А., Янов Ю.И. О логических схемах программ // Тр. конф. «Пути развития советского математического машиностроения и приборостроения». Ч. 3, 1956. — С. 5–8.
2. Янов Ю.И. О логических схемах алгоритмов // Проблемы кибернетики. Вып. 1. — М.: Физматгиз, 1958. — С. 75–121.
3. Глушков В.М. Теория автоматов и формальные преобразования микропрограмм // Кибернетика. — 1965. — № 5. — С. 1–9.
4. Глушков В.М., Летичевский А.А. Теория дискретных преобразователей // Избранные вопросы алгебры и логики: сб. статей. — Новосибирск: Наука, 1973. — С. 5–39.
5. Luckham D.C., Park D.M., Paterson M.S. On formalized computer programs // J. of Computer and System Science. — 1970. — **4**, N 3. — P. 220–249.
6. Paterson M.S. Program schemata // Machine Intelligence. — Edinburg: Univ. Press., 1968. — **3.** — P. 19–31.
7. De Bekker J.W., Skott D.A. Theory of programs. Unpublished notes. — Vienna: IBM Seminar, 1969.
8. Подловченко Р.И. Полугрупповые модели программ // Программирование. — 1981. — № 4. — С. 3–13.
9. Ершов А.П. Современное состояние теории схем программ // Проблемы кибернетики, Вып. 27. — М.: Наука, 1973. — С. 87–110.
10. Летичевский А.А. Функциональная эквивалентность дискретных преобразователей. II // Кибернетика. — 1970. — № 2 — С. 14–28.
11. Годлевский А.Б. Об одном случае специальной проблемы функциональной эквивалентности дискретных преобразователей // Кибернетика. — 1974. — № 3. — С. 32–36.
12. Летичевский А.А. Эквивалентность автоматов с заключительным состоянием относительно свободной полугруппы с правым нулем // Докл. АН СССР. — 1968. — **182**, № 5.
13. Летичевский А.А. Функциональная эквивалентность дискретных преобразователей. III // Кибернетика. — 1972. — № 1. — С. 1–4.
14. Летичевский А.А. Эквивалентность автоматов относительно полугрупп с сокращением // Проблемы кибернетики. Вып. 27. — М.: Физматгиз, 1973. — С. 195–212.
15. Летичевский А.А., Смикун Л.Б. О группах с разрешимой проблемой эквивалентности // Тез. 4-й Всесоюз. конф. по математической логике, 1976. — С. 77.
16. Тайцлин М.А. Эквивалентность автоматов относительно коммутативной полугруппы // Алгебра и логика. — 1968. — **8**, № 5. — С. 553–600.
17. Лисовик Л.П. Металинейные схемы с засылками констант // Программирование. — 1985. — № 2. — С. 29–38.

18. Лисовик Л.П. Стандартные схемы с магазинами // Докл. АН УССР. — 1989. — № 12. — С. 23–27.
19. 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.
20. Сабельфельд В.К. О преобразованиях унарных линейных рекурсивных программ // Кибернетика. — 1975. — № 5. — С. 55–63.
21. Сабельфельд В.К. Полиномиальная оценка сложности распознавания логико-термальной эквивалентности // Докл. АН СССР. — 1979. — **249**, № 4. — С. 793–796.
22. Сабельфельд В.К. Новый класс схем с разрешимой функциональной эквивалентностью // Информатика: инструментальные средства. — Новосибирск, 1988. — С. 109–126.
23. Ashcroft E., Manna Z., Pnueli A. A decidable properties of monadic functional schemes // J. of the ACM. — 1973. — **20**, N 3. — P. 489–499.
24. Paterson M.S. Decision problems in computational models // SIGPLAN Notices. — 1972. — **7.** — P. 74–82.
25. Котов В.Е., Сабельфельд В.К. Теория схем программ. — М.: Наука, 1991. — 348 с.
26. Лисовик Л.П. Схемы программ и преобразователи над размеченными деревьями. Вып. 2 // Мат. вопросы кибернетики. — М.: Наука, 1996. — С. 281–320.
27. Valiant L.G. The equivalence problem for deterministic finite-turn push-down automata // Information and Control. — 1974. — **25**, N 2. — P. 123–133.
28. Летичевский А.А. Практические методы распознавания эквивалентности дискретных преобразователей и схем программ // Кибернетика. — 1973. — № 4. — С. 15–26.
29. Подловченко Р.И., Захаров В.А. Полиномиальный по сложности алгоритм, распознающий коммутативную эквивалентность схем программ // Докл. РАН. — 1998. — **362**, № 6. — С. 27–31.
30. Захаров В.А. Быстрые алгоритмы разрешения эквивалентности операторных программ на уравновешенных шкалах // Мат. вопросы кибернетики. Вып. 7. — М.: Наука, 1998. — С. 303–324.
31. Zakharov V.A. An efficient and unified approach to the decidability of equivalence of propositional programs // Lecture Notes in Computer Science. — 1998. — **1443.** — P. 247–259.
32. Захаров В.А. Быстрые алгоритмы разрешения эквивалентности пропозициональных операторных программ на упорядоченных полугрупповых шкалах // Вестник Моск. ун-та. Сер. 15. Вычисл. математика и кибернетика. — 1999. — № 3. — С. 29–35.
33. Захаров В.А. Об эффективной разрешимости проблемы эквивалентности линейных унарных рекурсивных программ // Мат. вопросы кибернетики. Вып. 8. — М.: Наука, 1999. — С. 255–273.
34. Zakharov V.A. On the decidability of the equivalence problem for orthogonal sequential programs // Grammars. — 1999. — **2**, N 3. — P. 271–181.
35. Zakharov V.A. On the decidability of the equivalence problem for monadic recursive programs // Theoretical Informatics and Applications. — 2000. — **34**, N 2. — P. 157–171.
36. Zakharov V., Zakharyaschev I. On the equivalence-checking problem for a model of programs related with multi-tape automata // Lecture Notes in Computer Science. — 2005. — **3317.** — P. 293–305.
37. Щербина В.Л., Захаров В.А. Эффективные алгоритмы проверки эквивалентности программ в моделях, связанных с обработкой прерываний // Вестник Москов. ун-та. Сер. 15. Вычисл. математика и кибернетика. — 2008. — № 2. — С. 33–41.
38. Подловченко Р.И. Техника следов в разрешении проблемы эквивалентности в алгебраических моделях программ // Кибернетика и системный анализ. — 2009. — № 5. — С. 25–37.
39. Подловченко Р.И. Иерархия моделей программ // Программирование. — 1981. — № 2. — С. 3–14.
40. Podlovchenko R., Rusakov D., Zakharov V. On the equivalence problem for programs with mode switching // Lecture Notes in Computer Science. — 2006. — **3845.** — P. 351–352.
41. Harel D. Dynamic logics // Handbook of Philosophical Logics, D. Gabbay and F. Guenthner (eds.), 1984. — P. 497–604.
42. Johnson J.H. Rational equivalence relations // Theoretical Computer Science. — 1986. — **47**, N 1. — P. 39–60.
43. Vardi M. Y., Wolper P. Automata-theoretic techniques for modal logics of programs // J. of Computer and System Science. — 1986. — **32**, N 2. — P. 183–221.
44. Bouajjani A., Esparza J., Maller O. Reachability analysis of pushdown automata: application to model checking // Lecture Notes in Computer Science. — 1997. — **1243.** — P. 135–150.

Поступила 09.04.2010