Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики

На правах рукописи

Захаров Владимир Анатольевич

ПРОБЛЕМА ЭКВИВАЛЕНТНОСТИ ПРОГРАММ: МОДЕЛИ, АЛГОРИТМЫ, СЛОЖНОСТЬ

Специальность 01.01.09 —

Дискретная математика и математическая кибернетика

А В Т О Р Е Ф Е Р А Т

диссертации на соискание ученой степени

доктора физико-математических наук

Москва — 2011

Работа выполнена на кафедре математической кибернетики факультета вычислительной математики и кибернетики Московского государственного университета имени М.В. Ломоносова

Официальные оппоненты	доктор физико-математических наук, академик НАН Украины Летичевский Александр Адольфович
	доктор физико-математических наук, профессор Ломазова Ирина Александровна
	доктор физико-математических наук, профессор Соколов Валерий Анатольевич
Ведущая организация	Учреждение Российской академии наук Институт систем информатики им. А.П. Ершова Сибирского отделения РАН (ИСИ СО РАН)

Защита состоится 02 марта 2012 г. в 11.00 на заседании диссертационного совета Д 501.001.44 при Московском государственном университете имени М.В. Ломоносова по адресу 119991, ГСП-1, Москва, Ленинские горы, МГУ, 2-й учебный корпус, факультет ВМК, ауд. 685.

С диссертацией можно ознакомиться в библиотеке факультета ВМК МГУ

Автореферат разослан 29 ноября 2011 г.

Ученый секретарь
диссертационного совета
профессор Трифонов Н. П.

Общая характеристика диссертационной работы

Тема исследований диссертационной работы — проблема эквивалентности программ. Исследования охватывают несколько аспектов этой проблемы: математические модели программ, используемые для ее формализации, отношения между этими моделями, методы построения разрешающих алгоритмов в некоторых моделях программ, сложность проблемы эквивалентности.

Проблема эквивалентности программ состоит в том, чтобы для произвольной заданной пары программ выяснить, имеют ли эти программы одинаковое поведение. Строгие определения терминов «программа» и «поведение», фигурирующих в этой формулировке, могут варьироваться, и поэтому проблема эквивалентности программ охватывает целый спектр задач проверки схожести разных видов поведения программ в различных моделях вычислений.

Проблема эквивалентности является одной из первичных проблем в теории вычислений. Это обусловлено, в первую очередь, тем, что ее формулировка опирается на определения всего лишь двух базовых понятий всякой вычислительной модели — программы и ее вычислений. Поэтому проблема эквивалентности программ — это одна из первых содержательных задач, возникающих при построении всякой модели вычислений.

Проблема эквивалентности имеет важное эпистемологическое значение. Понимание смысла объектов некоторого класса проявляется, в частности, в способности распознавать, имеют ли два объекта одинаковый смысл. Смысл программы определяется ее вычислениями. Поэтому, изучая методы решения и сложность проблемы эквивалентности, мы тем самым оцениваем уровень математических средств и объем вычислительных ресурсов, которые потребуются для решения других задач семантического анализа программ. Формально это объясняется тем, что большое число задач анализа программ может быть сведено к проблеме эквивалентности программ.

С проблемой эквивалентности программ сталкиваются при решении ряда задач системного программирования и компьютерной безопасности; к их

числу относятся задачи оптимизации, верификации, реорганизации и обфускации программ, задача обнаружения вредоносных программ (вирусов) и др.

Принципиальная трудность задач семантического анализа программ объясняется тем, что в любой «естественной» универсальной системе программирования любое нетривиальное функциональное свойство программ нерекурсивно (теорема Райса-Успенского). Этот факт не отменяет возможности получения эффективно проверяемых достаточных условий функциональной эквивалентности программ, однако, ни одно из этих достаточных условий не будет необходимым. Систематический подход к поиску эффективно проверяемых достаточных условий эквивалентности программ и построению систем эквивалентных преобразований был предложен и развит в работах А.А. Ляпунова, Ю.И. Янова, А.П. Ершова, В.М. Глушкова, А.А. Летичевского, Р.И. Подловченко, В.К. Сабельфельда, М.С. Патерсона, З. Манны, Ш. Грейбах, Р. Милнера. Эти работы привели к созданию и развитию теории схем программ, в рамках которой сформировалась общая методика построения и применения моделей программ для решения проблем эквивалентности и эквивалентных преобразований. Основные положения этой методики таковы.

1). Формируется параметризованная модель вычислений, представляющая собой семейство моделей программ $\mathcal{M}(\sigma)$. Объекты каждой модели обычно называются схемами программ. Параметр σ определяет семантику базовых компонентов схем программ в модели $\mathcal{M}(\sigma)$ и позволяет ввести понятие вычисления и отношение функциональной эквивалентности схем программ $\sim_\sigma$.

2). В семействе моделей программ вводится отношение аппроксимации $\sqsubseteq$ таким образом, чтобы достаточным условием эквивалентности пары схем программ в модели $\mathcal{M}(\sigma_1)$ была эквивалентность этих же схем программ в аппроксимирующей модели $\mathcal{M}(\sigma_2)$.

3). Выделяется подкласс $\mathcal{ED}$ моделей программ $\mathcal{M}(\sigma)$, в которых задача проверки эквивалентности схем программ $\pi' \sim_\sigma \pi''$ имеет приемлемое решение. Для моделей программ семейства $\mathcal{ED}$ разрабатываются эффективные алгоритмы проверки эквивалентности схем программ.

Чтобы применить описанную методику для решения задачи проверки эквивалентности программ в заданной системе программирования, достаточно

а) выбрать модель программ $\mathcal{M}(\sigma_0)$, соответствующую этой системе программирования;

б) выбрать в классе $\mathcal{ED}$ минимальную по отношению аппроксимации модель программ $\mathcal{M}(\sigma_1)$, удовлетворяющую условию $\mathcal{M}(\sigma_0) \sqsubseteq \mathcal{M}(\sigma_1)$;

в) воспользоваться алгоритмом проверки эквивалентности схем программ в модели $\mathcal{M}(\sigma_1)$.

В соответствии с описанной методикой в диссертационной работе для решения задачи проверки функциональной эквивалентности программ предложены формальные системы вычислений, позволяющие моделировать поведение последовательных императивных и рекурсивных программ. Программы представляются на пропозициональном уровне абстракции конечными размеченными системами переходов или аннотированными контекстно-свободными грамматиками над конечными алфавитами базовых операторов и предикатов. Для интерпретации операторов и предикатов применяются динамические модели — структуры Крипке, заимствованные из пропозициональной динамической логики. Для каждого вида программ введены понятия вычисления программы в заданной динамической модели и результата вычисления. На основании этих понятий определено отношение эквивалентности программ на множествах динамических моделей (динамических семантик), а для динамических семантик определено отношение аппроксимации.

Цель диссертационной работы — решение следующих задач.

1. Установить необходимые и достаточные условия выполнимости отношения аппроксимации динамических семантик, а также тип отношения аппроксимации в различных классах динамических семантик.

2. Разработать общий метод построения эффективных алгоритмов проверки эквивалентности программ в рассматриваемых формальных системах вычислений.

3. Выделить классы динамических семантик, для которых задача проверки эквивалентности программ разрешима, и, в том числе, классы динамических семантик, для которых эта задача разрешима за время, полиномиально зависящее от размеров анализируемых программ.

Методы исследования. Для решения поставленных задач в работе применялись методы теории автоматов, теорий групп и полугрупп, теории сложности вычислений, комбинаторики, теории чисел.

Научная новизна и практическая ценность работы. Диссертационная работа представляет собой теоретическое исследование. Все результаты, представленные в ней, являются новыми и получены автором самостоятельно.

Теоремы главы 4, в которых приведено решение первой из указанных выше задач, могут быть использованы для выбора подходящих математических моделей программ, для сравнения выразительных возможностей этих моделей при решении задач семантического анализа компьютерных программ, а также для перенесения результатов решения этих задач из одних моделей программ в другие.

Принципиально новым является метод совместных вычислений для построения алгоритмов проверки эквивалентности программ, описанный в главах 5 и 6. В отличие от всех известных подходов к решению проблемы эквивалентности программ в моделях вычислений метод совместных вычислений параметризован относительно семантик, задающих интерпретацию базовых компонентов программ. Эта особенность метода совместных вычислений дает возможность единообразно конструировать разрешающие процедуры для широкого класса моделей последовательных, рекурсивных и простейших реагирующих программ; в отдельных случаях с его помощью удается построить алгоритмы, проверяющие эквивалентность программ за время, полиномиальное относительно их размера. Метод совместных вычислений может найти применение при разработке программно-инструментальных средств анализа поведения компьютерных программ.

Апробация работы Результаты исследований, изложенные в диссертационной работе, были представлены и обсуждены на следующих научных форумах: Международные конференции «Проблемы теоретической кибернетики» (Горький 1988, Волгоград 1991, Саратов 1993, Ульяновск 1996, Нижний Новгород 1999, Казань 2002, Пенза 2005, Казань 2008, Нижний Новгород 2011), Международные конференции «Дискретные модели в теории управляющих систем» (Красновидово 1997, Красновидово 1998, Красновидово 2000, Ратмино 2003, Москва 2004, Москва 2009), Международный семинар «Дискретная математика и ее приложения» (Москва 1988, Москва 2001, Москва 2004, Москва 2007), Международная алгебраическая конференция памяти А.Г.Куроша (Москва 1998), Всероссийская научная конференция «Методы и средства обработки информации» (Москва 2005), Международная школа-семинар «Синтез и сложность управляющих систем» (Санкт-Петербург 2006), Научно-практическая конференция «Информационная безопасность» (Таганрог 2006, Таганрог 2007), Международная конференция «Логика, методология, философия науки» (Обнинск, 1995), International Colloquium on Automata, Languages and Programming (Aalborg 1998), Mathematical Foundations of Computer Science Workshop on Grammar Systems (Brno, 1998), Logic Colloquium 2000 (Paris 2000), Fundamentals of Computation Theory Workshop on Formal Languages and Automata (Iassy 1999), International Conference «Machines, Computations, and Universality» (Chisinau 2001), International Workshop on Program Understanding (Алтай 2003, Новосибирск 2009), Congress of Mathematics of Serbia and Montenegro (Petrovac, 2004), International Conference on Implementations and Applications of Automata (Kingston 2004, Sophia Antipolis 2005), International Workshop «Automata, algorithms, and information technologies» (Киев, 2010)

Публикации. Материал диссертации опубликован в 60 печатных научных трудах (включая 15 публикаций в изданиях из списка ВАК): 20 статей в журналах и периодических изданиях, 40 статей в сборниках научных трудов и тезисов конференций.

Структура и объем работы. Диссертационная работа содержит 438 страниц машинописного текста; она состоит из 7 глав, включая введение и заключение. Список литературы содержит 550 наименований.

КРАТКОЕ СОДЕРЖАНИЕ ДИССЕРТАЦИИ

Первая глава диссертации — введение. В ней обсуждается математическая значимость проблемы эквивалентности программ в моделях вычислений, описан ряд задач системного программирования и компьютерной безопасности, при решении которых приходится сталкиваться с этой проблемой, объяснены принципиальные трудности ее алгоритмического решения и описан общий подход, применяемый в теории схем программ для получения эффективно проверяемых достаточных условий эквивалентности программ. Сформулированы ключевые вопросы, ответы на которые необходимо получить для практического применения указанной методики частичного решения проблемы эквивалентности программ, и перечислены математические предпосылки для решения поставленных задач. Приведено краткое содержание диссертационной работы и представлены ее основные результаты.

Во **второй главе** введены математические модели последовательных и рекурсивных программ, описаны их синтаксис и семантика, сформулированы основные задачи, решению которых посвящена диссертационная работа. Также рассмотрены простейшие свойства вычислений программ, необходимые для решения поставленных задач в последующих главах.

В **разделе 2.1** описана пропозициональная модель вычислений последовательных императивных программ; она является центральным объектом изучения всей работы. Эта модель программ представляет собой интеграцию и развитие двух концепций построения формальных моделей последовательных (операторных) программ, восходящих к схемам программ Ляпунова–Янова, — теории дискретных преобразователей Глушкова–Летичевского и теории алгебраических моделей программ Подловченко.

Последовательные пропозициональные программы определяются над конечными множествами *базовых операторов* $\mathcal{A}$ и *базовых предикатов* $\mathcal{P}$. Базовые операторы обозначают элементарные вычислительные действия, осуществляющие преобразования данных. Конечные последовательности базовых операторов называются $\mathcal{A}$-*цепочками*. Базовые предикаты обозначают элементарные отношения на множестве данных. Всякий набор истинностных значений всех базовых предикатов называется *логическим условием*. Множество всех логических условий предикатов из $\mathcal{P}$ обзначается записью $\mathcal{C}_\mathcal{P}$.

Определение 2.1.1. *Пропозициональная последовательная программа* — это конечная размеченная система переходов $\pi = \langle V, \mathbf{start}, \mathbf{stop}, T, B \rangle$, компонентами которой являются

- V — конечное множество *точек* программы, $V \neq \emptyset$,
- $\mathbf{start}$ — *точка входа в программу*, $\mathbf{start} \in V$,
- $\mathbf{stop}$ — *точка выхода из программы*, $\mathbf{stop} \in V$,
- $T : (V \setminus \{\mathbf{stop}\}) \times \mathcal{C}_\mathcal{P} \to V$ — тотальная *функция переходов* из одних точек программ в другие;
- $B : V \setminus \{\mathbf{stop}\} \to \mathcal{A}^*$ — *функция привязки*, ставящая в соответствие каждой точке программы, за исключением точки выхода, $\mathcal{A}$-цепочку.

Точки программы соответствуют ее линейным участкам; в каждой точке v выполняется последовательность операторов $B(v)$. Совокупность переходов из каждой точки v программы соответствует оператору ветвления, передающего управление в точки $T(v, \Delta)$ в зависимости от набора Δ, $\Delta \in \mathcal{C}_\mathcal{P}$, истинностных значений базовых предикатов.

Трассой, исходящей из точки v в программе π, называется всякая последовательность пар

$$tr = (v_1, \Delta_1), (v_2, \Delta_2), \ldots, (v_i, \Delta_i), (v_{i+1}, \Delta_{i+1}), \ldots ,$$

в которой $v_1 = v$ и для любого $i, i \geq 1$, выполняются следующие отношения: $v_i \in V$, $\Delta_i \in \mathcal{C}_\mathcal{P}$ и $v_{i+1} = T(v_i, \Delta_i)$. Трасса, исходящая из точки входа $\mathbf{start}$, называется *начальной*. Начальная трасса называется *полной*, если она бесконечна или завершается такой парой (v_n, Δ_n), что $T(v_n, \Delta_n) = \mathbf{stop}$.

Семантика пропозициональных последовательных программ описывается посредством шкал и моделей пропозициональной динамической логики. Интерпретации базовых операторов задают динамические шкалы.

Определение 2.1.3. *Динамическая шкала* задается тройкой $\mathcal{F} = \langle S, s_0, R \rangle$, состоящей из

— непустого счетного множества *состояний данных* S,

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

— всюду определенной *функции преобразования данных* $R : \mathcal{A} \times S \to S$.

Функция преобразования данных R распространяется на множество $\mathcal{A}$-цепочек посредством равенств $R^*(\varepsilon, s) = s$ и $R^*(ah, s) = R^*(h, R(a, s))$ для любых $\mathcal{A}$-цепочек h, операторов a и состояний данных s. Состояние данных s'' называется *вычислимым из состояния данных* s' на шкале $\mathcal{F}$ (для обозначения этого отношения используется запись $s' \preceq_{\mathcal{F}} s''$), если $s'' = R^*(h, s')$ для некоторой $\mathcal{A}$-цепочки h. Для каждой операторной цепочки h запись $[h]_{\mathcal{F}}$ обозначает состояние данных $s = R^*(h, s_0)$, вычислимое на шкале $\mathcal{F}$ из начального состояния данных s_0 при выполнении операторов $\mathcal{A}$-цепочки h.

В диссертационной работе особое внимание уделено двум классам шкал — упорядоченным и полугрупповым. Динамическая шкала $\mathcal{F}$ называется *упорядоченной*, если отношение вычислимости $\preceq_{\mathcal{F}}$ является отношением частичного порядка. Шкала $\mathcal{F}$ называется *полугрупповой*, если множество ее состояний данных образует моноид $\langle S, \circ, \lambda \rangle$, порожденный множеством базовых операторов $\mathcal{A}$; при этом нейтральный элемент моноида λ играет роль начального состояния данных, а функция преобразования данных R определяется равенством $R(a, s) = s \circ a$ для любых базовых операторов a и состояний данных s.

Полную интерпретацию всех элементов пропозициональных программ задают динамические модели.

Определение 2.1.4. *Динамической моделью* M, базирующейся на шкале $\mathcal{F}$, называется пара $(\mathcal{F}, \xi)$, в которой $\xi : S \times \mathcal{P} \to \{0, 1\}$ — *оценка базовых предикатов* на шкале $\mathcal{F}$.

Для каждого состояния данных s и базового предиката p оценка $\xi(s,p)$ определяет значение истинности предиката p в состоянии данных s. Оценку базовых предикатов ξ на шкале $\mathcal{F}$ можно распространить на множество логических условий $\mathcal{C}_{\mathcal{P}}$: будем использовать запись $\xi(s)$ для обозначения логического условия $\Delta = \langle \xi(s,p_1), \xi(s,p_2), \ldots, \xi(s,p_{|\mathcal{P}|}) \rangle$.

Определение 2.1.5. Всякое множество динамических моделей σ называется *динамической семантикой* программ. Динамическая семантика, все модели которой базируются на одной и той же шкале, называется *однородной динамической семантикой*.

Для обозначения однородных динамических семантик, базирующихся на шкале $\mathcal{F}$, используется запись вида $\sigma(\mathcal{F}, L)$, где L — множество допустимых оценок базовых предикатов. Множество всех моделей, базирующихся на шкале $\mathcal{F}$, обозначется записью $\sigma(\mathcal{F})$.

Определение 2.1.6. *Вычислением* программы $\pi = \langle V, \mathbf{start}, \mathbf{stop}, B, T \rangle$ в модели $M = \langle S, s_0, R, \xi \rangle$ 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:
— $v_0 = \mathbf{start}$, s_0 — начальное состояние данных;
— для любого $i,\ i \geq 1$, выполняются равенства $s_i = R^*(B(v_{i-1}), s_{i-1})$ и $v_i = T(v_{i-1}, \xi(s_i))$.

Если вычисление $comp(\pi, M)$ завершается парой $(\mathbf{stop}, s_N)$, то оно считается *успешным*, а состояние данных s_N называется *результатом вычисления* и обозначается записью $[comp(\pi, M)]$. Если $comp(\pi, M)$ — бесконечная последовательность, то это вычисление называется *бесконечным*, и его результат $[comp(\pi, M)]$ считается неопределенным.

Определение 2.1.7. Две программы π_1 и π_2 называются
— *эквивалентными в динамической модели* M (обозначается $\pi_1 \sim_M \pi_2$), если $[comp(\pi_1, M)] = [comp(\pi_2, M)]$;

— *эквивалентными в динамической семантике* σ (обозначается $\pi_1 \sim_\sigma \pi_2$), если отношение $\pi_1 \sim_M \pi_2$ выполняется для любой модели M из σ;
— *эквивалентными на динамической шкале* $\mathcal{F}$ (обозначается $\pi_1 \sim_\mathcal{F} \pi_2$), если выполняется отношение $\pi_1 \sim_{\sigma(\mathcal{F})} \pi_2$.

Определение 2.1.8. Динамическая семантика σ_2 *аппроксимирует* динамическую семантику σ_1 (обозначается $\sigma_1 \sqsubseteq \sigma_2$), если для любой пары π_1 и π_2 выполняется соотношение $\pi_1 \sim_{\sigma_2} \pi_2 \;\Rightarrow\; \pi_1 \sim_{\sigma_1} \pi_2$. Динамические семантики σ_1 и σ_2 называются *равносильными* (обозначается $\sigma_1 \simeq \sigma_2$), если выполняются оба соотношения $\sigma_1 \sqsubseteq \sigma_2$ и $\sigma_2 \sqsubseteq \sigma_1$.

В диссертационной работе для модели вычислений последовательных пропозициональных программ исследуются две основные задачи.

Задача проверки аппроксимируемости динамических семантик. Для двух заданных семантик (множеств динамических моделей) σ_1 и σ_2 выяснить, аппроксимирует ли семантика σ_2 семантику σ_1: $\sigma_1 \sqsubseteq \sigma_2$?

Задача проверки эквивалентности программ. Для заданной динамической семантики σ и пары последовательных пропозициональных программ π_1 и π_2 выяснить, эквивалентны ли эти программы в семантике σ: $\pi_1 \sim_\sigma \pi_2$?

Для решения первой из этих задач на множестве динамических семантик введено отношение гомоморфного сужения. Для динамической семантики σ запись $hom(\sigma)$ обозначает множество всех динамических моделей, каждая из которых является гомоморфным образом какой-либо динамической модели из множества σ. Динамическая семантика σ_1 является *гомоморфным сужением* динамической семантики σ_2, если выполняется включение $\sigma_1 \subseteq hom(\sigma_2)$. Достаточное условие аппроксимируемости динамических семантик описывает

Теорема 2.1.9. *Для любой пары семантик* σ_1, σ_2 *верно соотношение*

$$\sigma_1 \subseteq hom(\sigma_2) \;\Rightarrow\; \sigma_1 \sqsubseteq \sigma_2 \;.$$

Взаимосвязь между отношениями аппроксимации и гомоморфного сужения динамических семантик пропозициональных программ составляет предмет исследования главы 4 диссертационной работы.

Для решения задачи проверки эквивалентности программ с каждым вычислением $comp(\pi, M)$ программы π ассоциируется трасса $tr(\pi, M)$ в программе π — путь в системе переходов, исходящий из точки входа, — реализуемая в динамической модели M. На множествах начальных трасс в программах введено отношение совместности трасс: трассы tr_1 и tr_2 в программах π_1 и π_2 *совместны* в динамической семантике σ, если для некоторой динамической модели M, $M \in \sigma$, последовательности tr_1 и tr_2 являются префиксами трасс $tr(\pi_1, M)$ и $tr(\pi_2, M)$. В терминах отношения совместности трасс сформулированы необходимые и достаточные условия эквивалентности программ (теорема 2.1.5); на их основе в главе 5 разработаны новые методы и алгоритмы проверки эквивалентности программ.

В разделе 2.1 доказана теорема 2.1.11 о приведении любой последовательной программы к нормальной форме, в которой отсутствуют тупиковые и недостижимые точки, и в каждой точке программы, отличной от точек входа и выхода, выполняется только один базовый оператор.

Раздел 2.1 завершается описанием трех основных способов формальной спецификации динамических семантик (множества динамических моделей) пропозициональных программ — логического способа спецификации семантики при помощи формул пропозициональной динамической логики программ, алгебраического способа, позволяющего определять интерпретацию базовых операторов при помощи алгебраических тождеств, используемых в теориях групп и полугрупп, и теоретико-автоматного способа, использующего различные виды строковых и древесных автоматов для задания интерпретации базовых предикатов.

В **разделе 2.2** описана пропозициональная модель вычислений рекурсивных программ, предназначенная для решения задач анализа и преобразования программ в парадигме функционального программирования. Рекурсив-

ная программа состоит из запроса и конечного списка описаний процедур. Запрос — это конечная последовательность, состоящая из базовых операторов и вызовов процедур. Описание каждой процедуры — это оператор выбора **case**, каждая ветвь (альтернатива) которого представляет собой последовательность базовых операторов и вызовов процедур. Вызов каждой процедуры заключается в выборе в зависимости от значений базовых предикатов соответствующей альтернативы в описании процедуры и в подстановке этой альтернативы на место оператора вызова процедуры. Последовательность операторов, выполняемых при вызовах процедур, образует трассу в рекурсивной программе. Как и для последовательных программ, задачи семантического анализа рекурсивных программ могут быть сведены к задачам анализа трасс в рекурсивных программах. В дальнейшем основное внимание уделяется классу линейных рекурсивных программ, в которых каждая цепочка операторов в запросе и в альтернативах всех определений процедур содержит не более одного оператора вызова процедуры.

Так же, как и для последовательных программ, интерпретации базовых операторов и предикатов в рекурсивных программах задаются при помощи моделей пропозициональной динамической логики (структур Крипке). Подобно тому, как это было сделано в разделе 2.1 для последовательных программ, в разделе 2.2 определены понятия вычисления рекурсивной программы и результата вычисления, введены отношение эквивалентности рекурсивных программ в заданной динамической модели M и на множестве динамических моделей (в динамической семантике) σ, сформулирована задача проверки эквивалентности. Также представлен ряд утверждений о свойствах трасс в пропозициональных рекурсивных программах, позволяющих установить необходимые и достаточные условия эквивалентности рекурсивных программ в зависимости от устройства пар совместных трасс в сравниваемых программах. Как показано в разделе 6.3, новые методы проверки эквивалентности последовательных программ можно адаптировать для решения проблемы эквивалентности в некоторых классах рекурсивных программ.

В **разделе 2.3** исследована взаимосвязь между системой вычислений последовательных программ, введенной в разделе 2.1, и стандартными схемами программ — математической моделью императивных программ, определенной в рамках формализма логики предикатов первого порядка. Устройство интерпретаций языка пропозициональной динамической логики программ позволяет использовать хорошо изученные полугруппы конечных подстановок для вложения стандартных схем программ в пропозициональную модель вычислений последовательных программ. Как показано в разделе 6.4, предложенный способ вложения открывает новые возможности использования алгебраических особенностей полугрупп конечных подстановок для построения эффективных (в т. ч. полиномиальных по времени) алгоритмов проверки эквивалентности для некоторых классов стандартных схем программ.

Введенная в разделе 2.1 модель вычислений последовательных программ может применяться для решения задач анализа программ в разных парадигмах программирования. В заключительном **разделе 2.4.** второй главы диссертационной работы на основе этой модели вычислений введена пропозициональная модель потоковых программ, проводящих вычисления в режиме оперативного взаимодействия с внешней средой. Главная особенность потоковых программ состоит в том, что в их вычислениях нет обратной связи между состояниями управления (точками) программы и состояниями данных. Потоковые программы моделируют вычисления алгоритмов, работающих в оперативном режиме (режиме on-line). Трасса вычисления потоковой программы определяется последовательностью событий (запросов), которая не зависит от состояний данных, преобразуемых операторами программы в ответ на поступающие события. Предложенная модель потоковых программ имеет тесную связь с моделью вычислений обобщенных детерминированных конечных автоматов-преобразователей (transducers). Как показано в разделе 6.5, предложенные в главе 4 методы проверки эквивалентности последовательных программ применимы также и для решения проблемы эквивалентности потоковых программ.

В **третьей главе** приведен краткий обзор результатов исследований проблемы эквивалентности программ в различных моделях вычислений. Для того чтобы придать этому обзору целостность, автор счел возможным рассматривать известные ему результаты и достижения в изучении проблемы эквивалентности в хронологическом порядке. Почти шестидесятилетний период исследования этой проблемы, начинающийся с публикации статьи Райса, разделен на пять этапов. На каждом этапе выделены наиболее актуальные для своего времени (по мнению автора диссертации) задачи, перечислены основные результаты их исследований. История изучения проблемы эквивалентности программ прослеживается в контексте других задач теоретического и системного программирования, которые возникали и исследовались на каждом из выделенных этапов. Одно из предназначений этого обзора — обозначить место проблемы эквивалентности в ряду других задач семантического анализа программ, обосновать актуальность задачи разработки эффективных и практичных методов проверки эквивалентности программ в различных моделях вычислений и подчеркнуть новизну того метода ее решения, который предложен в диссертационной работе.

В **четвертой главе** представлены результаты исследования следующих вопросов, относящихся к задаче проверки аппроксимируемости динамических семантик пропозициональных последовательных программ.

1. К какому типу отношений порядка принадлежит отношение аппроксимации в различных классах динамических семантик пропозициональных программ?

2. В каких классах динамических семантик выполнимость требования $\sigma_1 \subseteq hom(\sigma_2)$ является не только достаточным, но необходимым условием аппроксимируемости семантик?

3. В каких случаях динамические семантики одного класса имеют наиболее точные аппроксимации в другом классе динамическеих семантик пропозициональных программ.

Изучению первого из поставленных вопросов посвящен **раздел 4.1**. В начале раздела приведены примеры равносильных динамических семантик пропозициональных программ, никакие две модели в которых не являются гомоморфными образами друг друга. Эти примеры показывают, что отношение аппроксимации между динамическими семантиками является нелокальным: его нельзя определить в терминах бинарных отношений между динамическими моделями, входящими в состав сравниваемых семантик.

Чтобы иметь возможность сравнивать динамические семантики программ и установить структуру отношения аппроксимации $\sqsubseteq$, выделена особая разновидность динамических семантик — насыщенные семантики. Динамическая семантика σ называется *насыщенной*, если для любой динамической модели M верно соотношение $M \notin \sigma \Rightarrow \sigma \not\simeq \sigma \cup \{M\}$. Каждый класс равносильности динамических семантик $[\sigma]_{\simeq}$ содержит единственную насыщенную семантику $\bigcup\limits_{\sigma' \simeq \sigma} \sigma'$. Для насыщенных динамических семантик справедлива

Теорема 4.1.9. *Класс насыщенных семантик программ с отношением аппроксимации $\sqsubseteq$ образует полную дистрибутивную решетку, в которой отношение аппроксимации совпадает с отношением гомоморфного сужения.*

Следствие 4.1.10. *Совокупность динамических семантик программ с отношением аппроксимации $\sqsubseteq$ образует квазирешетку.*

В **разделе 4.2** изучен вопрос о совпадении отношений аппроксимации и гомоморфного сужения для однородных динамических семантик, все модели в которых базируются на одной и той же шкале. Интерес к этому классу динамических семантик обусловлен тем, что 1) однородные динамические семантики описываются наиболее просто — интерпретация базовых операторов и допустимые оценки базовых предикатов специфицируются независимо друг от друга, и 2) именно однородные динамические семантики пропозициональных программ соответствуют свободным (эрбрановским) интерпретациям в теории стандартных (первопорядковых) схем программ и интерпрета-

циям в алгебраических моделях программ. В каждом классе равносильности однородных семантик выделяется максимальная по отношению теоретико-множественного включения однородно насыщенная семантика, выступающая в качестве канонического представителя этого класса равносильности. Но для однородно насыщенных семантик аналог теоремы 4.1.9 неверен.

Теорема 4.2.11. *Существуют такие однородно насыщенные семантики* $\sigma_1,\ \sigma_1 \subseteq \sigma(\mathcal{F}_1)$ *и* $\sigma_2,\ \sigma_2 \subseteq \sigma(\mathcal{F}_2)$, *что* $\sigma_1 \sqsubseteq \sigma_2$, *но неверно, что* $\sigma_1 \subseteq hom(\sigma_2)$.

Эта теорема свидетельствует о том, что класс однородных динамических семантик при всей простоте и естественности его устройства «алгебраически несовершенен». Вместе с тем, если шкала $\mathcal{F}$ обладает свойством *конечной представимости* (для любого состояния данных s множество $\mathcal{A}$-цепочек $\{h : [h]_{\mathcal{F}} = s\}$ конечно) или свойством *обратимости* (для любой $\mathcal{A}$-цепочки h существует такая цепочка h^{-1}, для которой верно равенство $[hh^{-1}]_{\mathcal{F}} = [\varepsilon]_{\mathcal{F}}$), то для любой пары однородно насыщенных семантик $\sigma,\ \sigma \subseteq \sigma(\mathcal{F})$ и $\sigma',\ \sigma' \subseteq \sigma(\mathcal{F}')$ верно соотношение $\sigma' \sqsubseteq \sigma \iff \sigma' \subseteq hom(\sigma)$ (теоремы 4.2.20 и 4.2.21).

В **разделе 4.3** исследована задача аппроксимации произвольных динамических семантик однородными динамическими семантиками. Введены понятия минимальной и наиболее точной аппроксимации заданной динамической семантики σ в заданном классе динамических семантик, и в теореме 4.3.1 показано, каким образом для каждой динамической семантики σ построить ее однородную аппроксимацию (стандартную аппроксимацию), которая является наименьшим однородным приближением динамической семантики σ по отношению гомоморфного сужения (утверждение 4.3.2). На основе понятия стандартной аппроксимации заданной динамической семантики σ предложены необходимые и достаточные условия аппроксимируемости произвольных динамических семантик однородными семантиками. При помощи этих утверждений установлены достаточные условия, при которых стандартная аппроксимация динамической семантики σ является ее наиболее точной аппроксимацией в классе однородных динамических семантик. Но справедливы также

Теорема 4.3.9. *Существуют такие динамическая семантики, для которых стандартная однородная аппроксимация не является наиболее точной однородной аппроксимацией.*

Теорема 4.3.10. *Существуют динамические семантики, не имеющие наиболее точной однородной аппроксимации*

На основании этих теорем показано (утверждения 4.3.11 и 4.3.12), что некоторые пары однородных семантик не имеют точных нижних и точных верхней граней по отношению аппроксимации $\sqsubseteq$. Тем самым было установлено, что квазиупорядоченное по отношению аппроксимации семейство однородных динамических семантик не является квазирешеткой, и в нем отношение аппроксимации отличается от отношения гомоморфного сужения.

В последующих двух главах диссертационной работы предложен, обоснован и опробован на многочисленных примерах новый подход к решению задачи проверки эквивалентности программ в различных динамических семантиках. Для того чтобы оценить сложность задачи проверки эквивалентности программ, предлагается воспользоваться следующей схемой построения разрешающих процедур:
1) установить условия совместности начальных трасс в программах, вычисления которых определяются в семантиках заданного класса;
2) на основе полученных условий совместности программных трасс выбрать подходящую разновидность машин (автоматов), позволяющих распознавать отношение равенства состояний данных, вычисляемых операторными цепочками, и тем самым проверять совместность пар программных трасс;
3) свести задачу проверки эквивалентности программ к задаче проверки пустоты языка (отношения), распознаваемого машинами из выбранного семейства, и оценить сложность проблемы пустоты;
4) выделить классы распознающих машин (и соответствующие им классы динамических семантик), для которых проблема пустоты имеет небольшую вычислительную сложность.

В основу этого метода положена следующая идея. Для проверки эквивалентности программ $\pi_1 \sim_\sigma \pi_2$ необходимо проверить, что каждая пара совместных в семантике σ трасс в программах π_1, π_2 имеет одинаковый результат. Но внимания заслуживают не вычисляемые этими программами состояния данных s_1 и s_2, а степень их отличия $\partial(s_1, s_2)$. Можно выбрать подходящую математическую структуру (алгебру, полугруппу, автомат) D, элементы которой выступают в роли оценок степени отличия между состояниями данных динамической шкалы $\mathcal{F} = \langle S, s_0, R\rangle$, потребовав при этом, чтобы в выбранной структуре D были определены

— интерпретация базовых операторов, обеспечивающая гомоморфное отображение $\mathcal{F} \times \mathcal{F}$ в D,

— подструктура D_0, которая гарантирует выполнимость для любых состояний данных s_1, s_2 соотношения $s_1 = s_2 \iff \partial(s_1, s_2) \in D_0$.

Тогда возникает возможность построить машину (автомат, размеченную систему переходов и др.) $K(\pi_1, \pi_2, D)$, которая воспроизводит $\mathcal{F}$-совместные вычисления программ π_1 и π_2, оперируя лишь с элементами структуры D. Выделенная подструктура D_0 обеспечивает проверку условий совместности программных трасс, а также наделяет систему $K(\pi_1, \pi_2, D)$ следующим важным свойством: программы π_1 и π_2 неэквивалентны на шкале $\mathcal{F}$ в том и только том случае, когда хотя бы один из прогонов машины $K(\pi_1, \pi_2, D)$ завершается вычислением элемента, не принадлежащего D_0. Таким образом, задача проверки эквивалентности программ π_1 и π_2 оказывается равносильной проблеме невычислимости (недостижимости) машиной $K(\pi_1, \pi_2, D)$ ни одного элемента из $D \setminus D_0$. Поскольку предлагаемый подход предусматривает построение вспомогательной машины, воспроизводящей пары совместных вычислений анализируемых программ, этот метод решения проблемы эквивалентности программ получил название *метод совместных вычислений*.

В **главе 5** описаны теоретико-автоматный и алгебраический варианты метода совместных вычислений проверки эквивалентности пропозициональных последовательных программ, семантика которых базируется на упорядочен-

ных динамических шкалах. Характерная особенность этих шкал состоит в том, что все базовые операторы выполняют на этих шкалах необратимые преобразования. Для упорядоченных шкал в качестве подходящих структур D, устанавливающих степень отличия состояний данных шкалы, могут быть выбраны детерминированные двухленточные автоматы и системы полугрупп специального вида — критериальные системы. Главное преимущество такого выбора состоит в том, что соответствующие системы $K(\pi_1, \pi_2, D)$ обладают полезным для разработки алгоритмов свойством: для произвольной пары $\mathcal{F}$-эквивалентных программ π_1 и π_2 множество состояний, достижимых системой $K(\pi_1, \pi_2, D)$, конечно. Во многих случаях размер этого множества состояний ограничен полиномом, зависящим от размеров программ π_1 и π_2, и это создает хорошие предпосылки для построения алгоритмов, разрешающих проблему эквивалентности программ за полиномиальное время.

В **разделе 5.1** введены детерминированные двухленточные односторонние машины (2-DM), распознающие в оперативном режиме (без использования маркеров конца слова) бинарные отношения на множестве $\mathcal{A}$-цепочек.

Определение 5.1.1. *Двухленточной детерминированной машиной* (2-DM) называется система $D = \langle \Sigma, Q_1, Q_2, q_0, F, \varphi \rangle$, состоящая из

- *входного алфавита* Σ,
- двух непересекающихся множеств *внутренних состояний* Q_1 и Q_2,
- *начального состояния* q_0,
- множества *допускающих состояний* F, $F \subseteq Q_1 \cup Q_2$, и
- частично определенной *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, $q \in Q_i$, $i = 1, 2$, она прочитывает очередную букву x (если таковая есть) слова w_i, помещенного на ленте i, и переходит в состояние $q' = \varphi(q, x)$. Пара слов (w_1, w_2) допускается машиной D, если она считывает оба слова w_1 и w_2, записанные на ее лентах, и оказывается по прочтении этой пары слов в допускающем состоянии.

Более формально поведение машины D определяется следующим образом. *Прогоном* 2-DM D называется последовательность пар

$$\alpha = (q_0, x_0), (q_1, x_1), \ldots, (q_n, x_n), \ldots$$

удовлетворяющая соотношениям q_0 — начальное состояние, $q_i \in Q_1 \cup Q_2$, $x_i \in \Sigma$ и $q_{i+1} = \varphi(q_i, x_i)$ для всех i, $i \geq 0$. Если эта последовательность оканчивается парой (q_n, x_n), то говорят, что прогон α достигает состояния $q' = \varphi(q_n, x_n)$. Для каждого $\delta \in \{1, 2\}$ можно выделить подпоследовательность

$$\alpha_\delta = (q_{i_1}, x_{i_1}), (q_{i_2}, x_{i_2}), \ldots, (q_{i_k}, x_{i_k}), \ldots \ ,$$

прогона α, состоящую из всех тех пар (q_{i_j}, x_{i_j}) последовательности α, которые удовлетворяют условию $q_{i_j} \in Q_\delta$. Тогда *δ-проекцией* прогона α называется слово $\alpha[\delta] = x_{i_1} x_{i_2} \ldots x_{i_k} \ldots$. Пара слов w_1, w_2 в алфавите Σ *допускается* машиной D, если существует прогон α, достигающий допускающего состояния q, $q \in F$, и проекциями этого прогона являются слова w_1, w_2. Запись E_D обозначает множество всех пар слов, допускаемых 2-DM D.

Считается, что 2-DM D описывает динамическую шкалу $\mathcal{F}$, если $E_D = \{(h_1, h_2) \ : \ [h_1]_\mathcal{F} = [h_2]_\mathcal{F}\}$.

Теорема 5.1.4. *Динамическая шкала $\mathcal{F}$ может быть описана некоторой 2-DM тогда и только тогда, когда $\mathcal{F}$ — упорядоченная шкала.*

Таким образом, выбранный класс машин пригоден для использования в качестве подходящих математических структур D, на которых можно вычислять оценки отличия состояний данных упорядоченных шкал.

В **разделе 5.2** описана концепция комбинированной машины $K(\pi_1, \pi_2, D)$ (определение 5.2.1), реализующая представленную выше идею метода совместных вычислений. Комбинированная машина $K(\pi_1, \pi_2, D)$ — это 2-DM, состоящая из трех взаимодействующих частей — программ π_1, π_2 и 2-DM D, описывающей ту шкалу $\mathcal{F}$, которая задает интерпретацию базовых операторов этих программ. На вход этой двухленточной машины подается пара последовательностей вида $(v_1, \Delta_1), (v_2, \Delta_2), \ldots$, записанных на ее лентах. Со-

стояния управления (мета-состояния) комбинированной машины $K(\pi_1, \pi_2, D)$ — это четверки вида $\langle v_1, v_2, q, Z\rangle$, где v_1, v_2 — точки в программах π_1 и π_2, q — это внутреннее состояние 2-DM D, а Z — это элемент множества $\mathcal{C}_{\mathcal{P}} \cup \{\perp\}$, используемый для проверки согласованности трасс, поступающих на вход комбинированной машины. Компоненты π_1, π_2 проверяют, являются ли последовательности, записанные на входных лентах, начальными трассами в программах π_1 и π_2. Компоненте D отводится роль синхронизатора, позволяющего прочитывать до конца только такие пары начальных трасс, которые совместны на шкале $\mathcal{F}$. Основным результатом раздела 5.2 является

Теорема 5.2.4. *Если 2-DM D описывает шкалу $\mathcal{F}$, то $\pi_1 \sim_{\mathcal{F}} \pi_2$ тогда и только тогда, когда комбинированная машина $K(\pi_1, \pi_2, D)$*
A: распознает пустое бинарное отношение $E_{K(\pi_1,\pi_2,D)}$ и
B: в каждом бесконечном прогоне бесконечно часто считывает данные на обеих лентах.

В **разделе 5.3** выявлены некоторые характерные особенности устройства комбинированных машин $K(\pi_1, \pi_2, D)$ для эквивалентных программ π_1 и π_2, проводящих вычисления на упорядоченной динамической шкале $\mathcal{F}$, описываемой 2-DM D. Основное внимание уделяется специальным мета-состояниям комбинированных машин — когерентным мета-состояниям. Мета-состояние $\langle v_1, v_2, q, Z\rangle$ комбинированной машины $K(\pi_1, \pi_2, D)$ называется *когерентным*, если q — это допускающее состояние управления 2-DM D. Особенности поведения комбинированных машин описывают две теоремы.

Теорема 5.3.5. *Если $\pi_1 \sim_{\mathcal{F}} \pi_2$, то любой бесконечный прогон комбинированной машины $K(\pi_1, \pi_2, D)$ проходит через когерентные мета-состояния бесконечно часто.*

Теорема 5.3.8. *Если $\pi_1 \sim_{\mathcal{F}} \pi_2$, и 2-DM D имеет N допускающих состояний, то комбинированная машина $K(\pi_1, \pi_2, D)$ на каждом участке длины $N(|\pi_1| + |\pi_2|)$ любого прогона проходит хотя бы через одно когерентное мета-состояние.*

Следствием этих теорем является основной результат раздела 5.3:

Теорема 5.3.9. *Если упорядоченная шкала $\mathcal{F}$ описывается 2-DM D, имеющей конечное множество F допускающих состояний, и программы π_1 и π_2 эквивалентны на шкале $\mathcal{F}$, то число мета-состояний комбинированной машины $K(\pi_1, \pi_2, D)$, достижимых из начальноого мета-состояния, ограничено величиной $2^{O(|F|(|\pi_1|+|\pi_2|))}$.*

В **разделе 5.4** приведены оценки сложности задачи проверки эквивалентности пропозициональных последовательных программ, семантика базовых операторов которых определяется упорядоченными динамическими шкалами, и описаны алгоритмы решения этой задачи. В основу алгоритмов, разрешающих отношение эквивалентности программ на упорядоченных шкалах, положены теорема 5.2.4, позволяющая свести проблему эквивалентности программ к задаче проверки пустоты комбинированной машины, и теорема 5.3.9, оценивающая размер комбинированной машины в зависимости от размера анализируемых программ и количества допускающих состояний 2-DM D, описывающей упорядоченную шкалу. Чтобы придать оценкам сложности наибольшую общность, алгоритмы проверки эквивалентности реализуются релятивизованными машинами Тьюринга, снабженными двумя оракулами E_D и U_D. Оракул E_D — это бинарное отношение на множестве $\mathcal{A}$-цепочек, распознаваемое 2-DM D, а U_D — это множество всех таких четверок $\mathcal{A}$-цепочек $(h'_1, h'_2, h''_1, h''_2)$, что по прочтении обеих пар (h'_1, h'_2) и (h''_1, h''_2) 2-DM D переходит в одно и то же состояние управления.

Теорема 5.4.1. *Если упорядоченная шкала $\mathcal{F}$ описывается 2-DM D с конечным множеством допускающих состояний, то задача проверки неэквивалентности программ $\pi_1 \not\sim_{\mathcal{F}} \pi_2$ принадлежит релятивизованному классу сложности NP^{E_D}.*

В частности, эта теорема верна для всякой полугрупповой упорядоченной шкалы, обладающей свойством левого сокращения, т. к. она может быть описана 2-DM с единственным допускающим состоянием.

Теорема 5.4.5. *Если шкала $\mathcal{F}$ может быть описана 2-DM с конечным множеством состояний (двухленточным детерминированным конечным автоматом), то задача проверки эквивалентности программ $\pi_1 \sim_{\mathcal{F}} \pi_2$ принадлежит классу сложности $NLOGSPACE$.*

Полинимиальные по времени алгоритмы проверки эквивалентности программ можно получить в тех случаях, когда на динамические шкалы и/или описывающие их 2-DM налагаются другие ограничения.

Пусть $W(n)$ — некоторая неубывающая функция натурального аргумента. Считается, что 2-DM D имеет *W-ограниченную ширину*, если для любой пары $\mathcal{A}$-цепочек h_1 и h_2, длина которых не превосходит n, множество состояний управления машины D, из которых она может достичь допускающего состояния при прочтении $\mathcal{A}$-цепочек h_1 и h_2, содержит не более $W(n)$ элементов.

Теорема 5.4.7. *Если шкала $\mathcal{F}$ описывается 2-DM D полиномиально ограниченной ширины, то задача проверки эквивалентности программ $\pi_1 \sim_{\mathcal{F}} \pi_2$ принадлежит релятивизованному классу сложности $PTIME^{E_D,U_D}$.*

Шкала $\mathcal{F} = \langle S, s_0, R\rangle$ называется *инъективной*, если интерпретация операторов $R(a,\cdot)$ является инъективным отображением для любого базового оператора $a, a \in \mathcal{A}$. В частности, упорядоченная полугрупповая шкала является инъективной тогда и только тогда, когда она обладает свойством правого сокращения.

Теорема 5.4.10. *Если инъективная шкала $\mathcal{F}$ описывается редуцированной 2-DM D, имеющей конечное множество допускающих состояний, то задача проверки эквивалентности программ $\pi_1 \sim_{\mathcal{F}} \pi_2$ принадлежит релятивизованному классу сложности $PTIME^{E_D,U_D}$.*

В **разделе 5.5** описан алгебраический вариант метода совместных вычислений. Здесь для оценки отличия состояний данных $\partial(s_1, s_2)$ полугрупповой шкалы $\mathcal{F}$ используются моноиды (полугруппы), образующие критериальную систему K. Четверка $K = \langle W, U, w^+, w^*\rangle$, состоящая из моноида W, в котором выделены подполугруппа U и пара элементов w^+, w^*, является *критериаль-*

ной системой для полугрупповой упорядоченной шкалы $\mathcal{F}$, если существует такой гомоморфизм $\varphi : \mathcal{F} \times \mathcal{F} \to U$, что для всякой пары состояний данных s_1, s_2 выполняется соотношение $s_1 = s_2 \iff w^+ * \varphi(\langle s_1, s_2 \rangle) * w^* = e$. На основе этой системы для пары проверяемых программ π_1 и π_2 строится размеченный корневой ориентированный граф совместных вычислений $\Gamma(\pi_1, \pi_2)$. Всякий маршрут, исходящий из корня этого графа, соответствует паре совместных вычислений программ π_1 и π_2 в некоторой $\mathcal{F}$-модели M. В графах совместных вычислений выделено особое множество опровергающих вершин.

Теорема 5.5.5. *Программы π_1 и π_2 эквивалентны на полугрупповой упорядоченной шкале $\mathcal{F}$ тогда и только тогда, когда из корня графа совместных вычислений $\Gamma(\pi_1, \pi_2)$ не достижима ни одна опровергающая вершина.*

Из этой теоремы следует, что в том случае, когда шкала $\mathcal{F}$ имеет конечную полугрупповую систему, графы совместных вычислений $\Gamma(\pi_1, \pi_2)$ конечны, и задача проверки эквивалентности программ $\pi_1 \sim_{\mathcal{F}} \pi_2$ принадлежит классу сложности NLOGSPACE. Но разрешающие алгоритмы полиномиальной сложности можно получить и для некоторых бесконечных критериальных систем. Критериальная система $K = \langle W, U, w^+, w^* \rangle$ для полугрупповой шкалы $\mathcal{F}$ называется *обратимой*, если для каждого элемента x из класса смежности $U * w^*$ существует не более одного элемента y из класса смежности $w^+ * U$, для которого выполняется равенство $y * x = e$. Тогда в том случае, когда полугрупповая упорядоченная шкала имеет обратимую критериальную систему K, в которой проблема тождеств разрешима за полиномиальное время, задача проверки эквивалентности программ $\pi_1 \sim_{\mathcal{F}} \pi_2$ принадлежит классу сложности PTIME (следствие 5.5.10).

Метод критериальных систем используется в главе 6 для построения алгоритмов проверки эквивалентности программ в других моделях вычислений — рекурсивных программ и стандартных (первопорядковых) программ.

В **разделе 5.6** обсуждаются приемы построения двухленточных машин и критериальных систем для полугрупповых шкал, обладающих определенными особенностями. В частности, показано, что

1. всякая полугрупповая шкала, вложимая в группу, может быть описана 2-DM 1-ограниченной ширины (утверждения 5.6.1 и 5.6.2);

2. всякая полугрупповая шкала, которая вложима в полугруппу, обладающую свойствами левого сокращения и наиболее общего левого унификатора, имеет обратимую критериальную систему (теорема 5.6.5).

Проверка эквивалентности программ существенно упрощается, если динамические шкалы, определяющие интерпретацию базовых операторов, обладают помимо свойства упорядоченности некоторыми дополнительными свойствами. В **разделе 5.7** рассмотрена задача проверки эквивалентности программ на уравновешенных динамических шкалах, характерная особенность которых состоит в том, что любые операторные цепочки, вычисляющие одно и то же состояние данных, обязаны иметь одинаковую длину. Показано, что уравновешенные шкалы могут быть описаны одноленточными машинами. Благодаря этой особенности, теоретико-автоматный и алгебраический варианты метода совместных вычислений могут быть существенно упрощены, а верхние оценки сложности задачи проверки эквивалентности программ на уравновешенных шкалах оказываются существенно меньше (в n^2 раз) аналогичных оценок сложности для проверки эквивалентности программ на упорядоченных (но не обязательно уравновешенных) шкалах, описываемых двухленточными машинами (теоремы 5.7.8 и 5.7.9).

Примеры некоторых одноленточных и двухленточных детерминированных машин, описывающих упорядоченные шкалы, представлены в **разделе 5.8**. В этих примерах рассмотрено устройство машин, описывающих упорядоченные шкалы, соответствующие
1) свободной полугруппе базовых операторов (пример 5.8.1);
2) свободной коммутативной полугруппе (пример 5.8.2);
3) полугруппе условно равносильных операторов (пример 5.8.3) с множеством определяющих соотношений вида $ab = ac$;
4) полугруппе подавляемых операторов (пример 5.8.4) с множеством определяющих соотношение вида $ab = b$.

Для всех указанных классов динамических шкал при помощи теоретико-автоматного варианта метода совместных вычислений удалось построить полиномиальные по времени алгоритмы проверки эквивалентности последовательных пропозициональных программ.

В **разделе 5.9** приведены примеры критериальных систем для некоторых классов полугрупповых шкал. В этих примерах описаны критериальные системы для уравновешенных полугрупповых шкал, соответствующих
1) свободной полугруппе базовых операторов (пример 5.9.1);
2) свободной коммутативной полугруппе базовых операторов (пример 5.9.2);
3) свободной частично коммутативной полугруппе базовых операторов (пример 5.9.3).

Область применения метода совместных вычислений, описанного в главе 5, не ограничивается проверкой эквивалентности лишь последовательных программ, оперирующих на упорядоченных шкалах. Как показано в **главе 6**, этот метод может быть модифицирован и применен для решения проблемы эквивалентности программ в других моделях вычислений.

В **разделе 6.1** исследована задача проверки эквивалентности пропозициональных последовательных программ в однородных семантиках $\sigma(\mathcal{F}, L)$ с определенными ограничениями на множества L допустимых функций оценки базовых предикатов. Вначале рассмотрены однородные динамические семантики $\sigma(\mathcal{F}, L)$, базирующиеся на упорядоченных шкалах, в которых множества L допустимых функций оценки специфицируются конечными детерминированными автоматами-преобразователями, определенным образом согласованными со шкалами $\mathcal{F}$. Однородные динамические семантики такого вида получили название *автоматных семантик*. Показано, что для проверки эквивалентности программ $\pi_1 \sim_{\sigma(\mathcal{F},L)} \pi_2$ в автоматной семантике $\sigma(\mathcal{F}, L)$, базирующейся на упорядоченной шкале $\mathcal{F}$, можно ввести комбинированные двухленточные детерминированные машины, аналогичные тем, которые были введены в разделе 5.2. Для введенных таким образом комбинированных машин оказывается справедливым утверждение, аналогичное теореме 5.2.4, а

также все те вытекающие из нее утверждения, представленные в разделе 5.3, на основании которых в разделе 5.4 были разработаны эффективные алгоритмы проверки эквивалентности последовательных пропозициональных программ. Таким образом, удалось выделить классы однородных семантик $\sigma(\mathcal{F}, L)$, базирующихся на упорядоченных шкалах, для которых задача проверки эквивалентности программ $\pi_1 \sim_{\sigma(\mathcal{F},L)} \pi_2$ может быть решена за счет сравнительно простой модификации метода совместных вычислений. Однако для некоторых других однородных семантик применение предложенного подхода требует внесения в него более существенных изменений. В качестве примера в разделе 6.1 исследована проблема эквивалентности программ в однородной динамической семантике с перестановочными операторами и монотонными функциями оценки предикатов. Для решения этой задачи предложен модифицированный вариант метода совместных вычислений, при помощи которого построен полиномиальный по времени разрешающий алгоритм.

Теорема 6.1.11. *Задача проверки эквивалентности программ π_1 и π_2 в однородной динамической семантике с перестановочными операторами и монотонными функциями оценки предикатами разрешима за время, оцениваемое величиной $n^{O(|\mathcal{A}||\mathcal{P}|)}$, где $n = \max(|\pi_1|, |\pi_2|)$.*

В **разделе 6.2** алгебраический вариант метода совместных вычислений распространен на рекурсивные программы. Показано, что для линейных рекурсивных программ проблема эквивалентности $\pi_1 \sim_{\mathcal{F}} \pi_2$ может быть сведена к проблеме достижимости опровергающих вершин в графе совместных вычислений $\Gamma(\pi_1, \pi_2)$, построенном на основе критериальной системы $K = \langle W, U, w^+, w^* \rangle$ для упорядоченной полугрупповой шкалы $\mathcal{F}$ (теорема 6.2.7), подобно тому, как это было осуществлено в разделе 5.6 для последовательных программ. Если критериальная система K обладает дополнительным свойством сильной обратимости, которое проявляется в однозначной разрешимости уравнений вида $x_1 * y = e$ и $y * x_2 = e$ для любых элементов x_1, x_2 из классов смежности $w^+ * U$ и $U * w^*$ соответственно, то справедлива

Теорема 6.2.11. *Если полугрупповая уравновешенная шкала $\mathcal{F}$ имеет критериальную систему $K = \langle W, U, w^+, w^* \rangle$, обладающую свойством сильной обратимости, и в моноиде W проблема тождества разрешима за полиномиальное время, то проблема эквивалентности линейных рекурсивных программ на шкале $\mathcal{F}$ принадлежит классу сложности $co - NP$.*

Приведены примеры 6.2.1 и 6.2.2, иллюстрирующие применение метода совместных вычислений для проверки эквивалентности рекурсивных программ на некоторых шкалах. Для того чтобы уменьшить сложность разрешающего алгоритма, на критериальную систему K приходится налагать дополнительные ограничения: требуется, чтобы критериальная система $K = \langle W,U, w^+, w^* \rangle$ для упорядоченной полугрупповой шкалы базировалась на разрешимой группе W. В этом случае справедлива

Теорема 6.2.16. *Если полугрупповая уравновешенная шкала $\mathcal{F}$ имеет критериальную систему $K = \langle W, U, w^+, w^* \rangle$, где W — конечно порожденная группа, в которой проблема равенства слов разрешима за время, полиномиальное относительно длин слов, то задача проверки эквивалентности линейных рекурсивных программ на шкале $\mathcal{F}$ разрешима за время, полиномиальное относительно размеров программ.*

Область применения автоматного варианта метода совместных вычислений, сводящего проблему эквивалентности программ к проблеме пустоты многоленточных автоматов, не ограничивается лишь семантиками, базирующимися на упорядоченных шкалах. В **разделе 6.3** рассматриваются последовательные программы с операторами сброса (эти операторы также называют операторами засылки констант или операторами выбора режима вычисления). Семантика таких программ определяется неупорядоченными шкалами, которые соответствуют полугруппам с правыми единицами. Показано, что задача проверки эквивалентности рассматриваемых программ сводится к проблеме пустоты конечных недетерминированных автоматов. Однако число состояний этих конечных автоматов может оказаться величиной, экспоненциально зависящей от размеров анализируемых программ. Установлено, что

задача проверки эквивалентности программ с операторами выбора режима вычисления является PSPACE-полной алгоритмической проблемой (теоремы 6.3.8 и 6.3.9). На основе этого результата показано, что и для многих других неупорядоченных шкал задача проверки эквивалентности программ является PSPACE-трудной задачей (теорема 6.3.10).

В разделе **разделе 6.4** исследована проблема эквивалентности стандартных схем программ. Как показано в разделе 1.3, эта проблема может быть сведена к задаче проверки эквивалентности пропозициональных последовательных программ, семантика которых определяется на основе полугруппы конечных подстановок. Поэтому метод совместных вычислений применим для построения алгоритмов проверки эквивалентности в модели вычислений стандартных схем программ. На основе этого метода выделен новый класс стандартных схем программ с разрешимой проблемой эквивалентности — класс *ортогональных схем программ.*

Два терма считаются *ортогональными*, если ни один из них не является подтермом другого. Подствновка $\theta : Var \to Term$ называется ортогональной, если 1) она не является переименованием, 2) все термы $\theta(x)$ не являются основными термами, и 3) для любой пары различных переменных x, y термы $\theta(x), \theta(y)$ ортогональны. В ортогональных схемах программ любая подстановка $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$, которая соответствует последовательной композиции операторов присваивания линейного участка программы, является ортогональной. Показано, что множество ортогональных подстановок с операцией композицией, пополненное тождественной подстановкой ε, образует моноид (теорема 6.4.3), упорядоченный (следствие 6.4.5) а также обладающий свойствами левого сокращения (следствие 6.4.7) и наиболее общего левого унификатора (утверждение 6.4.11). Таким образом, на основании теорем 5.4.7 и 5.6.5 справедлива

Теорема 6.4.12 *Задача проверки эквивалентности ортогональных стандартных схем программ разрешима за полиномиальное время.*

В последнем **разделе 6.5** шестой главы показано, как можно адаптировать метод совместных вычислений для проверки эквивалентности потоковых программ. Главная особенность потоковых программ, отличающая их от рассмотренных ранее вычислительных моделей последовательных и рекурсивных программ, состоит в том, что в вычислениях потоковых программ нет обратной связи между последовательностями состояний управления (точками) программы и состояний данных. Трасса вычисления такой программы определяется последовательностью событий (запросов), которая не зависит от состояний данных, преобразуемых операторами программы в ответ на поступающие события. Поэтому для описания динамических шкал можно использовать одноленточные детерминированные машины (1-DM). В то же время, для потоковых программ неверна теорема 2.1.5 о приведении произвольной программы к нормальной форме, и на каждом шаге вычисления потоковая программа может выполнять не одно базовое действие, а некоторую конечную последовательность операций. Поэтому комбинированные машины для потоковых программ вынуждены обрабатывать на каждом отдельном шаге своего прогона не отдельные буквы (базовые операторы), а конечные слова ($\mathcal{A}$-цепочки). Для потоковых комбинированных машин доказана теорема 6.5.2 о сводимости проблемы эквивалентности потоковых программ π_1, π_2 на произвольной шкале $\mathcal{F}$, описываемой 1-DM D, к проблеме пустоты соответствующей потоковой комбинированной машины $K(\pi_1, \pi_2, D)$. Также установлены достаточные условия (теорема 6.5.4) разрешимости проблемы пустоты потоковых комбинированных машин за полиномиальное время. На основании этих условий доказана теорема

Теорема 6.5.6 *Если полугрупповая шкала $\mathcal{F}$ вложима в конечно порожденную группу, в которой проблема тождества $w_1 = w_2$ разрешима за время $t(n)$, то задача проверки эквивалентности потоковых программ $\pi_1 \sim_{\mathcal{F}} \pi_2$ разрешима за время $O(n^2 t(n^2))$, где $n = \max(|\pi_1|, |\pi_2|)$.*

В заключительной **главе 7** перечислены основные результаты, полученные в диссертационной работе.

Основные результаты

1. Для динамических семантик — множеств моделей пропозициональной динамической логики, — задающих интерпретацию операторов и предикатов последовательных, рекурсивных и потоковых программ, установлены необходимые условия и достаточные условия выполнимости отношения аппроксимации в различных классах динамических семантик.

2. Установлен тип отношения аппроксимации в различных классах динамических семантик пропозициональных программ.

3. Установлены достаточные условия существования для заданной динамической семантики ее наиболее точной аппроксимации в классе однородных динамических семантик.

4. Для решения проблемы эквивалентности последовательных программ предложен новый подход — метод совместных вычислений, позволяющий, используя теоретико-автоматное или алгебраическое описание интерпретации программных операторов, сводить задачу проверки эквивалентности программ к проблеме пустоты для детерминированных двухленточных односторонних машин.

5. На основе метода совместных вычислений установлены достаточные условия, которым должна удовлетворять динамическая семантика, задающая интерпретацию операторов программ, для того чтобы задача проверки эквивалентности последовательных программ принадлежала классу сложности co-NP.

6. На основе метода совместных вычислений установлены достаточные условия, которым должна удовлетворять динамическая семантика, задающая интерпретацию операторов программ, для того чтобы задача проверки эквивалентности последовательных программ принадлежала классам сложности NLOGSPACE или PTIME.

7. Применением метода совместных вычислений доказана разрешимость за полиномиальное время задачи проверки эквивалентности последовательных программ, операторы которых обладают определенными алгебраическими свойствами, включая свойства перестановочности, условной эквивалентности и подавления.

8. При помощи метода совместных вычислений доказана разрешимость за полиномиальное время проблемы эквивалентности последовательных программ в некоторых динамических семантиках с ограничениями на допустимые интерпретации (оценки) базовых предикатов, используемых в программах.

9. Предложена модификация метода совместных вычислений, при помощи которой доказана PSPACE-полнота задачи проверки эквивалентности последовательных программ с операторами засылки констант.

10. Методом совместных вычислений установлены достаточные условия разрешимости задачи проверки эквивалентности унарных линейных рекурсивных программ и выделены классы динамических семантик, в которых указанная задача принадлежит классу сложности PTIME.

11. При помощи метода совместных вычислений выделен класс стандартных схем программ, в котором задача проверки эквивалентности принадлежит классу сложности PTIME.

12. При помощи метода совместных вычислений установлены достаточные условия разрешимости задачи проверки эквивалентности потоковых программ (обобщенных конечных автоматов-преобразователей) и выделены классы динамических семантик, в которых указанная задача принадлежит классу сложности PTIME.

Публикации по теме диссертации

1. Захаров В.А. Схемы Янова с автоматными сдвигами // Тезисы докладов VIII Международной конференции «Проблемы теоретической кибернетики». — 1988. — с. 101–102.

2. **Захаров В.А. Автоматные модели программ // Доклады АН СССР. — 1989. — т.309, N 1, — с. 24-27.**

3. Захаров В.А. Условия свободной схемы в формальных моделях программ // Тезисы докладов IX Международной конференции «Проблемы теоретической кибернетики». — 1991. — с. 94-96.

4. **Захаров В.А. Формальные модели и свободные схемы программ // Программирование. — 1992. — N 2. — с. 10-24.**

5. **Захаров В.А. Об одном критерии сравнения операторных формальных моделей // Программирование. — 1993. — N 4. — с. 12-25.**

6. Захаров В.А. Об одном типе эквивалентности схем программ // Методы и системы технической диагностики, Саратовский государственный университет. — 1993. — вып.18. — с.68-70.

7. **Захаров В.А. Об отношении аппроксимируемости семантик операторных программ // Вестник Московского университета. — Серия 15, вычислительная математика и кибернетика. — 1994. — N 3. — с. 54-60.**

8. **Захаров В.А. О свободных схемах в формальных моделях программ // Математические вопросы кибернетики, Вып. 5. — М.:Наука, 1994. — с. 208-239.**

9. **Захаров В.А. Условия сглаживаемости операторных формальных моделей программ // Программирование. — 1994. — N 5. — с. 23-40.**

10. Захаров В.А. Эквивалентные преобразования схем программ в моделях, порожденных формулами динамической логики // Материалы XI международной конференции "Логика, методология, философия науки Институт философии РАН. — 1995 — т. II. — с. 137-142.

11. Захаров В.А., Подловченко Р.И. Полиномиальный алгоритм разрешения эквивалентности схем программ // Тезисы докладов XI Международной конференции «Проблемы теоретической кибернетики». — 1996. — с. 68-69.

12. 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.

13. Захаров В.А. Полиномиальный алгоритм разрешения проблемы эквивалентности унарных линейных рекурсивных схем программ // Труды II Международной конференции «Дискретные модели в теории управляющих систем». — 1997. — с. 26-29.

14. **Подловченко Р.И., Захаров В.А. Полиномиальный по сложности алгоритм, распознающий коммутативную эквивалентность схем программ // Доклады РАН, серия Информатика. — 1998. — т. 362, N 6. — с. 27-31.**

15. **Захаров В.А. Быстрые алгоритмы разрешения эквивалентности операторных программ на уравновешенных шкалах // Математические вопросы кибернетики, вып.7. — М.:Физматлит, 1998. — с. 303-324.**

16. Захаров В.А. О проблеме эквивалентности операторных схем на упорядоченных полугрупповых моделях // Сборник трудов III Международной конференции «Дискретные модели в теории управляющих систем», Красновидово-98. — 1998. — с. 36-40.

17. **Захаров В.А. Аппроксимация абстрактных семантик формальными моделями программ // Дискретная математика. — 1998. — т. 10, вып. 4. — с. 119-141.**

18. Захаров В.А. О проблеме эквивалентности пропозициональных программ над полугруппами // Kurosh Algebraic Conference'98, Abstracts of Talks. — 1998. — с. 170-172.

19. Zakharov V.A.. An efficient and unified approach to the decidability of equivalence of propositional program schemes // Lecture Notes in Computer Science. — 1998. — v. 1443. — p. 247–258.

20. Захаров В.А. Об одном критерии сравнимости формальных моделей программ // Сборник трудов семинара по дискретной математики и ее приложениям, (2-4 февраля, 1998 г.), М.: Изд-во механико-математического факультета МГУ. — 1998. — с. 107-109.

21. **Захаров В.А. Быстрые алгоритмы разрешения эквивалентности пропозициональных операторных программ на упорядоченных полугрупповых шкалах // Вестник Московского университета, сер. 15, Вычислительная математика и кибернетика. — 1999, N 3. — с. 29-35.**

22. **Захаров В.А. Об эффективной разрешимости проблемы эквивалентности линейных унарных рекурсивных программ // Математические вопросы кибернетики, вып. 8. — М.:Наука, 1999. — с. 255-273.**

23. Захаров В.А. О разрешимости проблемы эквивалентности в одном классе операторных программ // Сборник "Прикладная математика и информатика вып. 5. — Изд-во ВМиК МГУ, 1999. — с. 90-100.

24. Zakharov V.A. On the decidability of the equivalence problem for orthogonal sequential programs // Grammars. — 1999. — v. 2, N 3. — p. 271-281.

25. Захаров В.А. О проблеме эквивалентности унарных металинейных рекурсивных схем // Тезисы докладов XII международной конференции «Проблемы теоретической кибернетики», Часть 1. — 1999. — с. 78.

26. Захаров В.А. Общие методы построения разрешающих алгоритмов для проблемы эквивалентности пропозициональных операторных программ // Сборник трудов IV Международной конференции «Дискретные модели в теории управляющих систем», Красновидово-00. — 2000. — с. 25-29.

27. Захаров В.А., Соколова К.А.. О разрешимости проблемы эквивалентности в одном классе металинейных унарных рекурсивных программ // Сборник трудов IV Международной конференции «Дискретные модели в теории управляющих систем», Красновидово-00. — 2000. — с. 29-31.

28. Захаров В.А. О проблеме эквивалентности для схем программ с операторами засылки констант // Сборник трудов IV Международной конференции «Дискретные модели в теории управляющих систем», Красновидово-00. — 2000. — с. 153-154.

29. Захаров В.А., Кузюрин Н.Н., Холодов А.Н., Шабанов Л.В., Шокуров А.В. Эффективные алгоритмы и их программные реализации // Труды Института Системного программирования: Том 1. — М.:ИСП РАН, 2000. — с. 115-124.

30. Zakharov V.A.. On the decidability of the equivalence problem for monadic recursive programs // Theoretical Informatics and Applications. — 2000. — v. 34, N 2. — p. 157-171.

31. Zakharov V.A. On the approximation relation on dynamic logic models // Abstracts of contributed papers, Logic Colloquium 2000, Paris, La Sorbonne, 23-31 juillet 2000. — 2000.

32. Захаров В.А. О проблеме эквивалентности операторных программ на одном классе уравновешенных шкал // Материалы VII Международного

семинара «Дискретная математика и ее приложения», Изд-во механико-математического ф-та МГУ. — 2001. — с. 54-57.

33. **Захаров В.А. О проблеме эквивалентности операторных программ на уравновешенных однородных обратимых шкалах // Математические вопросы кибернетики. Вып. 10. — Физматлит, 2001. — с. 155-166.**

34. Zakharov V.A.. The equivalence problem for computational models: decidable and undecidable cases // Lecture Notes in Computer Science. — 2001. — v. 2055. — p. 133–153.

35. Захаров В.А. Вычисление инвариантов последовательных программ // Тезисы докладов XIII Международной конференции «Проблемы теоретической кибернетики», Часть 1. — 2002. — с. 68.

36. Захаров В.А., Захарьящев И.М. Об одной полисемантической модели последовательных программ // Труды V Международной конференции «Дискретные модели в теории управляющих систем», (Ратмино, 26-29 мая 2003 г.). — 2003. — с. 33-34.

37. Zakharov V.A., Zakharyaschev I.M. An equivalence-checking algorithm for polysemantic models of sequential programs // Proceedings of the International Workshop on Program Understanding (14-16 July, Altai Mountains, Russia). — 2003. — p. 59–70.

38. Захаров В.А. Об одной алгебраической модели программ, связанной с обработкой прерываний // Материалы VIII Международного семинара «Дискретная математика и ее приложения», Изд-во механико-математического ф-та МГУ. — 2004. — с. 129-131.

39. Захаров В.А., Захарьящев И.М. О сложности проблемы эквивалентности в модели программ с перестановочными и монотонными операторами // Материалы VIII Международного семинара "Дискретная математика и

ее приложения Изд-во механико-математического ф-та МГУ. — 2004. — с. 131-134.

40. Захаров В.А., Захарьящев И.М. О проблеме эквивалентности для программ с частично перестановочными и монотонными операторами // Труды VI-ой Международной конференции «Дискретные модели в теории управляющих систем», (7-11 декабря 2004 г., Москва). — 2004. — с. 105-110.

41. Zakharov V.A., Zakharyaschev I.M. On the equivalence-checking problem for polysemantic models of sequential programs // Труды Института Системного программирования: Том 6. – М.:ИСП РАН., 2004. — с. 182-199.

42. Zakharov V.A., Zakharyaschev I.M. On the equivalence-checking problem for sequential programs with partially commuting and monotonic statements // Proceedings of the XI Congress of Mathematics of Serbia and Montenegro (September 28-October 2, 2004), Petrovac, Montenegro. — 2004. — p. 79.

43. Захаров В.А., Подловченко Р.И. Проверка эквивалентности программ: модели и алгоритмы // Тезисы докладов XIV Международной конференции «Проблемы теоретической кибернетики», Пенза, 23-28 мая, 2005. — 2005.

44. Захаров В.А., Подловченко Р.И., И.М. Захарьящев, Д.М. Русаков, В.Л. Щербина. О возможности применения быстрых алгоритмов проверки эквивалентности программ для обнаружения вирусов // Труды 2-ой Всероссийской научной конференции «Методы и средства обработки информации», Москва, 2005. — 2005. — с. 414-421.

45. Zakharov V.A., Zakharyaschev I.M. On the equivalence checking problem for a model of programs related with muti-tape automata // Lecture Notes in Computer Science. — 2005. — v. 3317. — p. 293—305.

46. Podlovchenko R.I., Rusakov D.M., Zakharov V.A.. On the equivalence problem for programs with mode switching // Lecture Notes in Computer Science.

— 2006. — v. 3845 — p. 351–352.

47. Захаров В.А., Щербина В.Л. О сложности распознавания эквивалентности машин Тьюринга без записи на ленту // Материалы XIV международной школы-семинара «Синтез и сложность управляющих систем» (Санкт-Петербург, 26-30 июня 2006 г.). — 2006. — с. 147-150.

48. Варновский Н.П., Захаров В.А., Кузюрин Н.Н., Шокуров А.В., Подловченко Р.И., Щербина В.Л. О применении методов деобфускации программ для обнаружения сложных компьютерных вирусов // Известия ТРТУ, Таганрог, Изд-во ТРГУ. — 2006. — с. 53-57.

49. Podlovchenko R.I., Rusakov D. M., Zakharov V.A. The equivalence problem for programs with mode switching is PSPACE-complete // Труды Института Системного программирования: Том 11. — М.:ИСП РАН, 2006. — с. 111-135.

50. **Варновский Н.П., Захаров В.А., Кузюрин Н.Н., Шокуров А.В. Современные методы обфускации программ: сравнительный анализ и классификация // Известия ЮФУ. — 2007. — N 1. — с. 93-99.**

51. Захаров В.А., Щербина В.Л. Об эквивалентности программ с операторами, обладающими свойствами коммутативности и подавления // Материалы 9-го Международного семинара «Дискретная математика и ее приложения», Москва, 2007. — 2007. — с. 191-194.

52. Zakharov V.A., Kuzurin N.N., Podlovchenko R.I., Shcherbina V.V. Using algebraic models of programs for detecting metamorphic malwares // Труды Института Системного программирования: Том 12. — М.:ИСП РАН, 2007. — с. 77-94.

53. **Захаров В.А., Щербина В.Л. Эффективные алгоритмы проверки эквивалентности программ в моделях, связанных с обработкой прерываний // Вестник Московского университета,**

сер. 15, Вычислительная математика и кибернетика. — 2008, N 2. — с. 33-41.

54. Захаров В.А. О проблеме эквивалентности в одном классе монадических линейных рекурсивных программ // Тезисы докладов XV-ой международной конференции «Проблемы теоретической кибернетики» (Казань, 2-7 июня, 2008 г.) — 2008. — с. 40.

55. Захаров В.А., Щербина В.Л. О сложности проверки эквивалентности программ с операторами засылки констант // Труды VIII-ой Международной конференции «Дискретные модели в теории управляющих систем», Москва, 6-9 апреля 2009 г. — 2009. — с. 369-374.

56. Zakharov V.A. Two-tape machinery for the equivalence checking of sequential programs // Proceedings of the International Workshop on Program Understanding, Novosibirsk. — 2009. — p. 28-40.

57. Захаров В.А. Проверка эквивалентности программ при помощи двухленточных автоматов // Кибернетика и системный анализ. — 2010. — N 4. — с. 39-48.

58. **Подымов В.В., Захаров В.А. Об одной полугрупповой модели программ, определяемой при помощи двухленточных автоматов // Научные ведомости Белгородского государственного университета, Серия История. Политология. Экономика. Информатика. — 2010. — вып. 14/1, N 7. — с. 94-101.**

59. Zakharov V.A. Equivalence checking of sequential programs using two-tape automata // International Workshop "Automata, algorithms, and information technologies". Abstracts. Kiev, May 19-21, 2010. — 2010. — с. 25.

60. Подымов В.В., Захаров В.А. О двухленточных машинах, описывающих полугруппы с сокращением // Материалы 16-й Международной конференции "Проблемы теоретической кибернетики Нижний Новгород, 20-25 июня 2011. — 2011. — с. 372-375.