

*На правах рукописи*

Кузьмин Егор Владимирович

**АЛГОРИТМИЧЕСКИЕ СВОЙСТВА  
ФОРМАЛЬНЫХ МОДЕЛЕЙ  
ПАРАЛЛЕЛЬНЫХ И РАСПРЕДЕЛЕННЫХ СИСТЕМ**

Специальность 05.13.17 —  
Теоретические основы информатики

**А В Т О Р Е Ф Е Р А Т**

диссертации на соискание ученой степени  
доктора физико-математических наук

Ярославль — 2010

Работа выполнена на кафедре теоретической информатики Ярославского государственного университета им. П. Г. Демидова.

Научный консультант:

доктор физико-математических наук, профессор  
Соколов Валерий Анатольевич

Официальные оппоненты:

доктор физико-математических наук, доцент  
Дехтярь Михаил Иосифович

доктор физико-математических наук, профессор  
Дурнев Валерий Георгиевич

доктор физико-математических наук, профессор  
Солон Борис Яковлевич

Ведущая организация:

Учреждение Российской академии наук  
Институт системного программирования РАН

Защита состоится 27 мая 2011 г. в 14 час. на заседании диссертационного совета ДМ002.084.01 при Учреждении Российской академии наук Институте программных систем им. А. К. Айламазяна РАН, расположенном по адресу: 152021, Ярославская область, Переславский район, с. Веськово, ул. Петра I, д. 4а.

С диссертацией можно ознакомиться в библиотеке Учреждения Российской академии наук Института программных систем им. А. К. Айламазяна РАН.

Автореферат разослан 5 апреля 2011 г.

Ученый секретарь  
диссертационного совета, к. т. н.

Пономарева С. М.

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность работы.** Современный период развития информатики и вычислительной техники характеризуется широким использованием параллельных и распределенных систем (вычислительные машины и комплексы с параллельной и распределенной архитектурой, параллельные программы, протоколы передачи данных, модели технологических и бизнес-процессов), поведение которых отличается высокой степенью сложности. Это обстоятельство выдвигает новые задачи в области моделирования и анализа корректности таких систем. Под корректностью понимается полное соответствие системы задачам, для которых она создаётся. Корректность определяется абстрактным образом в соответствии с формальной спецификацией, описывающей желаемое поведение системы. Процесс проверки соответствия поведения системы требованиям, заданным в спецификации, называется *верификацией*.

За время исследований по проблеме верификации был разработан ряд методов проверки корректности параллельных и распределенных систем и накоплено большое количество теоретических результатов и практического опыта в этой области. Среди отечественных исследований по спецификации, моделированию и анализу распределенных систем отметим работы Н. А. Анисимова, О. Л. Бандман, И. Б. Вирбицкайте, В. В. Воеводина, М. И. Дехтяря, В. А. Захарова, Ю. Г. Карпова, В. Е. Котова, И. А. Ломазовой, В. А. Непомнящего, Р. Л. Смелянского, В. А. Соколова, Л. А. Черкасовой, Н. В. Шилова.

Обычно в качестве формальных моделей параллельных и распределенных систем выступают помеченные системы переходов, представляющие собой средство технически простое, но очень удобное и достаточно общее для моделирования параллельного поведения.

Многочисленные методы и средства анализа параллельных и распределенных систем основаны на использовании помеченных систем переходов с *конечным* числом состояний. Однако существует большое количество примеров формальных моделей, которые могут быть рассмотрены как системы переходов с *бесконечным* числом состояний. В этом случае многие средства верификации, производящие полный перебор пространства состояний, становятся неприменимыми, так как по своей природе не способны анализировать системы, у которых число состояний бесконечно.

Для того чтобы преодолеть этот недостаток, были разработаны методы, применимые, по крайней мере, для некоторых ограниченных классов систем с бесконечным числом состояний. Можно упомянуть здесь работы, например, П. Абдуллы, К. Чёранса, А. Финкеля, Б. Йонссона, Ф. Моллера, Ф. Шнеблена. Более того, оказалось, что метод проверки модели, широко используемый при автоматической верификации систем с конечным числом состояний, может быть применен для некоторых классов систем с бесконечным числом состояний и подмножеств темпоральных логик.

Проверка модели (model checking) — один из подходов к решению проблемы верификации. В качестве языков спецификации для выражения свойств систем при этом подходе используются темпоральные логики. Задача проверки модели состоит в определении выполнимости для системы, заданной формальным образом (в виде формальной модели), свойства, представленного формулой темпоральной логики.

Исследования систем переходов с бесконечным числом состояний были мотивированы теорией формальных языков и грамматик. Во-первых, в этой теории бесконечные языки описываются грамматиками (т. е. имеют конечное представление), а во-вторых, некоторые проблемы для языков, например, проблема равенства регулярных языков, являются разрешимыми. Следовательно, не все проблемы систем переходов с бесконечным числом состояний неразрешимы. По аналогии с теорией формальных языков были

введены новые формализмы для описания бесконечных систем переходов.

Примерами формальных моделей (систем переходов с бесконечным числом состояний), позволяющих описывать параллельные и распределенные системы, являются сети Петри, базовые параллельные процессы (BPP — Basic Parallel Processes), системы с каналами, теряющими сообщения (LCS — Lossy Channel Systems) и автоматы реального времени (Real-Time Automata).

Указанные формализмы, как и многие другие, могут быть рассмотрены как *вполне структурированные системы помеченных переходов*.

Вполне структурированные системы помеченных переходов — это весьма широкий класс систем переходов с бесконечным числом состояний, для которых разрешимость многих свойств следует из существования совместимого (по возрастанию или по убыванию) правильного квазипорядка с отношением переходов, введенных на множестве состояний.

Описание свойств и методов исследования этого класса систем переходов является одной из целей данной работы. Основное внимание уделяется разрешимости таких проблем, как проблемы ограниченности, достижимости, покрытия, неизбежности, поддержки управляющего состояния, останова, эквивалентности и ряда других важных семантических свойств. Важность этих задач при анализе свойств систем с бесконечным числом состояний можно продемонстрировать на примере проблемы ограниченности. Одним из первых вопросов при верификации потенциально бесконечных систем является вопрос, действительно ли поведение той или иной системы порождает бесконечное число состояний. Если ответ отрицательный, то могут быть применимы методы, которые используются для верификации конечных систем. Если же множество состояний действительно бесконечно, остается работать лишь с ограниченной, «видимой» частью поведения системы с применением различных комбинаций методов и техник анализа.

В качестве общего средства для демонстрации неразрешимых проблем для вполне структурированных систем переходов

рассматриваются «слабые» счетчиковые машины. Абстрактная счетчиковая машина — довольно простой и удобный объект для исследования неразрешимых семантических и темпоральных свойств систем переходов. Большое количество формализмов, порождающих вполне структурированные системы переходов, с легкостью могут задавать поведение «слабых» счетчиковых машин. Но в то же время такие проблемы, как ограниченность и достижимость, для некоторых классов машин являются неразрешимыми. Поэтому все неразрешимые проблемы для того или иного класса «слабых» счетчиковых машин автоматически распространяются на вполне структурированные системы переходов (формальные модели), способные отображать поведение этих машин.

Абстрактные счетчиковые машины строятся с помощью различных ослаблений, например, (безусловного) недетерминизма переходов и отношения потери значений счетчиков, на основе счетчиковых машин Минского, которые являются равносильными машинам Тьюринга. Отсюда неразрешимость исследуемой проблемы устанавливается методом сведения неразрешимой проблемы машины Минского к данной проблеме.

Ранее в ряде публикаций таких авторов, как А. Буаджани, О. Буркарт, Х. Эспарза, Р. Майр, исследовались вопросы применимости метода проверки модели для некоторых формализмов, модели в рамках которых могут быть рассмотрены как вполне структурированные системы переходов, а именно: сетей Петри, ВРР, систем сложения векторов с потерями (LVAS). В данной же работе, исходя из позиции обобщения, акцентируется внимание на проблеме разрешимости темпоральных свойств для класса вполне структурированных систем переходов в целом.

Представляет интерес и выявление новых классов структурированных систем переходов. В частности, в данной работе предлагается и исследуется класс автоматных структурированных систем переходов, особенностью которого является одновременное выполнение свойств совместимости по возрастанию и совместимости по убыванию правильного квазипорядка с отношением переходов. В литературе практически не уделяется

внимание системам переходов этого класса (за исключением систем переходов с конечным числом состояний). Тем не менее, как будет показано в данной работе, формализмы, порождающие автоматные структурированные системы переходов могут быть достаточно выразительными и нетривиальными. Для демонстрации этого факта в диссертации предлагается специальный фрагмент алгебры процессов, построенный на основе таких хорошо известных алгебр процессов, как CCS (Calculus of Communicating Systems — исчисление взаимодействующих систем) Милнера и SCP (Communicating Sequential Processes — взаимодействующие последовательные процессы) Хоара, позволяющий строить формальные модели (параллельных и распределенных систем), которые могут быть рассмотрены как автоматные структурированные системы переходов.

В качестве примера конкретной реализации этого фрагмента в данной работе предлагается и исследуется формализм, названный «*взаимодействующие раскрашивающие процессы*» (ССР — Communicating Colouring Processes), позволяющий строить модели распределенных систем, где поведение каждого компонента описывается последовательным процессом и между ними организовано взаимодействие, направленное на обмен и передачу пакетов информации. Формализм ССР принимает во внимание факт передачи данных, а также позволяет отслеживать перемещение данных различного типа между компонентами системы. Но переход из одного состояния в другое не зависит от оперируемых данных, а определяется только управляющими состояниями.

**Цель работы** состоит в исследовании алгоритмической разрешимости семантических и темпоральных свойств формальных моделей параллельных и распределенных систем.

Достижение цели связывается с решением следующих задач:

1. Упорядочение теории вполне структурированных систем помеченных переходов и дополнение ее общими результатами о разрешимости темпоральных свойств для различных классов структурированных систем переходов. Выявление и изучение новых классов систем переходов.

2. Упорядочение, дополнение и улучшение результатов теории счетчиковых машин с целью ее последующего применения для дальнейших исследований формальных моделей параллельных и распределенных систем.
3. Разработка и строгое описание семантически нетривиального семейства формализмов, в рамках которых любая конкретная модель будет являться структурированной системой переходов, обладающей одновременно свойствами совместимости по возрастанию и убыванию правильного квази порядка с отношением переходов.
4. Построение достаточно выразительного формализма для моделирования и анализа распределенных систем, являющегося экземпляром выявленного семейства формализмов.
5. Определение для разработанного формализма математической семантики и исследование разрешимости проблем достижимости, ограниченности, включения и эквивалентности, а также проведение сравнения по выразительной способности с существующими классическими формальными моделями параллельных и распределенных систем.
6. Подготовка теоретической базы для построения программного комплекса моделирования, спецификации и верификации программ, написанных с применением автоматного подхода к программированию реактивных систем и систем логического управления.

**Методы исследования** базируются на аппарате математической логики, теории квази порядков, теории алгоритмов и теории автоматов, языков и вычислений. В частности, при исследовании свойств систем переходов использовались теории программных и временных логик, семантики параллелизма, вполне структурированных систем помеченных переходов. При построении и анализе новых формализмов применялись теории счетчиковых машин, сетей Петри и алгебр процессов. Выделяются три основные

направления исследований: структурированные системы переходов, счетчиковые машины, моделирование и анализ программных систем. В поддержку выделенных направлений соискателем были опубликованы соответственно монография «Вполне структурированные системы помеченных переходов» (в соавторстве с д. ф.-м. н., проф. В. А. Соколовым), учебные пособия «Счетчиковые машины» и «Верификация моделей программ».

**Научная новизна.** В работе получены новые результаты, связанные с алгоритмической разрешимостью ряда темпоральных и семантических свойств классов вполне структурированных систем помеченных переходов и счетчиковых машин. Также описан и исследован новый формализм для моделирования и анализа поведения параллельных и распределенных систем — взаимодействующие раскрашивающие процессы. Предложена технология моделирования и анализа автоматных программ.

**Теоретическая и практическая значимость.** Полученные результаты имеют в основном теоретический характер. Тем не менее, они могут быть использованы при решении практических задач верификации параллельных и распределенных систем (как с конечным, так и бесконечным числом состояний). Предложенный формализм взаимодействующих раскрашивающих процессов может быть использован как теоретическая основа для моделирования и анализа поведенческих свойств распределенных систем, в которых особое внимание уделяется отслеживанию перемещения данных различного типа среди их компонентов. Предложенная (на базе метода *model checking*) технология моделирования, спецификации и верификации программ, построенных на основе автоматного подхода к программированию, позволила создать программный комплекс «Система моделирования и анализа автоматных программ».

#### **Основные результаты и положения.**

1. Определен новый класс структурированных систем помеченных переходов — автоматные структурированные системы переходов (АССП). Доказано, что этот класс систем переходов совпадает с классом структурированных систем

переходов, для которых выполняются свойства совместности по возрастанию и убыванию отношения переходов с правильным квазипорядком, введенных на множестве состояний системы.

Для класса АССП доказана разрешимость темпоральных свойств, представимых формулами  $\mu$ -подмножества модального  $\mu$ -исчисления, построенных по следующей грамматике при интерпретации элементарных высказываний  $p$  замкнутыми кверху множествами:

$$\varphi ::= true \mid false \mid p \mid Z \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [a] \varphi \mid \mu Z. \varphi.$$

Результат распространяется на класс структурированных систем переходов с совместностью по возрастанию (ССПВ) при условии исключения из грамматики оператора всеобщности « $[ ]$ ».

2. Определен и исследован в качестве конкретной реализации взаимодействующих процессов, независимых от данных, новый формализм для моделирования параллельных и распределенных систем, позволяющий отслеживать перемещение данных различного типа между компонентами системы, — *взаимодействующие раскрашивающие автоматы* (ВРА). Доказывается, что модели в рамках этого формализма порождают автоматные структурированные системы переходов, для которых выполняются условия разрешимости квазипорядка, эффективности предбазиса и эффективности по пересечению. Следствием этого является разрешимость для ВРА проблем покрытия, субпокрытия, достижимости управляющего состояния, неизбежности и останова, а также всех темпоральных свойств, выраженных формулами  $\mu$ -подмножества модального  $\mu$ -исчисления.
3. Определен и исследован класс автоматных счетчиковых машин, использующихся как общее средство для демонстрации неразрешимости ряда проблем для систем, способных моделировать эти машины, в частности, для ВРА. Для автоматных счетчиковых машин доказана неразрешимость

проблем включения и эквивалентности, проблемы ограниченности (в случае трех счетчиков) и проблемы достижимости (в случае четырех счетчиков; однако, проблема достижимости конфигурации с нулевыми значениями всех счетчиков разрешима для произвольной автоматной счетчиковой машины). Приведен алгоритм построения полулинейного множества достижимых конфигураций для автоматных односчетчиковых машин.

4. Определен и исследован класс языков, допускаемых автоматными счетчиковыми машинами. Доказано, что этот класс замкнут относительно операций объединения, пересечения, конкатенации, бесконечной итерации, гомоморфизма и обратного гомоморфизма, т.е. является полным абстрактным семейством языков. Также установлено, что класс языков автоматных счетчиковых машин (ЯАСМ) замкнут относительно полной подстановки, но незамкнут относительно дополнения и операции обращения. Доказано, что для класса ЯАСМ разрешимы проблемы пустоты и распознавания слова языка, заданного автоматной счетчиковой машиной, но неразрешимы проблемы включения и равенства языков. Проведено сравнение с другими классами языков — регулярными, контекстно-свободными, контекстно-зависимыми языками и языками сетей Петри. Установлено, что класс АСМ-языков включает в себя регулярные языки, несравним по включению с классом контекстно-свободных языков и классом языков сетей Петри, но полностью входит в класс контекстно-зависимых языков.
5. Предложен способ моделирования трехсчетчиковых машин Минского с помощью двухсчетчиковых при взаимно однозначном соответствии конфигураций вида  $(2^N 3^K 5^L, 0)$  и  $(N, K, L)$ . Доказано, что для двухсчетчиковых машин Минского проблемы ограниченности, ограниченности одного счетчика, останова хотя бы при одном входе, достижимости заданной конфигурации не являются разрешимыми,

но имеют частичный алгоритм решения, а проблемы пустоты, тотальности, существования закливающего входа, эквивалентности, неограниченного исполнения, тотальной ограниченности, тотальной неограниченности и неограниченности хотя бы при одном входе не являются даже частично разрешимыми.

Была улучшена оценка (до квадратичной) максимальной длины конечного представления бесконечного исполнения односчетчиковой машины Минского и предложен алгоритм нахождения границ этого представления с квадратичной от числа состояний временной оценкой.

Доказана полиномиальная разрешимость проблемы тотальной ограниченности и тотальной остановки односчетчиковых машин Минского.

6. Исследована неразрешимость свойств формальных моделей систем с потерями (являющихся ССПВ) посредством счетчиковых машин с потерями, значения счетчиков которых могут внезапно уменьшаться. Для трехсчетчиковых машин с произвольным отношением потери доказана неразрешимость проблемы ограниченности. Для двухсчетчиковых машин с обнулениями установлены критерии неограниченности, с помощью которых доказываемость разрешимости проблемы ограниченности для машин этого класса. Доказана неразрешимость проблем включения и эквивалентности для слабых счетчиковых машин, а также неразрешимость проблемы достижимости для слабых четырехсчетчиковых машин с обнулениями.
7. На базе метода проверки модели (model checking) предложена технология моделирования, спецификации и верификации «автоматных» программ, рассматриваемых как область применения представленной теории АССП.

**Апробация работы.** Результаты диссертации докладывались на Международном симпозиуме «Temporal Representation and Reasoning» (Tatihou, Франция, 2004), Международном семинаре «Распределенные информационно-вычислительные ресурсы

и математическое моделирование» (Новосибирск, 2004), Международном семинаре «Program Understanding» (Новосибирск – Алтай, 2003, 2009), Всероссийской научной конференции, посвященной двухсотлетию ЯрГУ им. П. Г. Демидова (2003), Всероссийской научной конференции «Методы и средства обработки информации» (Москва, МГУ, 2005, 2009), Научно-методической конференции преподавателей математического факультета и факультета ИВТ ЯрГУ им. П. Г. Демидова (2007), семинаре «Go4IT — шаг к новым технологиям Интернета» (Москва, Институт системного программирования РАН, 2007), Областной научно-методической конференции (Ярославль, ЯрГУ, 2007), Международной научной конференции «Образование, наука и экономика в вузах. Интеграция в международное образовательное пространство» (Польша, Плоцк, 2008), Международной научно-практической конференции «Информационно-коммуникативная культура: наука и образование» (Ростов-на-Дону, 2009), Международной конференции «Компьютерные науки и информационные технологии» (Саратов, 2009), Международном рабочем совещании «Program Semantics, Specification and Verification: Theory and Applications (PSSV 2010)» (Казань, 2010), семинаре Института программных систем им. А.К. Айламазяна РАН (2010), семинаре кафедры распределенных вычислений и компьютерных сетей СПбГПУ (2010), семинаре «Теоретические основы информатики» кафедры информатики ТвГУ (2010), семинаре «Моделирование и анализ информационных систем» кафедры теоретической информатики ЯрГУ им. П. Г. Демидова (2001-2010).

**Гранты.** Работа по теме диссертации проводилась в соответствии с планами исследований по проектам, поддержанными следующими грантами: РФФИ №99-01-00-309 «Методы моделирования, анализа и верификации распределенных систем», №03-01-00-804 «Разработка новых методов и средств моделирования и анализа процессов обработки информации в распределенных системах», №07-01-00702а «Разработка формальных моделей распределенных систем и исследование их семантических свойств», №05-07095008 (публикация монографии «Структурированные системы переходов»), Федеральная целевая программа

«Интеграция», гранты ФАНИ №2007-4-1.4-18-02-041 «Разработка технологии верификации управляющих программ со сложным поведением, построенных на основе автоматного подхода» ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2012 годы», №2009-1.1-113-050-013 «Разработка фундаментальных принципов и инновационных прикладных методов для моделирования, анализа и верификации информационных систем» ФЦП «Научные и научно-педагогические кадры инновационной России на 2009-2013 годы».

**Авторские свидетельства.** Свидетельство об официальной регистрации программы для ЭВМ «Система моделирования и анализа автоматных программ», Федеральная служба по интеллектуальной собственности, патентам и товарным знакам, №2007611856. Зарегистрировано в Реестре программ для ЭВМ 7 мая 2007 г. Правообладатель: Ярославский госуниверситет им. П.Г. Демидова. Авторы: Е. В. Кузьмин, Р. А. Виноградов, В. А. Соколов.

**Личный вклад.** Выносимые на защиту результаты получены соискателем лично. В опубликованных совместных работах постановка и исследование задач осуществлялись совместными усилиями соавторов при непосредственном участии соискателя.

**Публикации.** Материалы диссертации опубликованы в 41 печатной работе, из них 11 статей в журналах из перечня ВАК, 9 статей в рецензируемых журналах, не вошедших в этот перечень, 13 статей в сборниках трудов конференций и 3 тезисов докладов, 4 учебных пособия (1 с грифом УМО) и 1 монография (изд-во «Физматлит»). Получено 1 свидетельство о регистрации программы для ЭВМ. Список публикаций приведен в конце автореферата.

**Структура и объём работы.** Диссертационная работа изложена на 312 страницах и состоит из введения, восьми глав, заключения и списка литературы. Иллюстративный материал включает 60 рисунков. Список литературы состоит из 177 наименований.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

**Во введении** обосновывается актуальность работы, приводится краткий обзор основных результатов и проблем в исследуемой области, формулируется цель диссертации, характеризуется ее научная новизна, дается краткий обзор работы.

**Первая глава** носит вводный характер. В ней приводятся понятия, определения, обозначения и результаты по теории структурированных систем помеченных переходов, используемые в диссертации.

**В разделе 1.1** определяются операции на мультимножествах, понятие правильного квазипорядка, замкнутые сверху и замкнутые снизу множества. Описываются их свойства.

**В разделе 1.2** приводятся понятия и определения теории вполне структурированных систем помеченных переходов.

Система помеченных переходов — одна из наиболее распространенных моделей для описания поведения параллельных и распределенных систем.

*Система помеченных переходов* представляет собой набор  $LTS = (S, T, \rightarrow, s_0)$ , где  $S$  — множество состояний с элементами  $s_0, s_1, s_2, \dots$ ,  $T$  — некоторый конечный алфавит пометок (множество имен действий),  $\rightarrow \subseteq (S \times T \times S)$  — отношение переходов между состояниями системы,  $s_0 \in S$  — начальное состояние системы.

*Последовательное исполнение* для LTS есть конечная или бесконечная цепочка переходов  $s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} s_2 \rightarrow \dots$ , где  $s_0$  — начальное состояние системы.

*Вполне структурированной системой переходов* со свойством совместимости *по возрастанию* (соот. *по убыванию*) называется система переходов  $LTS = (S, T, \rightarrow, s_0)$ , дополненная отношением правильного квазипорядка  $\leq \subseteq S \times S$ , совместимым *по возрастанию* (соот. *по убыванию*) с отношением переходов  $\rightarrow$ , т. е. для любых двух состояний  $s_1 \leq s'_1$  и любого перехода  $s_1 \xrightarrow{t} s_2$  существует переход  $s'_1 \xrightarrow{t} s'_2$  такой, что  $s_2 \leq s'_2$  (соот. для



любых состояний  $s_1 \leq s'_1$  и любого перехода  $s'_1 \xrightarrow{t} s'_2$  существует переход  $s_1 \xrightarrow{t} s_2$  такой, что  $s_2 \leq s'_2$ .

**Во второй главе** отмечается, что существующие формализмы, которые позволяют строить модели, являющиеся структурированными системами переходов, обладают либо свойством совместимости по возрастанию, либо свойством совместимости по убыванию. Однако, исследователями совсем не уделяется внимание формализмам, модели в рамках которых удовлетворяли бы двум условиям совместимости одновременно. Исходя из этого напрашивается вывод, что, возможно, допущение одновременного выполнения двух свойств совместимости по возрастанию и по убыванию отношения переходов с правильным квазипорядком на множестве состояний системы, приводит к вырожденной (весьма слабой по выразительной мощности) системе. В следующих главах показывается, что это не так и такого рода формализмы могут быть весьма интересны, имея достаточную мощность для построения осмысленных моделей прикладных систем.

В этой главе **в разделе 2.1** определяется новый класс структурированных систем переходов — автоматные структурированные системы переходов.

*Автоматная структурированная система переходов* — это система переходов  $LTS = (S, T, \rightarrow, s_0)$ , дополненная отношением правильного квазипорядка  $\leq \subseteq S \times S$ , где  $S = Q \times D$ ,  $Q$  — конечное множество управляющих состояний,  $D$  — конечное или бесконечное множество (на котором задан правильный квазипорядок) значений некоторых характеристик,  $s_0 \in S$  — начальное состояние, отношение переходов  $\rightarrow \subseteq S \times T \times S$  такое, что  $\exists \rightarrow \bullet \subseteq Q \times T \times Q$ :  $(q, t, q') \in \rightarrow \bullet$ , если  $\exists (q, d) \xrightarrow{t} (q', d')$ , а из существования перехода  $q \xrightarrow{t} \bullet q'$  следует, что  $\forall d_1, d'_1 \in D, d_1 \leq d'_1$ , существуют переходы  $(q, d_1) \xrightarrow{t} (q', d_2)$  и  $(q, d'_1) \xrightarrow{t} (q', d'_2)$  при  $d_2 \leq d'_2$ ; отношение  $\leq$  определяется так, что  $(q, d) \leq (q', d') \iff q = q' \text{ и } d \leq d'$ .

**В разделе 2.2** доказывается (**Теорема 2.1**), что этот класс систем переходов совпадает с классом структурированных систем переходов с совместимостью по возрастанию и убыванию.

Это означает, что если для структурированной системы переходов одновременно выполняются свойство совместимости по возрастанию и свойство совместимости по убыванию, то бесконечное множество состояний такой системы разбивается на конечное число классов эквивалентности, каждый из которых является замкнутым кверху множеством, а переход из одного состояния в другое определяется исключительно принадлежностью этих состояний соответствующим классам эквивалентности, поскольку это диктуется отношением переходов, имеющимся на конечном множестве классов эквивалентности. Другими словами, в основе отношения переходов по состояниям системы лежит конечный автомат, состояниями которого являются классы эквивалентности. Наличие такого управляющего автомата в основе отношения переходов на множестве состояний системы дало название введенному в этой главе новому классу автоматных структурированных систем переходов. С практической точки зрения автоматную структурированную систему переходов можно интерпретировать как конечный автомат, совершающий манипуляции с данными, переходы которого никак не зависят от этих данных.

**В разделе 2.3** для класса автоматных структурированных систем переходов исследуется возможность применения метода проверки модели при анализе поведенческих свойств.

Проверка модели (model checking) — один из подходов к решению проблемы верификации. В качестве языков спецификации для выражения свойств систем при этом подходе используются темпоральные логики. Задача проверки модели состоит в определении выполнимости для системы, заданной формальным образом (в виде формальной модели), свойства, записанного формулой темпоральной логики.

Метод проверки модели, широко используемый при автоматической верификации систем с конечным числом состояний, может быть применен для некоторых классов систем с бесконечным числом состояний и подмножеств темпоральных логик.

В ряде публикаций таких авторов, как А. Буаджани, О. Буркарт, Х. Эспарза, Р. Майр, исследовались вопросы разрешимости

задачи проверки модели для формализмов, порождающих различные подклассы вполне структурированных систем переходов, а именно: сетей Петри, базовых параллельных процессов (Basic Parallel Processes), систем сложения векторов с потерями (Lossy Vector Addition Systems) и различных темпоральных логик линейного времени и ветвящегося времени.

В данном же разделе исследуется разрешимость темпоральных свойств для класса автоматных структурированных систем и класса вполне структурированных систем помеченных переходов с совместимостью по возрастанию. В качестве темпоральной логики, на языке которой будут выражаться свойства системы переходов, выбрано модальное  $\mu$ -исчисление, поскольку является одной из самых мощных темпоральных логик ( $\mu$ -исчисление, например, выразительнее широко известных и часто применяемых темпоральных логик LTL и CTL).

Доказывается (**Теорема 2.2**), что все темпоральные свойства, построенные по следующей грамматике, описывающей  $\mu$ -подмножество модального  $\mu$ -исчисления, разрешимы для автоматных структурированных систем переходов:

$$\varphi ::= \text{true} \mid \text{false} \mid p \in P \mid Z \in V \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [a] \varphi \mid \mu Z. \varphi,$$

где элементарные высказывания интерпретируются замкнутыми кверху множествами, т.е.  $s \models p \implies \forall s' \geq s : s' \models p$ . С помощью формул, построенных по этой грамматике, можно задавать высказывания над конечными префиксами бесконечных последовательностей переходов (исполнений).

Этот результат распространяется на класс структурированных систем переходов с совместимостью по возрастанию при условии исключения из грамматики оператора всеобщности «[ ]» (**Теорема 2.3**).

**Третья глава** посвящена процессам, независимым от данных.

Изучение природы вычислительных процессов, в том числе параллельных и распределенных, привело к появлению и широкому применению для этих целей алгебраических подходов. В алгебрах процессов, каковыми являются, например, CCS (Calculus

of Communicating Systems — исчисление взаимодействующих систем) и CSP (Communicating Sequential Processes — взаимодействующие последовательные процессы), конструктивный подход играет важную роль: процесс в них описывается с указанием того, как он построен из меньших процессов.

В этой главе (**раздел 3.1**) представлен фрагмент алгебры процессов, определяемой в стиле CSP Хоара и CCS Милнера, позволяющий строить формальные модели (параллельных и распределенных систем), которые могут быть рассмотрены как автоматные структурированные системы переходов.

В **разделе 3.2** предлагается конкретная реализация этого фрагмента алгебры процессов, формализм для моделирования распределенных систем, позволяющий отслеживать перемещение данных различного типа между компонентами системы, — *взаимодействующие раскрашивающие процессы* (Communicating Colouring Processes — CCP):

$$E ::= \mathbf{0} \mid X \in Pr \mid v?x.E \mid v!e.E \mid x!e.E \mid e?x.E \mid E + E \mid E \parallel E.$$

$$\alpha.E \xrightarrow{\alpha} E, \quad \frac{E \xrightarrow{\alpha} E'}{X \xrightarrow{\alpha} E'} X \stackrel{\text{def}}{=} E, \quad \frac{E_j \xrightarrow{\alpha} E'}{\sum\{E_i\} \xrightarrow{\alpha} E'}$$

$$\frac{E \xrightarrow{\alpha} E'}{E \parallel F \xrightarrow{\alpha} E' \parallel F}, \quad \frac{E \xrightarrow{v!e} E' \quad F \xrightarrow{v?x} F'}{E \parallel F \xrightarrow{(v!e, v?x)} E' \parallel F'}$$

Каналы  $v$  используются для синхронизации процессов при передачи данных. Оба действия  $x!e$  и  $e?x$  обозначают присваивание  $x := e$ , где  $e$  — выражение над переменными (мультимножествами). В выражениях допустимы следующие операции:  $e_+(x_1, x_2) = x_1 + x_2$ ;  $e_{\cup}(x_1, x_2) = x_1 \cup x_2$ ;  $e_{\ominus}(x_1) = x_1 \ominus m$ ;  $e_{\cap}(x_1) = x_1 \cap m$ ;  $e_{\setminus}(x_1) = x_1 \setminus C$ ;  $e_{/}(x_1) = x_1 / C$ , где  $m \in \mathcal{M}(\Sigma)$ ,  $C \subseteq \Sigma$ ,  $\Sigma$  — конечный набор цветов,  $\mathcal{M}(\Sigma)$  — множество всех мультимножеств над  $\Sigma$ .

В рамках CCP возможно построение моделей параллельных и распределенных систем, где поведение каждого компонента описывается последовательным процессом и между ними организовано взаимодействие, направленное на обмен и передачу пакетов

информации. В данной модели игнорируются значения передаваемых данных и принимается в расчет только лишь тип данных, определяемый, возможно, типом источника данных и уровнем секретности. Фокусируется внимание на возможности отслеживать перемещение данных различного типа между компонентами системы. В этом аспекте безопасности информационных потоков могут рассматриваться вопросы о доставке пакетов данных до адресата и о недопущении передачи конфиденциальных данных в открытую небезопасную среду.

Переход из одного допустимого состояния компонента распределенной системы в другое осуществляется посредством исполнения действия одного из следующих типов: отправка или прием сообщения в виде пакета данных из одного компонента в другой или же во внешнюю среду, формирование пакетов данных из других информационных пакетов (включая добавление и извлечение данных определенного типа).

Модель распределенной системы, реализованная в рамках формализма ССР, представляет собой параллельную композицию взаимодействующих последовательных процессов, синхронизирующихся между собой посредством взаимодействия через общие переменные (каналы). Каждое действие процесса — это либо передача, либо прием значения некоторого выражения над переменными. Результатом вычисления такого выражения является мультимножество над типами (цветами) данных. Совершение действия ССР приводит к изменению значения (раскраски) переменной. Поэтому работа процессов интерпретируется как раскрашивание множеств переменных. В терминах раскрашивания могут быть адекватно выражены, например, свойства систем, связанные с безопасностью информационных потоков.

Формализм ССР принимает во внимание факт передачи данных, а также позволяет отслеживать перемещение данных различного типа между компонентами системы. Но переход из одного состояния в другое не зависит от данных, над которыми совершаются операции, а определяется только управляющими состояниями. Таким образом, формализм ССР занимает особое положение среди существующих формализмов.

**В разделе 3.3** доказываемся, что модели в рамках формализма взаимодействующих раскрашивающих процессов являются автоматными структурированными системами переходов, для которых выполняются условия разрешимости квазипорядка, эффективности предбазиса и эффективности по пересечению (**Утверждения 3.1 и 3.2, Теорема 3.1**). Следствием этого является разрешимость для взаимодействующих раскрашивающих процессов проблем покрытия, субпокрытия, достижимости управляющего состояния, неизбежности и останова, а также всех темпоральных свойств, выраженных формулами  $\mu$ -подмножества модального  $\mu$ -исчисления (при интерпретации элементарных высказываний как замкнутых кверху множеств).

Также в главе обращается внимание на основные объекты исследования данной диссертационной работы, расположенные по включению подклассов порождаемых автоматных структурированных систем переходов:  $АСМ \subseteq ВРП \subseteq АССП$ . Для класса автоматных структурированных систем переходов (АССП) в целом исследуются семантические и темпоральные свойства, разрешимость которых следует из наличия свойства совместимости по возрастанию и убыванию правильного квазипорядка и отношения переходов, заданных на множестве состояний системы. Введение взаимодействующих раскрашивающих процессов (ВРП) демонстрирует тот факт, что формализмы, порождающие автоматные структурированные системы переходов могут быть достаточно выразительными, чтобы иметь практическое применение. Нетривиальность формализмов класса АССП подтверждается результатами о неразрешимых свойствах для автоматных счетчиковых машин, которые рассматриваются в шестой главе. *Автоматные счетчиковые машины* (АСМ) используются в качестве общего средства для демонстрации неразрешимости ряда проблем для систем, которые могут моделировать эти машины, в частности для *взаимодействующих раскрашивающих процессов*. Неразрешимость доказываемся методом сведения неразрешимых проблем машин Минского (см. следующую главу) к соответствующим проблемам автоматных счетчиковых машин.

Таким образом, формализмы из класса АССП занимают особое место в иерархии формализмов, выстраиваемой по выразительной мощности от конечного автомата до машины Тьюринга.

**Четвертая глава** посвящена теории счетчиковых машин.

**В разделе 4.1** приводится определение машин Минского. Счетчиковая машина Минского представляет собой конечный автомат, имеющий доступ к памяти в виде нескольких неограниченных счетчиков, значениями которых могут быть лишь целые неотрицательные числа. Над каждым счетчиком допустимы лишь две операции: безусловное увеличение на единицу и условное вычитание единицы (т. е. в случае, когда счетчик равен нулю, происходит переход в альтернативное состояние).

Счетчиковые машины были введены в работах М. Минского, получив название «программные машины». Известно, машины Минского с двумя счетчиками равноможны машинам Тьюринга. Счетчиковые машины используются для доказательства тьюринговской мощности различных языков программирования и формальных моделей программных систем.

*Счетчиковая машина Минского  $M$*  представляет собой набор  $(q_0, q_n, Q, X, \Delta)$ , где  $Q = \{q_0, \dots, q_n\}$  — множество состояний машины;  $q_0 \in Q$  — начальное состояние;  $q_n \in Q$  — финальное состояние;  $X = \{x_1, \dots, x_m\}$  — множество счетчиков, которые могут принимать значения из  $\mathbb{N} \cup \{0\}$ ;  $\Delta = \{\delta_0, \dots, \delta_{n-1}\}$  — набор правил перехода по состояниям машины;  $\delta_i$  — правило перехода для состояния  $q_i$ . Состояния  $q_i$ ,  $0 \leq i \leq n-1$ , подразделяются на два типа. Состояния первого типа имеют правила переходов вида:  $\delta_i: x_j := x_j + 1; \text{ goto } q_k$ , где  $1 \leq j \leq m$ ,  $0 \leq k \leq n$ . Для второго типа:  $\delta_i: \text{ if } x_j > 0 \text{ then } (x_j := x_j - 1; \text{ goto } q_k) \text{ else goto } q_l$ , где  $1 \leq j \leq m$ ,  $0 \leq k, l \leq n$ . При попадании в состояние  $q_n$  машина Минского  $M$  завершает свою работу.

*Конфигурация* счетчиковой машины — набор  $(q_i, c_1, \dots, c_m)$ , где  $q_i$  — состояние машины,  $c_1, \dots, c_m \in \mathbb{N} \cup \{0\}$  — значения соответствующих счетчиков.

В этом разделе предлагается способ моделирования трехсчетчиковой машины Минского  $3сМ$  с помощью двухсчетчиковой  $2сМ$ , при котором конфигурация со значениями счетчиков

$y_1 = 2^a \cdot 3^b \cdot 5^c$  и  $y_2 = 0$  машины  $2сМ$  будет существовать тогда и только тогда, когда существует конфигурация  $x_1 = a$ ,  $x_2 = b$  и  $x_3 = c$  машины  $3сМ$  (**Теорема 4.3**).

Этот результат используется далее при исследовании проблем достижимости заданной конфигурации и ограниченности для счетчиковых машин разных классов.

**В разделе 4.2** на основе «базовых» неразрешимых проблем остановки и зацикливания, которые машины Минского наследуют от машин Тьюринга, с помощью метода сведения доказываются неразрешимость ряда проблем для машин с двумя счетчиками.

Одной из таких задач, например, является проблема ограниченности, которая состоит в нахождении алгоритма, определяющего для произвольной машины Минского, ограничены ли (или неограниченно возрастают) значения счетчиков на протяжении ее бесконечного исполнения.

Доказывается, что для двухсчетчиковых машин Минского проблемы ограниченности, ограниченности одного счетчика, остановки хотя бы при одном входе, достижимости заданной конфигурации не являются разрешимыми, но имеют частичный алгоритм решения, а проблемы пустоты, тотальности, существования зацикливающего входа, эквивалентности, неограниченного исполнения, тотальной ограниченности, тотальной неограниченности и неограниченности хотя бы при одном входе не являются даже частично разрешимыми (**Теоремы 4.4–4.21**).

**В разделе 4.3** приводятся вспомогательные преобразования Р. Шреппеля, которые позволяют представлять двухсчетчиковые машины Минского в виде однорегистровых машин.

**В разделе 4.4** рассматриваются односчетчиковые машины Минского. В отличие от машин с большим количеством счетчиков для односчетчиковых машин Минского разрешимы такие классические проблемы, как достижимость заданной конфигурации, ограниченность, пустота, включение и равенство множеств достижимых конфигураций двух машин. Разрешимость перечисленных проблем является следствием того, что полное поведение (конечное или бесконечное) односчетчиковой машины может быть представлено за полиномиальное время от размера

машины в удобном (для дальнейших исследований) конечном виде. Ранее была известна оценка  $|Q|^3$  (где  $|Q|$  — количество состояний) длины конечного представления бесконечного исполнения односчетчиковой машины. В этом разделе предлагается алгоритм (максимально приближенный по структуре к способу описания функционирования самой счетчиковой машины) нахождения границ этого конечного представления с временной оценкой  $O(|Q|^2)$  (**Теорема 4.24**). Если односчетчиковая машина имеет бесконечное исполнение  $(q_0, c_0)(q_1, c_1)(q_2, c_2) \dots$ , границами его конечного представления являются константы  $K_1$ ,  $K_2$  и  $K_3$  такие, что  $K_1 + K_2 \leq |Q|^2$ ,  $K_3 \leq |Q|$  и для любого  $i \geq K_1$  выполняется  $(q_{i+K_2}, c_{i+K_2}) = (q_i, c_i + K_3)$ .

Также в разделе доказываемся (**Теорема 4.25**) полиномиальная разрешимость проблем тотальной ограниченности и тотальной остановки односчетчиковых машин Минского.

В пятой главе рассматриваются «слабые» счетчиковые машины в качестве общего средства для демонстрации алгоритмической неразрешимости ряда проблем для структурированных систем переходов с совместимостью по возрастанию, способных моделировать поведение этих машин.

В разделе 5.1 определяются счетчиковые машины Минского с потерями, т. е. машины со счетчиками, значения которых могут самопроизвольно уменьшаться в любое время. Эти машины слабее по выразительной мощности, чем машины Минского, но в то же время счетчиковые машины с потерями имеют интересные свойства, которые по-прежнему остаются неразрешимыми.

Счетчиковые машины с потерями были введены Р. Майром как средство для доказательства неразрешимости проблем для систем с потерями, например, проблемы верификации систем с ненадежными каналами (lossy FIFO-channel systems), задачи проверки модели для сетей Петри с потерями, некоторых проблем для сетей Петри с обнуляющими дугами и т. п.

В пункте 5.1.1 определяются различные отношения потери для счетчиковых машин Минского. Важным примером счетчиковой машины с потерями является *счетчиковая машина с обнулениями*, для которой предполагается, что в состоянии второго

типа машины Минского при проверке на ноль счетчик может внезапно обнуляться. Счетчиковая машина с обнулениями для состояний  $q_i$  второго типа имеет следующее правило переходов:

$\delta_i$ : if  $x_j > 0$  then  $(x_j := x_j - 1; \text{goto } q_k)$   $\square$   $(x_j := 0; \text{goto } q_l)$ , где  $\square$  — оператор недетерминированного выбора.

Пункт 5.1.2 посвящен проблеме ограниченности для счетчиковых машин с потерями. Ранее в работе авторов К. Дюфура, П. Жанкара и Ф. Шнеблена при исследовании свойств сетей Петри с обнуляющими дугами была показана неразрешимость проблемы ограниченности для трехсчетчиковых машин с обнулениями. В этом пункте производится обобщение этого результата на трехсчетчиковые машины с произвольным отношением потери (**Теорема 5.2**).

В пункте 5.1.3 вводится понятие однорегистровых машин с обнулениями, способных моделировать двухсчетчиковые машины с отношением обнуления. Устанавливаются критерии неограниченности. Доказывается, что проблема ограниченности для однорегистровых машин с обнулениями (а значит, и для двухсчетчиковых машин с обнулениями) разрешима (**Теорема 5.4**).

В разделе 5.2 определяется новый класс *слабых счетчиковых машин*, у которых при проверке на нуль значения счетчика может происходить неверное распознавание нуля (т. е. положительное значение счетчика может быть принято машиной за нулевое). В этом случае правило перехода для состояний второго типа счетчиковой машины Минского преобразуется в следующее:

$\delta_i$ : if  $x_j > 0$  then  $(x_j := x_j - 1; \text{goto } q_k)$   $\square$   $(\text{goto } q_l)$ .

Для класса этих машин доказываемся неразрешимость проблем включения и равенства множеств достижимых векторов значений счетчиков двух машин (**Теоремы 5.6 и 5.7**). Кроме того, доказываемся неразрешимость проблемы достижимости заданного вектора значений счетчиков для слабых четырехсчетчиковых машин с отношением обнуления (**Теорема 5.8**).

В шестой главе с помощью автоматных счетчиковых машин исследуются неразрешимые свойства автоматных структурированных систем переходов, в частности взаимодействующих раскрашивающих процессов.

Распределенные и параллельные системы обычно моделируются системами переходов с конечным числом управляющих состояний и с различными видами переменных и структур данных, таких как счетчики, очереди и т. д. В данной главе рассматриваются системы переходов, где каждый переход определяется в соответствии с управляющими состояниями и независимо от данных, с которыми система совершает манипуляции. Вводятся *автоматные счетчиковые машины* в качестве общего средства для демонстрации неразрешимости ряда проблем для систем, которые могут моделировать эти машины, в частности для *взаимодействующих раскрашивающих процессов*, используемых для моделирования перемещения данных различного типа между компонентами распределенной системы. Показывается неразрешимость для автоматных счетчиковых машин проблем ограниченности, достижимости заданной конфигурации, включения и равенства множеств достижимых векторов значений счетчиков двух машин. Из чего можно сделать вывод, что даже при допущенных ограничениях на структуру системы переходов могут строиться довольно выразительные формализмы, такие как взаимодействующие раскрашивающие процессы.

Исследуются автоматные счетчиковые машины малой размерности, т. е. машины с одним, двумя и тремя счетчиками, и другими ограничениями.

**В разделе 6.1** вводятся автоматные счетчиковые машины. *Автоматная счетчиковая машина* представляет собой набор  $(\bar{c}_0, q_0, q_f, Q, X, \rightarrow, T, E)$ , где  $X = \{x_1, \dots, x_m\}$  — множество счетчиков,  $Q = \{q_0, \dots, q_f\}$  — множество состояний,  $q_0$  — начальное состояние,  $q_f$  — финальное состояние,  $\bar{c}_0$  — вектор начальных значений счетчиков,  $T$  — конечное множество меток действий, соответствующих выражениям (преобразованиям) над счетчиками,  $\rightarrow \subseteq Q \setminus \{q_f\} \times T \times Q$  — отношение переходов. Каждой метке перехода  $t$  сопоставлено некоторое выражение  $E(t)$  из множества  $E$ , представляющего собой конечное множество выражений над счетчиками вида  $x_i := x_i$  (значение счетчика не меняется),  $x_i := x_i + 1$ ,  $x_i := x_i \ominus 1$  или  $x_i := x_i + \min(x_j, 1)$ , где  $\ominus$  — операция вычитания до нуля,  $i, j = 1 \dots m$ . Переход

зависит лишь от управляющего состояния машины. Во время перехода происходит изменение значения одного из счетчиков в соответствии с выражением, приписанным переходу.

**В разделе 6.2** доказывается неразрешимость проблемы ограниченности (всех бесконечных последовательностей переходов) автоматных трехсчетчиковых машин (**Теорема 6.2**).

**В разделе 6.3** доказывается неразрешимость проблем включения и равенства множеств достижимых векторов значений счетчиков автоматных счетчиковых машин (**Теоремы 6.3 и 6.4**).

**В разделе 6.4** доказывается разрешимость задачи достижимости конфигурации с нулевыми значениями всех счетчиков для всего класса автоматных счетчиковых машин (**Теорема 6.5**). Также доказывается, что для автоматных четырехсчетчиковых машин проблема достижимости ненулевого вектора значений счетчиков является неразрешимой (**Теорема 6.6**).

**Раздел 6.5** посвящен автоматным счетчиковым машинам малой размерности.

В теории формальных моделей особое пограничное место занимают абстрактные машины с двумя счетчиками, поскольку уменьшение количества счетчиков до одного обычно приводит к резкому росту числа разрешимых свойств, в то время как уже для трехсчетчиковых машин того же класса имеем неразрешимыми подавляющее большинство полезных свойств. В связи с этим представляется интересным исследование выразительной способности (мощности) абстрактных двухсчетчиковых машин разных классов.

Известно, что для двумерных систем векторного сложения с состояниями (сетей Петри, среди позиций которых неограниченными являются только две) и некоторых их расширений (с помощью добавлений обнуляющих, тестирующих на ноль и перемещающих значения счетчиков переходов) классические проблемы эквивалентности, включения и достижимости являются алгоритмически разрешимыми. Например, алгоритмы, представленные в работах Дж. Хопкрофта, Ж.-Ж. Пансио, Г. Сютре и А. Финкеля, производят построение множеств достижимых конфигураций, которые в случае двух счетчиков (двух неогра-

ниченных позиций) представляют собой полулинейные множества, имеющие конечное представление. Таким образом, каждая из указанных проблем может быть легко задана формулой арифметики Пресбургера на основе полученного конечного представления множества достижимости. Известно, что арифметика Пресбургера разрешима, т. е. существует алгоритм определения истинности любой ее формулы.

Применение подобных алгоритмов ограничивается случаем абстрактных машин с двумя счетчиками, поскольку уже для «слабых» машин с тремя счетчиками, например трехмерных систем векторного сложения с состояниями (сетей Петри с тремя неограниченными позициями), множество достижимых конфигураций не является полулинейным.

В этом разделе объектом исследования является класс автоматных двухсчетчиковых машин. Основной вопрос — будет ли полулинейным множество достижимости каждой машины из этого класса. В случае полулинейности множества достижимости могли бы быть применены выше упомянутые алгоритмы. К сожалению, эта проблема пока остается открытой.

Показывается, что автоматные трехсчетчиковые машины, множества достижимости которых не являются полулинейными, могут иметь весьма простой вид (**Лемма 6.1**).

В этом разделе доказываемся, что множество достижимых конфигураций автоматных односчетчиковых машин является полулинейным (**Теорема 6.7**). Предлагается алгоритм построения множества достижимых значений счетчика в каждом локальном состоянии машины, т. е. для каждого локального состояния машины строится (полулинейное) множество значений счетчика, которое может быть достигнуто в этом локальном состоянии.

**В разделе 6.6** кратко обсуждаются свойства классов автоматных счетчиковых машин, имеющих ограничения на число перемен направлений роста/убывания значений счетчиков и структуру графа переходов.

Ранее в статье А. Финкеля изучался специальный класс расширенных  $n$ -счетчиковых машин с ограничением на количество перемен направлений роста/убывания значений счетчиков.

В ней было показано, что множество достижимых конфигураций машин этого класса представляет собой полулинейное множество. Поскольку, очевидно, этот класс машин включает в себя класс автоматных счетчиковых машин с ограничением на количество перемен направлений роста/убывания значений счетчиков, то все положительные результаты (в том числе про полулинейность множества достижимости) могут быть распространены и на последний подкласс счетчиковых машин. Однако в работах О. Ибарры и А. Финкеля была доказана неразрешимость проблемы определения, является ли произвольная  $n$ -счетчиковая машина Минского ограниченной по числу перемен направлений роста/убывания значений счетчиков. В них утверждается, что, например, для систем векторного сложения с состояниями указанная проблема разрешима. Является ли разрешимой эта задача для произвольной автоматной счетчиковой машины — открытая проблема. Из работ А. Финкеля и Д. Леру фактически следует, что если граф управления некоторой автоматной  $n$ -счетчиковой машины не содержит вложенных циклов, т. е. автоматная счетчиковая машина является «плоской», то множество достижимых конфигураций такой машины полулинейно.

**Седьмая глава** посвящена классу формальных языков, которые допускаются автоматными счетчиковыми машинами.

**В разделе 7.1** автоматная счетчиковая машина рассматривается как распознаватель слов в некотором алфавите  $\Sigma$ , которые считываются с входной ленты. Для этого каждой метке перехода  $t$  машины сопоставляется символ из  $\Sigma \cup \{\varepsilon\}$ , где  $\varepsilon$  — «пустая» строка. Машина-распознаватель работает следующим образом. Начав с крайнего слева символа на ленте, она на каждом шаге считывает символ с ленты и 1) или совершает переход  $t$  из одной конфигурации в другую, который соответствует символу на ленте, и передвигает считывающую головку вправо на одну ячейку ленты, 2) или совершает переход  $t$ , соответствующий символу  $\varepsilon$  без перемещения считывающей головки по ленте. Машина-распознаватель стартует в начальном состоянии  $q_0$  при нулевых значениях счетчиков. Если существует такая последовательность переходов машины, которая после полного прочтения

слова (с ленты) приводит машину в заключительное состояние  $q_f$  также при нулевых значениях счетчиков, то такое слово *допускается*, или *принимается*, машиной. Во всех других случаях входное слово *отвергается*. Примеры языков автоматных счетчиковых машин-распознавателей:  $L_{\geq n^2} = \{a^n b^{n^2} b^* \mid n \geq 0\}$ ,  $L_{\geq 2^n} = \{a^n b^{2^n} b^* \mid n \geq 0\}$  и  $L_{k \leq l \leq m} = \{a^k b^l c^m \mid k \leq l \leq m\}$ .

**В разделах 7.2–7.4** доказывается, что класс языков автоматных счетчиковых машин-распознавателей (ЯАСМ) замкнут относительно операций объединения, пересечения, конкатенации, бесконечной итерации, гомоморфизма и обратного гомоморфизма (**Теоремы 7.4–7.7**). Отсюда следует, что он является полным абстрактным семейством языков со всеми вытекающими из этого свойствами. Более того, класс ЯАСМ замкнут относительно полной подстановки (**Теорема 7.8**), но незамкнут относительно дополнения и операции обращения (**Теорема 7.9**). Для класса ЯАСМ разрешимы проблемы пустоты и распознавания слова языка (**Теоремы 7.1 и 7.3**), заданного автоматной счетчиковой машиной, но неразрешимы проблемы включения и равенства языков (**Теоремы 7.10 и 7.11**). Проводится сравнение с другими классами языков — регулярными, контекстно-свободными, контекстно-зависимыми языками и языками сетей Петри. Показывается, что класс АСМ-языков включает в себя регулярные языки, несравним по включению с классом контекстно-свободных языков и классом языков сетей Петри, но полностью входит в класс контекстно-зависимых языков.

**Восьмая глава** посвящена описанию, спецификации и верификации моделей программ, построенных на основе автоматного подхода к программированию.

Автоматное программирование здесь рассматривается как одна из возможных *реальных* областей практического применения представленной в предыдущих главах теории автоматных структурированных систем переходов. Предлагаемая в главе технология спецификации и верификации автоматных программ на базе метода проверки модели (model checking) ориентирована прежде всего на формальные модели, порождающие конечные системы переходов. Описанная теория автоматных структурированных си-

стем переходов задает верхнюю границу возможного расширения автоматных (конечных) моделей до моделей с бесконечным числом состояний при сохранении разрешимости ряда темпоральных и семантических свойств.

Во введении к главе (**раздел 8.1**) отмечается, что технология автоматного программирования является современной отечественной разработкой, которая активно развивается и поддерживается рядом российских исследовательских групп. При автоматном подходе к проектированию и построению программ выделяются две части: системно независимая и системно зависимая. Первая часть реализует логику программы и задается системой взаимодействующих автоматов. Проектирование каждого автомата состоит в создании схемы связей, описывающей его интерфейс, и графа переходов, определяющего его поведение. По этим двум документам строится модуль программы (системно зависимая часть), соответствующий автомату.

«Автоматное» программирование не зависит от платформы, операционной системы или языка программирования. Технология автоматного программирования является достаточно эффективной при построении программного обеспечения для «реактивных» систем и систем логического управления. Эта технология, не исключая других методов построения программного обеспечения «без ошибок», существенно более конструктивна, так как позволяет начинать «борьбу с ошибками» еще на стадии алгоритмизации.

Для анализа корректности поведения «автоматных» программ удобным является метод проверки модели. При этом методе для программы строится формальная конечная модель (т. е. с конечным числом состояний), а проверяемые свойства задаются с помощью формул темпоральной логики. Проверка выполнимости темпоральных формул, задающих свойства модели, происходит автоматическим образом. Построение модели программы для автоматической верификации является весьма серьезной и сложной задачей. При построении модели для программы, написанной традиционным способом, возникает серьезная проблема адекватности этой программной модели исходной программе. Модель



может не учитывать ряд программных свойств или порождать несуществующие свойства. Автоматный подход к программированию с точки зрения моделирования и анализа программных систем имеет ряд преимуществ по сравнению с традиционным подходом. При автоматном программировании исключена проблема адекватности модели, поскольку набор взаимодействующих автоматов, описывающий логику программы, уже является адекватной моделью, по которой формальным образом строится программный модуль, что является бесспорным плюсом автоматной технологии. Модель имеет конечное число состояний, что является необходимым на практике условием для успешной автоматической верификации.

Одними из наиболее популярных темпоральных логик для спецификации и верификации свойств программных систем являются логика CTL (branching-time logic или computation tree logic) и логика линейного времени LTL (linear-time logic). Для целей верификации автоматных программ логика LTL заслуживает особого внимания, поскольку любая формула в рамках этой логики по сути представляет собой автомат Бюхи, описывающий бесконечные допустимые пути структуры Крипке, которая в свою очередь задает поведение (все возможные исполнения) проверяемой на корректность «автоматной» модели. Это позволяет при спецификации и верификации «автоматных» программ оперировать в основном таким простым понятием, как «автомат».

Существует ряд программных средств верификации абстрактных моделей, построенных в рамках некоторых формализмов (сети Петри, взаимодействующие асинхронные процессы, автоматы с реальным временем, гибридные автоматы и т. д.), базирующихся на общих подходах и технологиях. Среди этих средств верификации классическим методом проверки модели, можно выделить, например, CPNTools, SPIN, SMV и CADP, использующие в качестве языков спецификации формальных моделей соответственно раскрашенные сети Петри высокого уровня, язык Promela, синхронные процессы и язык LOTOS. С одной стороны, можно считать, что во многом целью создания такого рода программных продуктов является отработка тех или

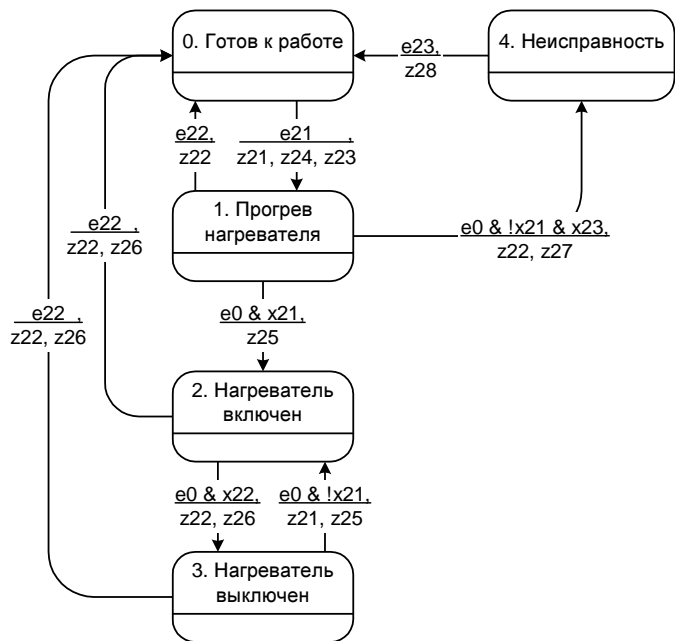
иных теоретических и прикладных методов верификации моделей, но в то же время, многие из них имеют ориентированный на конкретную область применения язык спецификации. Однако программных средств верификации, направленных непосредственно на поддержку технологии автоматного программирования, не существует, что не исключает, а скорее предполагает, активное использование уже имеющихся (проверенных временем) разработок ведущих лабораторий мира, которые обычно долгое время сопровождаются тематическими конференциями (зачастую полностью посвященными этим продуктам). Цель исследований по теме данной главы — программный комплекс, рассчитанный на проектирование программных продуктов с помощью автоматных моделей, из которых после соответствующего анализа на корректность происходит дальнейшее построение и порождение программы. Таким образом, в этом случае будет анализироваться адекватная автоматная модель, отвечающая за функциональную логику программы. По автоматной модели проверяются (относительно спецификации) функциональные требования, предъявляемые к программному обеспечению. Если после проверки автоматной модели в программе обнаружатся ошибки, то они, скорее всего, будут относиться к ошибкам деталей реализации, и никак не будут касаться функциональной логики программы, что не потребует (при исправлении этих ошибок) глобальной перестройки (перепроектирования) всей программы.

В этой главе предлагается и исследуется одна из возможных моделей построения автоматных программ — иерархическая модель. Для спецификации и верификации автоматных программ, построенных по этой модели, предлагается технология применения метода проверки модели (метода model checking). Исследуется возможность спецификации (с точки зрения простоты и адекватности восприятия) структурных и семантических свойств автоматных программ с помощью темпоральной логики. Более того, важным моментом в настоящей работе является выбор обозреваемого поведения модели автоматных программ, так как именно от него зависят как классы (виды) проверяемых свойств, так и сама возможность адекватной спецификации.

В разделах 8.2 и 8.3 вводится и демонстрируется на примере иерархическая модель автоматных программ.

Ниже (в качестве примера) представлен граф переходов и схема связей автомата управления нагревателем.

Автомат управления нагревателем



Модель представляет собой набор автоматов, между которыми установлена иерархия по вызываемости (вложенности). Выделяется основной автомат. Остальные автоматы являются вложенными. Автомат может вызывать (передавать управление) стоящему ниже по иерархии автомату. В таком случае он называется *главным*, а тот, которому передаются управление, — *вызываемым*. У каждого вложенного автомата существует только один главный автомат, из которого он вызывается.

Система автоматов рассматривается как система управления некоторым объектом. Система получает от объекта управления события *e*, характеризующие, например, изменение его состояния, а также сама запрашивает текущие параметры объекта *x*, что также считается входным воздействием на систему. В то же время система управления, реагируя на поступающую информацию, сама оказывает воздействия *z* на объект управления.

Кроме описанного взаимодействия с «внешней средой», происходит аналогичное взаимодействие автоматов между собой внутри системы, посредством передачи (со стороны главного автомата) управления с некоторым событием вложенным автоматам и отслеживанием их текущих состояний.

В рассматриваемой модели события от объекта управления принимает и реагирует на них только основной автомат, вложенные же автоматы обращаются к объекту управления только с запросами состояния его параметров (причем вложенный автомат может сделать запрос лишь в том случае, если главный автомат передал ему управление).

В разделах 8.4 и 8.5 обсуждаются вопросы спецификации и верификации автоматных программ на базе метода проверки модели. Предлагается и обосновывается (с точки зрения удобства спецификации и проверки свойств) структура Крипке для автоматных программ, построенных в рамках введенной иерархической модели.

Разделы 8.6 и 8.7 являются вспомогательными. В них дается синтаксис и семантика темпоральных логик LTL и CTL.

В разделе 8.8 рассматриваются типичные темпоральные свойства моделей автоматных программ.

Пример свойства для приведенного выше автомата управлением нагревателем  $A2$ :

«Перегрев нагревательного элемента» — не существует такой ситуации (такого бесконечного процесса), при которой нагревательный элемент будет продолжать нагреваться после преодоления максимального температурного порога. Эта ситуация возможна, когда автомат  $A2$  бесконечно долго не имеет возможности проверить свои условия переходов — не может обратиться с запросом к датчику температуры, или не получает команды выключения от главного автомата бойлера  $A1$ . Это свойство на языке темпоральной логики LTL относительно определенной структуры Крипке имеет следующий вид:

$$!<>[(y_2 = 2 \wedge (auto = 2 \longrightarrow evnt \neq e_0 \wedge evnt \neq e_{22}))],$$

где  $y_2$  отвечает за текущее состояние автомата  $A2$ , переменная  $auto$  хранит номер автомата, которому было передано управление, а в  $evnt$  содержится передаваемое в вызываемый автомат событие.

**В разделе 8.9** обсуждаются вопросы редукции числа состояний структуры Крипке автоматной модели относительно темпоральных свойств.

Поскольку при автоматической верификации методом проверки модели применяются переборные алгоритмы, очень важно породить структуру Крипке автоматной модели таким образом, чтобы она содержала как можно меньше состояний, но по-прежнему позволяла бы верифицировать темпоральные свойства автоматной модели. В данном случае возможно осуществление редукции (уменьшения) введенной структуры Крипке автоматной модели относительно проверяемых темпоральных свойств, заданных формулами логик CTL и LTL.

**Раздел 8.10** рассматриваются проблемы практического применения метода проверки модели для моделирования, спецификации и верификации автоматных программ.

**Раздел 8.11** посвящен построению, спецификации и верификации программных систем реального времени с использованием автоматного подхода к программированию.

**В разделе 8.12** рассматриваются перспективы использования метода формальных утверждений о трассах как средства спецификации «автоматных» программ.

В заключении к главе (**раздел 8.13**) отмечается, что автоматная программа является исключительно удобным объектом для верификации методом проверки модели. Свойства программной системы в виде автоматов формулируются и специфицируются естественным и понятным образом, легко соотносятся со взаимодействующими автоматами, которые задают логику «автоматной» программы, так как элементами управляющих автоматов являются либо четко выраженные состояния объекта управления, либо понятные действия над ним. Проверка свойств осуществляется в терминах, которые естественно вытекают из автоматной модели программы. Элементарные высказывания в рамках свойств определяются над элементами модели — событиями, входными и выходными воздействиями и состояниями. Если после верификации методом проверки модели тестирование выявит ошибку, то вид этой ошибки скорее всего будет относиться к некорректной программной реализации выходных воздействий, а не к нарушению логики программы, что при исправлении не потребует глобальной перестройки «автоматной» программы (все сведется к локальным исправлениям внутри одной или нескольких отдельных обычно небольших процедур, корректность которых затем может быть доказана методом дедуктивного анализа). Разработка формальных методов и технологий моделирования, спецификации и верификации автоматных программ и построения на этой базе интегрированного программного комплекса, позволит проектировать и реализовывать надежные программные системы управления ответственными объектами.

Исследования в этом направлении привели к созданию прототипа программы верификации под названием «Система моделирования и анализа автоматных программ» (регистр. номер свидетельства: 2007611856; авторы: Кузьмин Е. В., Соколов В. А., Виноградов Р. А.).

**В заключении** формулируются основные результаты и выводы диссертационной работы.

**ПУБЛИКАЦИИ АВТОРА ПО ТЕМЕ ДИССЕРТАЦИИ****Статьи в журналах из перечня ВАК**

1. Кузьмин Е. В., Чалый Д. Ю. О языках автоматных счетчиковых машин // Моделирование и анализ информационных систем, 2010. Т. 17, №2. 24 с. (лично автором – 12 с.)
  2. Кузьмин Е. В., Соколов В. А., Чалый Д. Ю. Проблемы ограниченности счетчиковых машин Минского // Программирование, 2010. Т. 36, №1. С. 3–10 (лично автором – 3 с.).
  3. Кузьмин Е. В., Чалый Д. Ю. О множестве достижимости автоматных счетчиковых машин // Моделирование и анализ информационных систем, 2010. Т. 17, №1. С. 52–64 (лично автором – 7 с.).
  4. Кузьмин Е. В., Чалый Д. Ю. О множестве достижимости автоматных трехсчетчиковых машин // Моделирование и анализ информационных систем, 2009. Т. 16, №3. С. 77–84 (лично автором – 4 с.).
  5. Кузьмин Е. В., Чалый Д. Ю. Об одном классе счетчиковых машин // Моделирование и анализ информационных систем, 2009. Т. 16, №2. С. 75–82 (лично автором – 4 с.).
  6. Кузьмин Е. В., Соколов В. А., Чалый Д. Ю. Применение метода формальных утверждений о трассах для спецификации, построения и верификации автоматных программ // Программирование, 2009. Т. 35, №1. С. 61–77 (лично автором – 6 с.).
  7. Кузьмин Е. В., Чалый Д. Ю. Алгоритмы для проблемы ограниченности счетчиковых машин // Моделирование и анализ информационных систем, 2008. Т. 15, №4. С. 42–55 (лично автором – 7 с.).
  8. Кузьмин Е. В. Проблема ограниченности для счетчиковых машин с потерями // Моделирование и анализ информационных систем, 2008. Т. 15, №3. С. 14–27 (лично автором – 14 с.).
  9. Кузьмин Е. В., Соколов В. А., Чалый Д. Ю. Проблемы ограниченности счетчиковых машин Минского // Доклады Академии наук, 2008. Т. 421, №6. С. 741–743 (лично автором – 1 с.).
  10. Кузьмин Е. В., Чалый Д. Ю. О разрешимости проблем ограниченности для счетчиковых машин Минского // Моделирование и анализ информационных систем, 2008. Т. 15, №1. С. 16–26 (лично автором – 6 с.).
  11. Кузьмин Е. В., Соколов В. А. Моделирование, спецификация и верификация «автоматных» программ // Программирование, 2008. Т. 34, №1. С. 38–60 (лично автором – 12 с.).
- Работы, опубликованные в других журналах и сборниках**
12. Кузьмин Е. В., Соколов В. А., Чалый Д. Ю. О языках автоматных счетчиковых машин // Материалы Международного семинара «Семантика, спецификация и верификация программ: теория и приложения». Казань, 2010. С. 70–75 (лично автором – 2 с.).
  13. Кузьмин Е. В. Счетчиковые машины. Ярославль: ЯрГУ, 2010. 128 с. (лично автором – 128 с.).
  14. Кузьмин Е. В., Соколов В. А., Чалый Д. Ю. Об одном формальном подходе к построению автоматных программ // Труды межд. конф. «Компьютерные науки и инф. технологии». Саратов, 2009. С. 239–240 (лично автором – 1 с.).
  15. Кузьмин Е. В., Соколов В. А., Чалый Д. Ю. Автоматные счетчиковые машины // Сборник статей международной

- научно-практической конференции «Информационно-коммуникативная культура: наука и образование», Ростов-на-Дону, 2009. С. 321–325 (лично автором – 2 с.).
16. Кузьмин Е. В., Соколов В. А., Чалый Д. Ю. Формальное построение автоматных программ // Труды Третьей Всероссийской научной конференции «Методы и средства обработки информации». Москва, МГУ, 2009. С. 130–135 (лично автором – 2 с.).
  17. Kuzmin E. V., Sokolov V. A., Chaly D. Ju. Automaton Counter Machines // Proc. of Int. Workshop on Program Understanding (sat. of PSI'09), 2009. P. 1–4 (лично автором – 2 с.).
  18. Кузьмин Е. В. Верификация моделей программ. Ярославль: ЯрГУ, 2008. 176 с. (лично автором – 176 с.).
  19. Кузьмин Е. В., Соколов В. А., Чалый Д. Ю. Об одном классе алгоритмических проблем для машин Минского // Сборник материалов международной научной конференции «Образование, наука и экономика в вузах» Польша, Плоцк, 2008. С. 536–537 (лично автором – 1 с.).
  20. Кузьмин Е. В., Виноградов Р. А., Соколов В. А. Система моделирования и анализа автоматных программ. Свидетельство об офиц. рег. программы для ЭВМ №2007611856. Федеральная служба по интеллектуальной собственности, патентам и товарным знакам, 2007.
  21. Кузьмин Е. В. Верификация программ // Сборник мат. IX областной научно-методической конф. «Актуальные проблемы совершенствования подготовки специалистов в вузе». Ярославль, ЯрГУ, 2007. С. 18–21 (лич. автор. – 4 с.).
  22. Кузьмин Е. В., Васильева К. А. Верификация автоматных программ с использованием LTL // Моделирование и анализ информационных систем, 2007. Т. 14, №1. С. 30–42 (лично автором – 6 с.).

23. Кузьмин Е. В., Соколов В. А. О некоторых подходах к верификации автоматных программ // Сборник докладов семинара «Go4IT – шаг к новым технологиям Интернета». Москва, Институт системного программирования, 2007. С. 43–48 (лично автором – 4 с.).
24. Кузьмин Е. В., Соколов В. А. О дисциплине специализации «Верификация программ» // Материалы II научно-методической конференции преподавателей математического факультета и факультета ИВТ «Преподавание математики и компьютерных наук в классическом университете». Ярославль, ЯрГУ, 2007. С. 91–101 (лично автором – 6 с.).
25. Кузьмин Е. В., Соколов В. А. О верификации «автоматных» программ // Сборник статей к 20-летию факультета ИВТ «Актуальные проблемы математики и информатики». Ярославль, ЯрГУ, 2007. С. 31–35 (лично автором – 3 с.).
26. Кузьмин Е. В., Соколов В. А. Структурированные системы переходов. М.: Физматлит, 2006. 178 с. (л. автор. – 6 п. л.).
27. Кузьмин Е. В. Введение в теорию вычислительных процессов и структур. ЯрГУ, 2006. 140 с. (л. автор. – 8 п. л.).
28. Кузьмин Е. В., Виноградов Р. А., Соколов В. А. Верификация автоматных программ средствами CPN/Tools // Моделирование и анализ информационных систем, 2006. Т. 13, №2. С. 4–15 (лично автором – 4 с.).
29. Кузьмин Е. В. Иерархическая модель автоматных программ // Моделирование и анализ информационных систем, 2006. Т. 13, №1. С. 27–34 (лично автором – 8 с.).
30. Кузьмин Е. В., Соколов В. А. Вполне структурированные системы помеченных переходов. М.: Физматлит, 2005. 176 с. (лично автором – 6 п. л.).
31. Кузьмин Е. В., Соколов В. А. Исследование свойств класса вполне структурированных систем переходов // Труды Второй Всероссийской научной конференции «Методы

- и средства обработки информации». Москва, МГУ, 2005. С. 388–393 (лично автором – 4 с.).
32. Kouzmin E. V., Shilov N. V., Sokolov V. A. Model Checking  $\mu$ -Calculus in Well-Structured Transition Systems // Proc. of 11th International Symposium on Temporal Representation and Reasoning, Tatihou, France, IEEE Press, 2004. P. 152–155 (лично автором – 2 с.).
33. Kouzmin E. V., Shilov N. V., Sokolov V. A. Model Checking  $\mu$ -Calculus in Well-Structured Transition Systems // Joint Bulletin of NCC & IIS, Comp. Science, Novosibirsk, 2004. №20. P. 97–107 (лично автором – 6 с.).
34. Кузьмин Е. В., Соколов В. А. Проверка модели для вполне структурированных систем переходов автоматного типа // Мат. Межд. семинара «Распределенные информационно-вычислительные ресурсы и математическое моделирование», Новосибирск, 2004. С. 73–85 (л. автор. – 7 с.).
35. Кузьмин Е. В., Соколов В. А. Взаимодействующие расширяющие процессы // Моделирование и анализ информационных систем, 2004. Т. 11, №2. С. 8–17 (л. автор. – 5 с.).
36. Кузьмин Е. В. О разрешимости задачи проверки модели для автоматной логики и вполне структурированных систем переходов автоматного типа // Моделирование и анализ информационных систем, 2004. Т. 11, №1. С. 8–17 (лично автором – 10 с.).
37. Kouzmin E. V., Sokolov V. A. Communicating Colouring Automata // Proc. of Int. Workshop on Program Understanding (sat. of PSI'03), 2003. P. 40–46 (лично автором – 4 с.).
38. Кузьмин Е. В., Соколов В. А. Проверка свойств вполне структурированных моделей // Мат. Всероссийской научной конференции, посвященной 200-летию Ярославского государственного университета им. П. Г. Демидова, 2003. С. 50–54 (лично автором – 3 с.).

39. Кузьмин Е. В. Недетерминированные счетчиковые машины // Моделирование и анализ информационных систем, 2003. Т. 10, №2. С. 41–49 (лично автором – 9 с.).
40. Кузьмин Е. В. О разрешимости задачи проверки модели для модального  $\mu$ -исчисления и некоторых классов вполне структурированных систем переходов // Моделирование и анализ информационных систем, 2003. Т. 10, №1. С. 50–55 (лично автором – 6 с.).
41. Кузьмин Е. В. Верификация графов потоков данных с использованием символьной проверки модели для CTL // Моделирование и анализ информационных систем, 2001. Т. 8, №1. С. 38–43 (лично автором – 6 с.).
42. Кузьмин Е. В., Рубцов С. А. Модели и инструментальные средства для разработки и анализа графов потоков данных // Материалы межвузовской научно-технической конференции «Управляющие и вычислительные системы», Вологда, ВоГТУ, 2000. С. 96–97 (лично автором – 1 с.).

Подписано в печать 24.03.2011. Формат 60×84/16.

Бумага офсетная. Отпечатано на ризографе.

Тираж 100 экз. Заказ 9/11.

Отдел оперативной полиграфии ЯрГУ  
150000, г. Ярославль, ул. Советская, 14