На правах рукописи

Вирбицкайте Ирина Бонавентуровна

ФОРМАЛЬНЫЕ МОДЕЛИ И АНАЛИЗ КОРРЕКТНОСТИ ПАРАЛЛЕЛЬНЫХ СИСТЕМ И СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание ученой степени
доктора физико-математических наук

Красноярск, 2001

Работа выполнена в Институте систем информатики им. А.П. Ершова Сибирского отделения Российской академии наук

Официальные оппоненты:	доктор физико-математических наук, профессор Касьянов В.Н.
	доктор физико-математических наук, профессор Добронец Б.С.
	доктор технических наук, профессор Марков Н.Г.
Ведущая организация:	Институт программных систем РАН (г. Переславль-Залесский)

Защита состоится “ 20 ” декабря 2001 года в 14 час. 00 мин. на заседании диссертационного совета Д 212.098.03 в Красноярском государственном техническом университете по адресу: 660074, Красноярск, ул. Киренского, 26

С диссертацией можно ознакомиться в читальном зале библиотеки Красноярского государственного технического университета

Автореферат разослан “ 19 ” ноября 2001 г.

Ученый секретарь
специализированного совета Д 212.098.03
кандидат технических наук Вейсов Е.А.

Общая характеристика работы

Актуальность. Современный этап развития теоретической информатики характеризуется бурным ростом активности исследований в области разработки формальных методов спецификации, анализа и моделирования параллельных/распределенных систем, имеющих сложную структурную организацию и функционирующих в режиме реального времени. Разработка корректных систем такого типа — нетривиальная задача, требующая для своего успешного решения проведения комплексных фундаментальных исследований, основанных на различных формальных методах и средствах, которые варьируются в зависимости от класса моделируемых систем, степени детализации их структуры и поведения, а также характера изучаемых проблем. На основе результатов и рекомендаций теоретических исследований ведется поиск и проверка новых архитектурных принципов конструирования параллельных/распределенных систем, изучаются методы распараллеливания алгоритмов и программ, проверяются новые способы организации программ и процессов, обосновываются программные конструкции, вводимые в языки параллельного программирования, отрабатываются методы структурного и семантического анализа параллельных программ и т.д.

В течение трех последних десятилетий теория параллелизма породила большое разнообразие моделей, теорем, алгоритмов и инструментов, предназначенных для спецификации, разработки и верификации параллельных/распределенных систем. Изучаются фундаментальные понятия и законы параллельной обработки информации и на основании обнаруженных закономерностей строятся более частные формальные модели исследуемых объектов, на которых ставятся и решаются прикладные задачи. Исследования ведутся в трех основных направлениях: формальные модели, алгебры процессов, логики процессов.

Среди отечественных исследований в области разработки формальных методов анализа параллельных систем и процессов отметим работы Н.А. Анисимова, О.Л. Бандман, В.А. Вальковского, Ю.Г. Карпова, В.Е. Котова, Р.М. Смелянского, В.А. Соколова, Л.А. Черкасовой.

Таким образом, можно констатировать, что к настоящему моменту уже сложился некоторый, условно говоря, ‘классический’ подход к разработке корректных параллельных систем, который, как хорошо известно, имеет ряд ограничений: существует возможность исследования только систем с простой структурной организацией и конечным числом

состояний; не до конца изучены взаимосвязи между базовыми отношениями на событиях параллельных систем; детально проработана только интерливинговая семантика параллельных процессов; отсутствуют эквивалентностные понятия, отражающие базовые отношения на событиях распределенных систем; не установлены взаимосвязи между различными моделями и подходами (например, семантическими, алгебраическими и логическими); эффективные верификационные алгоритмы (алгоритмы проверки на моделях) разработаны только для темпоральной логики CTL; имеет место значительное снижение эффективности верификации из-за проблемы 'взрыва состояний'; недостаточно проработаны временные аспекты функционирования параллельных систем и т.д.

Поэтому в рамках диссертационной работы была предпринята попытка расширить, обобщить и развить существующий подход с целью преодоления указанных ограничений.

Все вышесказанное говорит об актуальности исследований, проводимых в рамках диссертационной работы.

Цель диссертации состоит в развитии, обосновании и обобщении системы формальных понятий, моделей, методов и средств проектирования корректных параллельных систем и систем реального времени. Достижение цели связывается с решением следующих задач:

1. разработка для нетрадиционных параллельных архитектур широкого спектра формальных моделей и методов их автоматического построения;
2. развитие и совершенствование методов 'компаративной семантики' с целью установления взаимосвязей, классификации и унификации различных поведенческих, алгебраических и логических моделей параллелизма, а также моделей реального времени в семантиках интерливинг/'истинный параллелизм' и линейное/ветвящееся время;
3. увеличение выразительных мощностей формальных средств описания и изучения систем посредством введения дополнительных возможностей, позволяющих рассуждать о параллельных и альтернативных поведениях, а также временных характеристиках функционирования распределенных систем;
4. эффективная верификация поведенческих свойств параллельных систем реального времени, представленных различными временными формальными моделями.

Методы исследований. В рамках данной работы использовались методы и понятия теории графов, теории множеств, теории алгоритмов, теории категорий, математической логики и линейного программирования. В качестве формальных моделей параллелизма применялись различные классы и обобщения структур событий, сетей Петри, сетей-процессов, потоковых схем, аппарат темпоральных логик. Кроме того, использовались различные понятия поведенческих эквивалентностей параллельных процессов, техники временных регионов, временных зон, частичного порядка и раскрутки. В экспериментальных исследований применялись методы системного программирования и теории структур данных.

Научная новизна. В результате выполненных исследований автором разработан оригинальный подход к решению задач спецификации и анализа корректности параллельных систем и систем реального времени. В основе этого подхода лежат разработанные автором различные формальные модели с семантикой 'истинного параллелизма', методы их построения, сравнения, унификации, а также практически реализуемая концепция верификации систем.

Практическая ценность. Разработанные автором диссертации методы могут лечь в основу широкого спектра промышленных программных систем: блоков автоматического распараллеливания в трансляторах и интерпретаторах, систем построения семантических представлений и эквивалентных преобразований параллельных процессов, систем верификации процессов реального времени. Результаты диссертационной работы были успешно реализованы в рамках экспериментальной системы RT-MEC (Real-Time Model and Equivalence Checker), которая поддерживает различные методы спецификации, анализа, верификации параллельных систем и систем реального времени, представленных различными сетевыми моделями с временными характеристиками, и которая является частью системы PEP (Programming Environement based on Petri nets), совместно разрабатываемой несколькими немецкими университетами (г. Хильдесхайма, г. Ольденбурга, г. Мюнхена).

Апробация работы. Основные идеи и конкретные результаты диссертационной работы обсуждались на следующих всесоюзных и международных научных симпозиумах, конференциях и семинарах: Всесоюзная школа-семинар 'Распределенная обработка информации' (Львов, 1987, Львов 1989); Всесоюзная конференция 'Однородные вычислитель-

ные среды и систолические структуры' (Львов, 1990); International Conference of Young Computer Scientists (Beijing, 1993); International Seminar 'Concurrency: Specification and Programming' (Nieborow, Poland, 1993; Berlin 1994; Warsaw, 1999; Berlin, 2000); International Conference 'Formal Methods in Programming and their Applications' (Novosibirsk, 1993); A.P Ershov International Memorial Conferences on Perspectives of System Informatics (Novosibirsk, 1996; Novosibirsk, 2001); International Conference 'CONPAR 94 - VAPP VI' (Linz, Austria, 1994); 4th International Conference on Applied Logics (Irkutsk, Russia, 1995); International Symposium on Fundamentals of Computation Theory (Krakow, Poland, 1997; Iassy, Romania 1999; Riga, 2001); International Workshop on Logic, Language, Information and Computation (Fortaleza, Brazil, 1997; Natal, Brazil, 2000); IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics (Berlin, 1997); International Workshop on Distributed Data Processing (Novosibirsk, 1998); International Workshop on Discrete Event Systems (Cagliari, Italy, 1998); MFCS'98 International Workshop on Concurrency (Brno, Chezh Republic, 1998); International Conference on Parallel Computing Technology (Sankt-Peterburg, 1999; Novosibirsk 2001).

Кроме того, доклады по теме работы были сделаны на ряде семинаров Института информатики Университета г. Хильдесхайма (Германия), Института прикладной математики (г. Гренобль, Франция), Института кибернетики (г. Киев), Киевского государственного университета, Московского государственного университета, Московского электротехнического института, Института программных систем РАН (г. Переславль-Залесский), Института математики СО РАН (г. Новосибирск), Института систем информатики СО РАН (г. Новосибирск), кафедр Новосибирского государственного университета и др.

Публикации. По теме диссертации опубликовано 69 научных работ, в том числе одна монография, 17 работ — в зарубежных периодических изданиях и журналах, 16 — в трудах международных симпозиумов, конференций и семинаров, 7 — в отечественной центральной печати.

Структура и объем работы. Диссертация состоит из введения, трех глав, заключения, списка литературы и приложения.

Содержание работы

Во **введении** обосновывается актуальность рассматриваемых вопросов; формулируются цели исследований, представленных в диссертации; описываются научная новизна результатов и практическая ценность работы; приводится краткое описание содержания диссертации по главам.

Первая глава посвящена формальным моделям потоковых вычислений и методам их автоматического построения.

В **разделе 1.1** дается аналитический обзор состояния исследований в области потоковых вычислений. В частности, рассматриваются общие принципы организации ЭВМ, управляемых потоками данных, описаны механизмы, обеспечивающие корректные реентерабельные вычисления, — блокирование фишек, сигналы подтверждений, копирование кода, тегирование фишек, очереди. Определены подходы к организации потоковых вычислений над структурами данных, рассматриваются особенности и преимущества использования таких структур данных, как потоки и I-структуры. Приведена классификация потоковых ЭВМ, являющаяся развитием и обобщением классификации Вина путем добавления ветвей по механизмам поддержки корректных реентерабельных вычислений и за счет увеличения числа рассматриваемых проектов потоковых систем почти вдвое. В качестве принципов классификации выбраны тип организации процессорного элемента (ПЭ) и тип реализуемых взаимосвязей между ПЭ. Конкретные архитектурные решения продемонстрированы на наиболее известных проектах потоковых систем.

Далее рассматриваются три направления исследований в области высокоуровневого потокового программирования — потоковых (аппликативных), функциональных и традиционных императивных языков. Особое внимание уделено свойствам потоковых языков (Val, Id, Lucid, SISAL): строгой функциональности, единственности присваивания, отсутствия побочных эффектов, ориентации на значения, использовании имен значений и определений имен вместо переменных и присваиваний переменным соответственно. Приведены примеры, иллюстрирующие особенности языков такого типа.

Следует отметить, что широкое применение потоковые системы получили в таких областях, как вычислительные задачи, обработка сигналов, искусственный интеллект, логическое программирование. Привнесение потоковой синхронизации в традиционные многопроцессорные системы предназначено для решения проблем максимального использова-

ния внутреннего программного параллелизма, загрузки оборудования, увеличения отказоустойчивости.

Кроме того, на основе приведенных в диссертации аналитических исследований можно сделать вывод, что наиболее перспективной является организация потоковых вычислений с использованием механизма тегирования фишек, позволяющая достичь наибольших ускорений вычислений и предполагающая эффективное выполнение массовых параллельных вычислений.

В **разделе 1.2** сначала рассматриваются известные формальные модели потоковых вычислений и обосновывается введение семейства интенсиональных моделей потоковых схем с тегированными фишками (п-схем). Затем исследуются поведенческие свойства введенных моделей.

Потоковая схема (далее просто (базовая) схема) характеризуется вершинами и дугами. Вершины делятся на акторы и вершины связи. Имеются четыре типа акторов (операторы, распознаватели, клапаны и акторы модификации тега, т.е. акторы генерации нового тега и акторы восстановления старого тега) и два типа вершин связи (информационные и управляющие). Дуги, соединяющие акторы с вершинами связи и вершины связи с акторами, называются информационными или управляющими дугами в соответствии с типом вершины связи. *Потоковая схема с тегированными фишками (п-схема)* CDS состоит из базовой схемы, множества информационных и управляющих фишек, множества тегов и функции тегирования фишек. *Разметка* п-схемы — это функция, сопоставляющая наборы фишек дугам так, что типы фишек совпадают с соответствующими типами дуг. Разметка называется *начальной* (обозначается M_{in}), если для любой входной дуги п-схемы теги любых двух различных фишек, сопоставленных этой дуге, различны, а всем другим дугам фишки не сопоставлены. Пусть (CDS, M_{in}) обозначает начально размеченную п-схему. *Интерпретация* для (CDS, M_{in}) задается областью значений и сопоставлением каждому оператору функции и каждому распознавателю предиката. Определяются правила срабатывания вершин, понятие процесса и наиболее важные поведенческие свойства (живости, безопасности, ‘очищаемости’) п-схемы.

Далее предлагается алгебраический язык спецификации потоковых вычислений, позволяющий конструировать структурированные п-схемы на основе специального набора примитивных схем и алгебраических операций (параллельной композиции, слияния вершин связи, последовательной композиции, альтернативной композиции, итерационной ком-

позиции). В *теореме 1.2.1* показывается, что структурированные п-схемы обладают свойствами живости, безопасности, 'очищаемости'.

Затем вводятся классы обогащенных (с массивами и с процедурами) п-схем и исследуются их поведенческие свойства. П-схема с массивами $CDSA$ представляет собой расширение п-схемы CDS за счет следующего изменения ее базовой схемы: добавления счетного множества имен массивов данных, которое разбито на конечное множество непересекающихся линейноупорядоченных подмножеств (массивов); разбиения множества информационных фишек на непересекающиеся подмножества: фишек с именами массивов, фишек с индексами, фишек со значениями массивов; аналогичного разбиения множества информационных вершин связи; введения в множество вершин подмножества вершин обработки массивов, состоящего из непересекающихся подмножеств: вершин порождения массивов, вершин модификации значения элемента массива, вершин выбора значения элемента массива. Определения правил срабатывания вершин, процесса и свойств п-схемы CDS являются корректными и для п-схемы с массивами $CDSA$. *Структурированная п-схема с массивами* — это обобщение структурированной п-схемы путем расширения множества примитивных схем схемами с вершинами обработки массивов. Дополнительные структурные ограничения состоят в следующем: для каждого имени массива существует не более одной вершины модификации значения в массиве. В *теореме 1.2.2* устанавливается, что структурированная п-схема с массивами является бесконфликтной, живой, 'очищаемой'.

Базовая схема *п-схемы с процедурами* $CDSP$ состоит из главной схемы и множества схем процедур, с каждой из которых связано некоторое уникальное имя. Главная схема представляет собой расширение базовой схемы п-схемы с массивами за счет введения в множество вершин подмножества вершин вызова процедур, в каждой из которых указывается имя вызываемой процедуры. Схема процедуры — это схема того же типа, что и главная, с тем лишь отличием, что каждая входная дуга процедуры является входной дугой вершины генерации нового тега (уникально идентифицирующего имя процедуры и номер ее вызова), а каждая выходная дуга — выходной дугой вершины восстановления старого тега (который был у фишек до входа в процедуру). Определения правил срабатывания вершин, процесса и свойств п-схемы с массивами $CDSA$ корректны и для п-схемы с процедурами $CDSP$.

Структурированная п-схема с процедурами — это структурирован-

ная главная п-схема и множество структурированных п-схем процедур, для построения которых используется множество примитивных схем структурированной п-схемы с массивами, дополненное схемами, содержащими вершины вызова процедур. В *теореме 1.2.3* показывается, что структурированная п-схема с процедурами является бесконфликтной, живой, 'очищаемой'.

В **разделе 1.3** обсуждается проблема использования и реализации конструкций традиционных императивных языков на потоковых ЭВМ. Формально определяются некоторые существенные атрибуты стандартных операторных схем. Вводится понятие поведенческой эквивалентности с- и п-схемы, основанной на совпадении информационно-логических графов их процессов. Представляется и обосновывается ряд алгоритмов преобразования стандартных схем, схем с массивами, схем с процедурами в соответствующие п-схемы.

Алгоритм $A1$, предназначенный для преобразования ациклической с-схемы SS в потоковый вид, предварительно осуществляет анализ информационно-логических зависимостей в с-схеме и элиминацию логически узловых вершин (т.е. вершин, логически зависимых от нескольких распознавателей) посредством размножения этих вершин и соответствующих им дуг. В процессе преобразования с-схемы выполняется последовательный переход от одной вершины к другой, причем каждая условная конструкция с-схемы рассматривается как одна вершина. При преобразовании вершин ввода и вывода с-схемы строятся соответственно входные и выходные вершины связи п-схемы, а при преобразовании каждого оператора с-схемы и его информационных зависимостей осуществляется построение соответствующего оператора п-схемы и его входных и выходных информационных вершин связи. Преобразование последовательных условных конструкций в потоковый вид осуществляет вспомогательный рекурсивный алгоритм $ACG1$, в котором при преобразовании распознавателя и его информационно-логических связей строятся соответствующий распознаватель п-схемы и его входные информационные и выходная управляющая вершины связи; кроме того, для каждой входной и выходной переменной условной конструкции строится клапан п-схемы и его входные и выходные вершины связи, а при преобразовании операторов условной конструкции и их информационных связей выполняются те же действия, что и в основном алгоритме. В *теореме 1.3.1* устанавливается, что построенная конструкция $A1(SS)$ является структурированной п-схемой, эквивалентной исходной

с-схеме SS.

В алгоритме $A2$, осуществляющем преобразование ациклической с-схемы SS в потоковый вид, элиминации подлежат только логически узловые распознаватели. Для каждого логически узлового оператора определяется множество LP распознавателей, от которых логически зависит данный оператор. При преобразовании вершин с-схемы (за исключением распознавателей) и их информационных зависимостей выполняются те же действия, что и в алгоритме $A1$. При преобразовании условных конструкций с-схемы используется вспомогательный рекурсивный алгоритм $ACG2$, в котором при преобразовании вершин (за исключением логически узловых операторов) условной конструкции и их информационно-логических зависимостей выполняются соответствующие действия алгоритма $ACG1$, а при преобразовании логически узлового оператора построение соответствующей потоковой вершины и ее выходных вершин связи осуществляется при преобразовании первого распознавателя из соответствующего множества LP, а построение ее входных вершин связи — при преобразовании последнего распознавателя из множества LP, что позволяет корректно отображать логические зависимости данного логически узлового оператора. В *теореме 1.3.2* показывается, что построенная конструкция $A2(SS)$ является п-схемой, эквивалентной исходной с-схеме SS.

Алгоритм $A3$ является развитием алгоритма $A1$ на случай с-схемы SS со структурированными вложенными циклическими конструкциями. Предварительно аналогично алгоритму $A1$ осуществляется анализ информационно-логических зависимостей в SS и ее модификация, а затем — преобразование ее элементов в потоковый вид. При этом выполняется последовательный переход от одной вершины к другой, причем каждая циклическая конструкция рассматривается как одна вершина. Преобразование циклических конструкций с-схемы выполняет рекурсивный алгоритм $ALG1$, в котором в дополнение к алгоритму $ACG1$ для каждой входной переменной данной циклической конструкции строится вершина генерации нового тега (уникально идентифицирующего каждую итерацию цикла) и для каждой выходной переменной — вершина восстановления старого тега. В *теореме 1.3.3* устанавливается, что конструкция $A3(SS)$ является структурированной п-схемой, эквивалентной с-схеме SS.

Алгоритм $A4$ осуществляет преобразования с-схемы SS, содержащей циклические конструкции, включающие неструктурированные услов-

ные конструкции, в потоковый вид и является развитием алгоритма $A2$. Предварительно аналогично алгоритму $A2$ осуществляется анализ информационно-логических зависимостей в SS и ее модификация, а затем — преобразование ее элементов в потоковый вид. При преобразовании циклических конструкций выполняются действия алгоритма $ALG1$, а при преобразовании условных конструкций — действия алгоритма $ACG2$. В *теореме 1.3.4* показывается, что построенная конструкция $A4(SS)$ является п-схемой, эквивалентной с-схеме SS.

Алгоритм $A5$ представляет собой развитие алгоритма $A3$ на случай с-схем с массивами SSA. Предварительно наряду с анализом информационно-логических зависимостей и модификацией с-схемы SSA выполняется переименование массивов так, чтобы предотвратить повторные присваивания значений элементам массивов. Дополнительно для каждого имени массива строится вершина порождения массива; каждой вершине модификации значения элемента массива и каждой вершине выбора значения элемента массива сопоставляются соответствующие потоковые вершины. Преобразование информационно-логических зависимостей вершин обработки массивов в потоковый вид требует действий, аналогичных соответствующим действиям алгоритма $A3$ при преобразовании операторов. В *теореме 1.3.5* устанавливается, что построенная конструкция $A5(SSA)$ является структурированной п-схемой, эквивалентной с-схеме SSA.

Алгоритм $A6$ является развитием алгоритма $A5$ на случай с-схем с процедурами SSP. Анализ информационно-логических зависимостей и модификация с-схемы с процедурами, а также преобразование вершин главной п-схемы и их информационно-логических зависимостей в потоковый вид требуют действий, аналогичных действиям алгоритма $A5$. Дополнительно каждой вершине вызова процедуры сопоставляется соответствующая потоковая вершина. Преобразование вершин с-схемы процедуры в потоковый вид осуществляет алгоритм AP. Строгая функциональность и отсутствие побочных эффектов, свойственные потоковым процедурам, требуют явного задания интерфейсов между процедурами, поэтому глобальные переменные с-схем процедур при преобразовании представляются в виде параметров п-схем процедур. В алгоритме AP с каждым входным параметром с-схемы процедуры связываются входная вершина связи и вершина генерации нового тега, а с каждым выходным параметром — выходная вершина связи и вершина восстановления старого тега. Преобразование вершин с-схемы процедуры и их

информационно-логических зависимостей осуществляется так же, как вершин главной с-схемы. В *теореме 1.3.6* показывается, что построенная конструкция $A6(SSP)$ является структурированной п-схемой, эквивалентной с-схеме SSP.

Отметим, что предложенные алгоритмы имеют линейную сложность относительно размера исходной с-схемы. Кроме того, данные алгоритмы были протестированы в рамках ЭСПП (Эсперимента́льная Система Потокового Программирования), которая была реализована автором в системе программирования Барроуз и общий объем которой составил 4 тыс. ПЛ-строк.

Вторая глава посвящена разработке, исследованию и сравнительному анализу семантических моделей параллельных процессов и процессов реального времени.

В **разделе 2.1** в контексте различных классов модели структур событий вводится и исследуется ряд новых вариантов 'аксиом параллельности' (свойств дискретности, плотности и перекрестности), которые позволяют избежать несоответствия между синтаксическим и семантическим представлениями параллельных процессов. Устанавливаются взаимосвязи между исследуемыми свойствами и формулируются необходимые и достаточные условия, гарантирующие выполнение свойств такого типа. Дается алгебраическая характеризация рассматриваемых 'аксиом параллельности'. Отметим, что основным достоинством модели структур событий является то, что в ней явным образом задаются три базовых отношения (причинной зависимости, параллелизма и недетерминированного выбора) между событиями параллельных и распределенных систем.

Данный раздел состоит из двух частей. В первой сначала определяется понятие первичной структуры событий (prime event structure) $\mathcal{E} = (E, \leq, \#)$, где E — множество событий, на котором определены отношение $\leq$ *причинной зависимости* (частичный порядок), удовлетворяющее принципу 'конечности причин', и отношение $\#$ *конфликта* (иррефлексивное и симметричное отношение), удовлетворяющее принципу 'наследования конфликта'. Два события, не связанные ни отношением причинной зависимости, ни конфликта, считаются *параллельными*. *Конфигурацией* структуры называется левозамкнутое (относительно причинной зависимости) подмножество бесконфликтных событий. *Начальная конфигурация* — это пустое множество. Далее вводится ряд определений 'аксиом параллельности'. Свойства K-, N-плотности и K-

перекрестности предназначены для изучения отношений причинной зависимости (li) и параллелизма (co) и основаны на пересечении линий, образованных множествами событий, находящихся в указанных отношениях. Свойства L-плотности и L-перекрестности являются обобщением соответствующих K-свойств на отношения причинной зависимости (li) и недетерминированного выбора (cf). Оказалось, что конечность первичной структуры гарантирует свойства K- и L-'перекрестности', которые являются ослаблением свойств K- и L-плотности соответственно. Установлено, что для всего класса первичных структур событий свойства L-плотности и L-перекрестности совпадают, тогда как совпадение свойств K-плотности и K-перекрестности обеспечивается только при выполнении свойства N-плотности. В отличие от сетей-процессов и частично упорядоченных множеств, рассматриваемая модель первичных структур позволила ввести и изучить ряд новых понятий — свойства R-плотности и N'-плотности, ориентированные на отношения параллелизма (co) и недетерминированного выбора (cf). Устанавливается, что данные свойства совпадают для структур с бинарными отношениями параллелизма и конфликта. Далее вводится свойство M^{li}-плотности (M^{co}-плотности, M^{cf}-плотности), основанное на пересечение двух плоскостей, одна из которых образована отношениями li и co (co и cf, cf и li соответственно), а другая — отношениями li и cf (co и li, cf и co соответственно). Благодаря этому стало возможным установление тесных взаимосвязей между различными формулировками понятия плотности. В частности, в *теореме 2.1.1* для конечных и бесступиковых первичных структур показывается совпадение следующих свойств: K- и M^{cf}-плотности, L- и M^{co}-плотности, а также R- и M^{li}-плотности.

Во второй части раздела данные и новые варианты 'аксиом параллельности' формулируются и изучаются в контексте локальных структур событий (flow event structures), которые являются обобщением первичных структур за счет снятия ограничений 'конечности причин' и 'наследования конфликта'. Данное обобщение позволяет сформулировать унифицированные определения 'аксиом параллельности' (любое отношение рассматривается просто как некоторая 'связка', и поэтому любая 'аксиома параллельности' может быть рассмотрена как представление взаимосвязей между любыми различными 'связками'). В *теореме 2.1.2* дается алгебраическая характеризация свойств плотности и перекрестности локальных структур событий.

В **разделе 2.2** для структурированных потоковых схем с тегиро-

ванными фишками (п-схем) разрабатывается семантика 'истинного' параллелизма в терминах структур событий. Устанавливаются строгие взаимосвязи между данным семантическим представлением и графом причинной зависимости (з-граф) — некоторой модификацией традиционного понятия информационно-логического графа — с точностью до изоморфизма.

Сначала для каждого процесса ρ (последовательности срабатываний вершин) из множества процессов $\mathbf{R}$ начально размеченной структурированной п-схемы (CFS, M_{in}) строится *з-граф*, $G_\rho((CFS, M_{in})) = (V_\rho, E_\rho, l_\rho)$, где V_ρ — множество вершин, соответствующих срабатываниям из ρ, E_ρ — множество дуг, отражающих причинные зависимости между срабатываниями из ρ, $\widehat{l}_\rho$ — функция, помечающая дуги графа соответствующими срабатываниями. Вводится понятие *проекции з-графа* на множество срабатываний операторов и распознавателей — $\widehat{G}_\rho = (\widehat{V}_\rho, \widehat{E}_\rho, \hat{l}_\rho)$, которое позволяет обобщить несущественные зависимости в з-графе. Пусть $\mathbf{G}$ обозначает множество конечных проекций з-графов для начально размеченной п-схемы (CFS, M_{in}). На множестве $\mathbf{G}$ вводится отношение частичного порядка $\subseteq' \subseteq \mathbf{G} \times \mathbf{G}$ обычном образом. Далее на основе множества $\mathbf{G}$ строится структура $\mathcal{E}((CDS, M_{in})) = (E, \leq, \#, l)$, где $E = \cup_{\rho\in\mathbf{R}}(\widehat{V}_\rho)$; $\leq = \cup_{\rho\in\mathbf{R}}(\widehat{E}_\rho)^*$; любые два события e и e' находятся в отношении конфликта ($e \# e'$), если и только если e и e' одновременно не принадлежат множеству вершин ни одного з-графа из $\mathbf{G}$; $l = \cup_{\rho\in\mathbf{R}}\widehat{l}_\rho$. Доказано, что $\mathcal{E}((CDS, M_{in}))$ — помеченная структура событий. Пусть $\mathbf{C}$ обозначает множество конечных конфигураций в $\mathcal{E}((CDS, M_{in}))$. В *теореме 2.2.1* показано, что $(\mathbf{G}, \subseteq')$ и $(\mathbf{C}, \subseteq)$ — изоморфные множества, а в *теореме 2.2.3* установлено, что $(\mathbf{G}, \subseteq')$ и $(\mathbf{C}, \subseteq)$ — когерентные, первично алгебраические и ограниченные частично-упорядоченные множества.

В **разделе 2.3** в контексте помеченных первичных структур событий вводятся различные бисимуляционные эквивалентности, учитывающие все базовые отношения (причинной зависимости, параллелизма и конфликта) между событиями структур. Устанавливаются взаимосвязи между вновь введенными и известными ранее бисимуляциями, также дается логическая характеризация всех рассмотренных эквивалентностей. Кроме того, предлагаются алгоритмы распознавания эквивалентностей для конечных структур событий.

Пусть $\mathcal{E}$ — структура событий, помеченная над множеством действий Act, и $\mathcal{C}(\mathcal{E})$ — множество ее конфигураций. Определим ряд от-

ношений на конфигурациях $C, C' \in \mathcal{C}(\mathcal{E})$: а) $C \xrightarrow{p}_{\varepsilon} C'$ (отношение причинности), если $C \subseteq C'$ и $C' \setminus C = p$ (p — частично упорядоченное мультимножество над Act); б) $C \not\frown_{\varepsilon} C'$ (отношение конфликта), если не существует такой конфигурации C'', что $C \subseteq C''$ и $C' \subseteq C''$; в) $C \uparrow'_{\varepsilon} C'$ (отношение параллелизма), если не выполняется ни одно из условий: $C \subseteq C'$, $C' \subseteq C$ и $C \not\frown_{\varepsilon} C'$.

Интерливинговой бисимуляцией (обозначается $\approx_i$) между структурами событий $\mathcal{E}$ и $\mathcal{F}$ называется отношение $\mathcal{B} \subseteq \mathcal{C}(\mathcal{E}) \times \mathcal{C}(\mathcal{F})$ такое, что пара, состоящая из начальных конфигураций данных структур, принадлежит $\mathcal{B}$, и для всех пар $(C, D) \in \mathcal{B}$ выполняется: а) если $C \xrightarrow{a}_{\varepsilon} C'$, то найдется конфигурация D' в $\mathcal{F}$ такая, что $D \xrightarrow{a}_{\mathcal{F}} D'$ и $(C', D') \in \mathcal{B}$; б) то же верно для конфигурации D. При этом если для всех $(C, D) \in \mathcal{B}$ сужение $\mathcal{E}$ на множество C изоморфно сужению $\mathcal{F}$ на множество D, то $\mathcal{B}$ называется *сохраняющей историю бисимуляцией* (обозначается $\approx_h$). Пусть $\alpha \in \{i, h\}$. В определении обратных бисимуляций (обозначаются $\approx_{\alpha b}$) вместо пары $(C, D) \in \mathcal{B}$ рассматривается пара $(C', D') \in \mathcal{B}$ и наоборот, отношение причинности на конфигурациях рассматривается в обратном направлении. В определении сохраняющих параллелизм бисимуляций (обозначаются $\approx_{\alpha c}$) вместо отношения причинности рассматривается отношение параллелизма на конфигурациях, а в определении сохраняющих конфликт бисимуляций (обозначаются $\approx_{\alpha a}$) — отношение конфликта на конфигурациях. Для каждого события структуры вводится понятие *локальной конфигурации* как множества событий, являющихся предшественниками (по причинной зависимости) данного события.

Пусть $\beta \in \{a, b, c\}^*$. Определяя аналогичные эквивалентностные отношения на множествах локальных конфигураций структур событий $\mathcal{E}$ и $\mathcal{F}$, получаем $l\beta$-бисимуляции (обозначаются $\approx_{l\beta}$). Структура событий называется *структурой без автопараллелизма*, если любые два одинаково помеченных события не параллельны; — *структурой с конечным автоконфликтом*, если любое множество одинаково помеченных ее событий, состоящих в конфликте друг с другом, конечно. В этом разделе мы ограничимся рассмотрением структур без автопараллелизма и с конечным автоконфликтом. В *утверждениях 2.3.1-2.3.3* и *теореме 2.3.1* устанавливаются следующие равенства и строгие включения:

а) $\approx_{h\beta'} \subset \approx_{i\beta'}$, $\approx_{ib\beta'} = \approx_{hb\beta'}$, $\approx_{hb\beta''} = \approx_{hc\beta''}$ ($\beta' \in \{a, c\}^*$, $\beta'' \in \{a\}^*$);

б) $\approx_{\alpha\beta' a} \subset \approx_{\alpha\beta'}$, $\approx_{\alpha\beta'' c} \subset \approx_{\alpha\beta''}$ ($\beta' \in \{b, c\}^*$, $\beta'' \in \{a\}^*$);

в) $\approx_{h\beta} \subset \approx_{l\beta}$; $\approx_{la\beta'} \subset \approx_{l\beta'}$, $\approx_{lb\beta''} \subset \approx_{l\beta''}$, $\approx_{lc\beta'''} \subset \approx_{l\beta'''}$ ($\beta' \in \{b, c\}^*$, $\beta'' \in$

$\{a,c\}^*$, $\beta''' \in \{a,b\}^*$).

С целью логической характеризации введенных выше бисимуляционных эквивалентностей на конфигурациях определяются расширения логики ветвящегося времени CTL^* посредством введения новых операторов: CTL_b^* с операторами ветвящегося прошлого, CTL_c^* с оператором параллелизма, CTL_a^* с оператором конфликта и CTL_{abc}^* — комбинация указанных логик, которая отражает все отношения (причинную зависимость, параллелизм и конфликт) между событиями систем. Пусть $\beta \in \{a,b,c,abc\}$. Нам понадобятся два типа логик: CTL_β^*, в которых в качестве множества атомарных пропозиций рассматривается множество всех частично-упорядоченных мультимножеств над Act, и ${}_0CTL_\beta^*$, в которых в качестве атомарных пропозиций используются только действия из Act. В записи: ${}_\gamma CTL_\beta^*$, где $\gamma \in \{.,0\}$, символ '.' обозначает 'пусто'. *Вычислением* π в структуре событий $\mathcal{E}$ называется максимальная последовательность конфигураций $C_0C_1\ldots$ такая, что $C_i \xrightarrow{a}_\varepsilon C_{i+1}$ для всех $i \geq 0$. Формально определяется выполнимость ${}_\gamma CTL_\beta^*$-формулы λ на вычислении π в момент времени $n = 0,1,\ldots$ (обозначается $\pi,n \models_{{}_\gamma CTL_\beta^*} \alpha$). Используем $\mathcal{E} \models_{{}_\gamma CTL_\beta^*} \alpha$ для обозначения того, что $\pi,n \models_{{}_\gamma CTL_\beta^*} \alpha$ для любого вычисления π в $\mathcal{E}$ и любого момента времени n. Для структур событий $\mathcal{E}$ и $\mathcal{F}$ *модальная эквивалентность*, порождаемая логикой ${}_\gamma CTL_\beta^*$ (обозначается $\mathcal{E} \sim_{{}_\gamma CTL_\beta^*} \mathcal{F}$), определяется следующим образом: $\mathcal{E} \sim_{{}_\gamma CTL_\beta^*} \mathcal{F} \stackrel{def}{\iff} (\mathcal{E} \models_{{}_\gamma CTL_\beta^*} \lambda \iff \mathcal{F} \models_{{}_\gamma CTL_\beta^*} \lambda$ для всех${}_\gamma CTL_\beta^*$-формул λ). В *теореме 2.3.1* устанавливается следующее:

а) $\mathcal{E} \approx_{i\beta} \mathcal{F} \iff \mathcal{E} \sim_{{}_0CTL_\beta^*} \mathcal{F}$;

б) $\mathcal{E} \approx_{h\beta} \mathcal{F} \iff \mathcal{E} \sim_{CTL_\beta^*} \mathcal{F}$.

Затем в данном разделе описывается темпоральная логика L_1, введенная Мукундом и Тиагараджаном для логических спецификаций первичных структур событий. Отличительной чертой этой логики является то, что в ней наряду со стандартными темпоральными модальностями будущего и прошлого используются модальности параллелизма и недетерминированного выбора. Для наших целей удобно в качестве множества атомарных пропозиций взять множество действий Act. Семантика логики L_1 интерпретируется с использованием локальных конфигураций. Определяется понятие *выполнимости* L_1-формулы λ на локальной конфигурации $\downarrow e$ (обозначается $\downarrow e \models_{L_1} \lambda$). Модальная эквивалентность, порождаемая логикой L_1 (обозначается $\sim_{L_1}$), определяется

следующим образом: $\mathcal{E} \sim_{L_1} \mathcal{F} \stackrel{def}{\iff} (\mathcal{E} \models_{L_1} \alpha \iff \mathcal{F} \models_{L_1} \alpha$ для любой L_1-формулы α). В *теореме 2.3.2* устанавливается следующее:

$$\mathcal{E} \sim_{L_1} \mathcal{F} \iff \mathcal{E} \approx_{labc} \mathcal{F}.$$

Далее решается проблема распознавания указанных выше эквивалентностей для конечных структур событий. С этой целью приводятся алгоритмы 'на лету' распознавания как обычных, так и локальных бисимуляций, показывается корректность данных алгоритмов и дается оценка их сложности.

В **разделе 2.4** сначала вводится модель структур событий с глобальным непрерывным временем, а затем определяются временные трассовые и бисимуляционные эквивалентности, основанные на трех семантиках: последовательных действий, мультимножеств действий, частично-упорядоченных мультимножеств действий. Далее исследуется поведение временных структур с учетом присущего им параллелизма и недетерминизма. Устанавливаются взаимосвязи между рассмотренными тремя семантиками как на всем классе временных структур событий, так и на его подклассах.

Введенная модель временных структур событий расширяет модель первичных структур добавлением временных ограничений (интервалов) на событиях, обозначающих моменты глобального времени, в которые эти события могут произойти. Пара, состоящая из конфигурации структуры и временной функции, записывающей для каждого события момент глобального времени, в который данное событие произошло, является *временной конфигурацией*, если выполнены следующие условия: а) событие может произойти только в тот момент времени, когда его временные ограничения выполнены; б) для всех произошедших событий e и e' если e предшествует e' по причинной зависимости, тогда e также должно предшествовать e' по времени; в) выполнение одних событий не должно препятствовать по времени выполнению других событий, за исключением таких событий, конфликтные события которых успели произойти раньше. *Начальная временная конфигурация* состоит из начальной конфигурации и нулевой временной функции.

В *интерливинговой семантике* временная структура событий 'функционирует' посредством смены одной временной конфигурации другой при выполнении временного действия. Две временные структуры событий *интерливингово временно трассово эквивалентны*, если их интерливинговые временные языки (множества временных слов) совпадают; *временно бисимуляционно эквивалентны*, если существует от-

ношение между их временными конфигурациями, которому принадлежат начальные временные конфигурации, и все временные конфигурации, полученные выполнением временных действий из временных конфигураций, принадлежащих данному отношению, также принадлежат этому отношению. В *шаговой семантике* смена одной временной конфигурации другой происходит при выполнении мультимножеств временных действий (шагов). Используя шаговое отношение смены временных конфигураций, получаем шаговую временную трассовую эквивалентность и шаговую временную бисимуляцию точно так же, как соответствующие интерливинговые эквивалентности. Шаговая временная бисимуляция очевидно сильнее и интерливинговой временной бисимуляции, и шаговой временной трассовой эквивалентности. Определяется понятие *временного частично упорядоченного множества* как временная бесконфликтная структура событий с точечными временными интервалами. Изоморфные классы временных частично упорядоченных множеств называются *временными частично упорядоченными мультимножествами*. Рассматривается отношение смены временных конфигураций посредством выполнения временного частично упорядоченного мультимножества. Используя данное отношение смены состояний, получаем частично упорядоченную временную трассовую эквивалентность и частично упорядоченную временную бисимуляционную эквивалентность аналогично соответствующим интерливинговым эквивалентностям. Очевидно, временная частично упорядоченная бисимуляция сильнее и шаговой временной бисимуляции, и частично упорядоченной временной трассовой эквивалентности. Общая схема определения временных поведенческих эквивалентностей в контексте временных структур событий позволяет изучать взаимосвязи между тремя введенными семантиками. В *теоремах 2.4.2—2.4.4* показано следующее: для структур, которые представляют временные последовательные процессы, все три семантики совпадают в контексте как временной трассовой эквивалентности, так и временной бисимуляции; для детерминированных временных структур событий верно, что шаговая и частично-упорядоченная семантики временной трассовой и временной бисимуляционной эквивалентностей совпадают; для бесконфликтных временных структур событий верно, что временная трассовая эквивалентность совпадает с временной бисимуляцией в контексте соответствующих друг другу семантик.

Третья глава посвящена разработке формального подхода к про-

верке корректности параллельных систем реального времени, представленных различными временными сетевыми моделями, с использованием темпоральных логик реального времени и временных эквивалетностных понятий.

В **разделе 3.1** разрабатываются методы 'проверки на модели' (model checking) с использованием аппарата темпоральных логик реального времени. Сначала предлагается базовый метод верификации поведенческих свойств систем реального времени, представленных сетями Петри с непрерывным временем, основанный на темпоральной логике реального ветвящегося времени $TCTL$. С целью повышения эффективности метода применяются техники частичных порядков, зон и параметризации. Дается оценка сложности и показывается корректность предложенных алгоритмов.

В начале раздела приведены определения, связанные с понятием временной сети Петри $\mathcal{TN} = (P, T, F, D, M_0)$, которая характеризуется непересекающимися конечными множествами мест P и переходов T, отношением инцидентности $F \subseteq (P \times T) \cup (T \times P)$, начальной разметкой M_0 и временной функцией D, сопоставляющей каждому переходу временной интервал. Ограничимся рассмотрением однобезопасных временных сетей. Во временной сети смена одного состояния другим осуществляется либо при истечении некоторого времени, либо при срабатывании некоторого перехода. Поведение временной сети моделируется путями – последовательностями состояний, связанных срабатыванием переходов или истечением времени.

Далее приводятся синтаксис и семантика известной темпоральной логики реального времени $TCTL$. Данная логика является расширением языка ветвящегося времени CTL за счет добавления временных ограничений на его операторы. Семантика $TCTL$ определяется на состояниях и путях временной сети.

Поскольку понятие временной сети Петри базируется на модели непрерывного времени, то число состояний любой временной сети бесконечно. Чтобы получить конечное представление сетевого поведения, вводится понятие региона (аналог региона Алура). Два состояния временной сети принадлежат одному и тому же региону, если они в некотором смысле эквивалентны, т.е. их разметки совпадают и значения соответствующих счетчиков согласованы по целым частям и порядку дробных частей. В качестве конечного представления поведения временной сети Петри $\mathcal{TN}$ при анализе $TCTL$-формулы ϕ строим граф ре-

гионов $G(\mathcal{TN}, \phi)$ с вершинами, соответствующими регионам, и дугами, соответствующими срабатываниям переходов или истечению времени. Размер $G(\mathcal{TN}, \phi)$ экспоненциален относительно размера $\mathcal{TN}$.

Алгоритм верификации $TCTL$-формулы ϕ на временной сети $\mathcal{TN}$ состоит в построении графа регионов $G(\mathcal{TN}, \phi)$ и пометке его вершин подформулами формулы ϕ или их отрицанием. Устанавливается корректность алгоритма пометки. В *теореме 3.1.2* утверждается, что существует алгоритм, проверяющий, удовлетворяет ли временная сеть Петри $\mathcal{TN}$ $TCTL$-формуле ϕ, который линеен по длине ϕ и размеру $G(\mathcal{TN}, \phi)$ (и, следовательно, экспоненциален по размеру $\mathcal{TN}$).

В следующем пункте описывается использование техники частичных порядков для редукции числа анализируемых состояний временной сети Петри. Данный метод редукции использует тот факт, что многие свойства не 'чувствительны' к порядку, в котором выполняются параллельные переходы временной сети, что позволяет избежать конструирования эквивалентных состояний (т.е. состояний, достижимых срабатыванием различных интерливинговых последовательностей переходов). Особенность предложенной редукции состоит в учете как параллелизма сети, так и существенности временных ограничений при проверке выполнимости заданного свойства. Идея редукции состоит в следующем: для каждой вершины v строящегося редуцированного графа регионов $G_R(\mathcal{TN}, \phi)$ при порождении следующих вершин рассматриваются не все срабатывающие переходы (как при построении $G(\mathcal{TN}, \phi)$), а их подмножество и, кроме того, истечение времени, если в вершине v время существенно для данной вершины и формулы ϕ. Вводится понятие временной статтеринг-эквивалентности для графов регионов, позволяющее осуществлять их корректную редукцию. В *теореме 3.1.3* для заданных временной сети Петри $\mathcal{TN}$ и $TCTL$-формулы ϕ показывается, что графы регионов $G(\mathcal{TN}, \phi)$ и $G_R(\mathcal{TN}, \phi)$ статтеринг-эквивалентны. Таким образом, алгоритм пометки для графа регионов $G(\mathcal{TN}, \phi)$ сводится к алгоритму пометки для редуцированного графа $G_R(\mathcal{TN}, \phi)$. Показано, что редукционная процедура полиномиальна относительно размера временной сети Петри.

В следующем пункте раздела делается попытка редуцировать число анализируемых сетевых состояний с использованием техники зон. Вводится понятие временной сети с зонами $\mathcal{ZN} = (P, T, F, Z, M_0)$, которая является модификаций временной сети Петри за счет использования функции Z, связывающей с каждым переходом зону временной задерж-

ки его срабатывания. *(Временной) зоной* Z называется многогранник из $\mathbf{R}^n$ ($n \in \mathbf{N}$), содержащий все решения конечного числа некоторых линейных неравенств. Как и во временной сети Петри, так и во временной сети с зонами, смена одного состояния другим осуществляется либо при истечении некоторого времени, либо при срабатывании некоторого перехода сети. Вводятся понятие обобщенного состояния, являющееся аналогом региона, но с использованием временных зон, и понятие стабильного разбиения ρ сетевых состояний на обобщенные состояния, которое означает, что все состояния из обобщенного состояния, принадлежащего ρ, эквивалентны относительно их достижимости. Таким образом, проблема достижимости состояний может быть сведена к проблеме достижимости обобщенных состояний при стабильном разбиении. На вход верификационного алгоритма поступают временная сеть с зонами $\mathcal{ZN}$, начальное разбиение ρ_0 множества состояний временной сети с зонами и свойство сети, представленное $TCTL$-формулой ϕ. Суть данного алгоритма состоит в следующем: строится конечное графовое представление сетевого поведения — граф обобщенных состояний при начальном разбиении ρ_0, $G(\mathcal{ZN}, \rho_0)$; вычисляется множество $F(\phi)$ вершин данного графа, для которых формула ϕ истинна. Результатом работы данного верификационного алгоритма является ответ на вопрос, принадлежит ли начальное обобщенное состояние при разбиении ρ_0 множеству $F(\phi)$. Устанавливается корректность данного алгоритма и дается оценка его сложности.

Далее определяется понятие *параметрической временной сети* $\mathcal{PN} = (P, T, F, \mathcal{T}, M_0)$, которая является модификацией временной сети Петри за счет введения функции $\mathcal{T}$, сопоставляющей каждому переходу некоторый *временной предикат*, индуктивно определяемый следующим образом: $\eta = false \mid t \sim \theta \mid \eta_1 \to \eta_2$, где θ – параметр или натуральное число, t – переход в $\mathcal{PN}$, η_1 и η_2 – временные предикаты и $\sim$ — одно из бинарных отношений $<, \leq, =, \geq, >$. Ограничимся рассмотрением однобезопасных параметрических временных сетей. Вводится понятие *означивания* как функции, сопоставляющей каждому параметру некоторое натуральное число. Означивание называется *c-ограниченным* для некоторого натурального числа c, если оно сопоставляет каждому параметру значение, не превышающее c. Через $\mathcal{PN}^\chi$ обозначим параметрическую временную сеть $\mathcal{PN}$ с означенными посредством χ параметрами. *Задача анализа* параметрической временной сети $\mathcal{PN}$ относительно $TCTL$-формулы ϕ, $TBA(\mathcal{PN}, \phi)$, состоит в нахождении такого означи-

вания χ, что формула ϕ выполняется в $\mathcal{PN}$ при означивании χ.

Граф регионов G^χ для $\mathcal{PN}^\chi$ определяется так же, как и для временной сети Петри. Граф регионов $\mathbf{G}$ параметрической временной сети $\mathcal{PN}$ строится как объединение графов регионов G^χ по всем возможным $c_{\mathcal{PN}:\phi}$-ограниченным означиваниям χ ($c_{\mathcal{TN}:\phi}$ — максимальная константа, появляющаяся во временных ограничениях параметрической сети и логической формулы). Для определения временной длительности путей в графе вводится понятие кактус-структуры, которая представляет собой набор простых циклов, связанных с некоторым простым путем. Данное понятие позволяет выразить время произвольного пути в графе регионов через времена простых путей и циклов. Алгоритм решения задачи $TBA(\mathcal{PN}, \phi)$ заключается в построении графа регионов G и пометке пар (v, ϕ') (v – вершина графа G и ϕ' – некоторая подформула формулы ϕ) некоторой формулой логики первого порядка $L(v, \phi')$, называемой *условием*, с параметрами в качестве свободных переменных. Будем говорить, что χ является решением задачи $TBA(\mathcal{PN}, \phi)$, если для начальной вершины v_0 графа G означивание χ удовлетворяет условию $L(v_0, \phi)$. В *теореме 3.1.4* показана корректность данного алгоритма анализа, а в *теореме 3.1.5.* устанавливается существование алгоритма решения задачи $TBA(\mathcal{PN}, \phi)$, который линеен по размеру ϕ и дважды экспоненциален по размеру $\mathcal{PN}$.

В **разделе 3.2** в рамках различных сетевых моделей решаются проблемы распознавания временных трассовой, тестовой и бисимуляционной эквивалетностей.

Сначала данные эквивалентности определяются и исследуются в контексте вновь введенного семантического понятия — сетевого процесса с глобальными непрерывными временными характеристиками (временного процесса). В контексте данной модели вводится понятие графа регионов (дискретного представления поведения временного процесса), а на его основе — понятие графа классов (детерминированного представления поведения временного процесса). На графах классов двух временных процессов определяются традиционные отношения *бисимуляции* и П-*бисимуляции*. В теореме *3.2.1* показывается, что два временных процесса являются временно трассово эквивалентными (временно тестово эквивалентными), если и только если их графы классов бисимуляционно (П-бисимуляционно) эквивалентны. Тогда алгоритм распознавания временно трассовой эквивалентности (временно тестовой эквивалентности) состоит из следующих шагов: построения графа регио-

нов, построения графа классов и анализа бисимуляционной (П-бисимуляционной) эквивалентности по известному из литературы алгоритму. Показано, что сложность алгоритмов анализа указанных эквивалентностей экспоненциальна относительно размера временных процессов.

Проблема распознавания временной бисимуляционной эквивалентности в контексте временных процессов решается с использованием методов теории категорий. Сначала определяется категория временных процессов и выделяется подкатегория конечных временных слов процессов. Затем в рамках категории временных сетей вводится традиционное понятие теории категорий — понятие TW-открытого морфизма. В теореме *3.2.5* формулируется критерий TW-открытости, а в теореме *3.2.8* показывается разрешимость данного понятия с применением техники регионов. Далее определяется вспомогательное подобие по временным словам и показывается его разрешимость с использованием разрешимости TW-открытости морфизмов; доказывается совпадение подобия по временным словам и временной бисимуляции, что позволяет решить проблему распознавания последней.

Также предлагается решение рассматриваемых проблем в контексте дискретно-временных сетей Петри (ДВСП). Для этих целей сначала вводится понятие корректного таймирования, позволяющего корректно отображать временные ограничения переходов в ДВСП на переходы раскрутки МакМиллана (конечного представления сетевого поведения), что приводит к понятию временной экспансии, которая, как показывается, является временным процессом. Две дискретно-временные сети Петри являются *временно трассово эквивалентными* (*временно тестово эквивалентными*, *временно бисимуляционными*), если временные экспансии их раскруток МакМилана временно трассово эквивалентны (временно тестово эквивалентны, временно бисимуляционны). Алгоритм распознавания временной трассовой эквивалентности (временной тестовой эквивалентности, временной бисимуляции) состоит из следующих шагов: 1) конструирования раскруток МакМиллана базовых сетей ДВСП; 2) вычисления множеств корректных таймирований сконструированных раскруток; 3) конструирования временных экспансий данных раскруток на основе корректных таймирований; 4) распознавания временной бисимуляции сконструированных временных экспансий. Дается оценка сложности и показывается корректность предложенного алгоритма.

В **разделе 3.3** сначала представляется краткий аналитический об-

зор инструментальных систем, базирующихся на сетевых моделях. При этом обосновываются достоинства ряда программных комплексов, среди которых отмечается система PEP (Programming Environment based on Petri nets). Предлагается программный комплекс RT-MEC (Real-Time Model and Equivalence Checher), поддерживающий спецификацию, валидацию и верификацию параллельных систем реального времени, представленных различными моделями временных сетей Петри, и функционирующий в составе системы PEP. Рассматриваются структура и функции предлагаемого программного комплекса RT-MEC. Особенное внимание уделено модулям, поддерживающим анализ, симуляцию и верификацию временных расширений сетей Петри. В комплексе RT-MEC анализ осуществляется посредством исследования структуры моделируемой системы, валидация — путем имитационного моделирования (симуляции), а верификация — с помощью как 'проверки на модели', так и проверки на поведенческую эквивалентность. Отличительная черта комплекса RT-MEC состоит в том, что делается попытка объединить в единое целое различные методы анализа и верификации распределенных систем реального времени с возможностью последующего их сравнения. Также описываются состояние реализации и результаты экспериментов, проведенных средствами RT-MEC.

В **заключении** перечисляются основные результаты, полученные в рамках диссертационной работы.

В **приложении** приводятся формальная модель последовательных вычислений и указатель символов, используемых в тексте диссертации.

Результаты исследований, изложенные в диссертации, легли в основу ряда научно-исследовательских проектов, поддержанных в разные годы различными грантами Российского фонда фундаментальных исследований (гранты 93-01-00986, 96-01-01655, 00-01-00898), Фондом фирмы Фольксваген (грант I/70 564), Фондом ИНТАС (грант 1010-CT93-0048), Фондом ИНТАС-РФФИ (грант 95-0378) и др.

Диссертация содержит результаты работ, выполненных автором в Вычислительном центре СО РАН с 1986 по 1990 гг. и в Институте систем информатики СО РАН с 1990 по 2001 гг. В целом, результаты, изложенные в диссертации, получены автором самостоятельно. Исключение составляет следующее: в разработке модели сетей Петри с временными параметрами и алгоритма верификации временных сетей Петри

с использованием техники частичных порядков (см. раздел 3.2) принимала участие аспирантка НГУ Е.А. Покозий, а также система RT-MEC (см. раздел 3.3) была реализована при участии с.н.с. ИСИ СО РАН А.В. Быстрова и группы студентов ММФ НГУ.

Основные выводы и результаты

Перечислим основные результаты, полученные в рамках диссертационной работы.

- Для параллельных систем, управляемых потоками данных, разработан широкий спектр формальных моделей и предложены алгоритмы автоматического их построения:
  - дан аналитический обзор состояния исследований в области потоковых архитектур, выполнено сравнение возможностей языков потокового программирования, а также проанализированы известные формальные модели вычислений потокового типа;
  - введена формальная модель тегированных потоковых схем (п-схем), определены наиболее важные их поведенческие свойства (живости, безопасности и 'очищаемости');
  - предложен алгебраический язык спецификации структурированных п-схем, показано, что п-схемы из данного класса обладают свойствами живости, безопасности, 'очищаемости';
  - введены различные классы обогащенных (с массивами и с процедурами) п-схем и исследованы их поведенческие свойства;
  - обсуждена проблема использования и реализации конструкций традиционных императивных языков на потоковых процессорах, представлен и обоснован ряд алгоритмов преобразования стандартных схем, схем с массивами, схем с процедурами в соответствующие классы п-схем.
- Предложен оригинальный подход к разработке и сравнительному анализу семантических моделей параллельных процессов и процессов реального времени, основную идею которого раскрывают следующие результаты:
  - для разработки семантических моделей, адекватно представляющих реальные параллельные процессы, даны унифицированные определения и введены новые варианты 'аксиом

параллельности' (свойств дискретности, плотности и перекрестности) в контексте различных моделей структур событий; установлена иерархия взаимосвязей этих аксиом; сформулированы необходимые и достаточные условия, гарантирующие выполнение данных свойств, а также дана их алгебраическая характеризация;
  - для структурированных потоковых сетей с тегированными фишками введена семантика 'истинного параллелизма' в терминах структур событий и установлены взаимосвязи между данным семантическим представлением и графом зависимостей (модификацией традиционного семантического представления — информационно-логического графа) с точностью до изоморфизма;
  - на структурах событий исследованы новые варианты бисимуляционных эквивалентностей, отражающие базовые отношения (причинной зависимости, параллелизма и конфликта) на событиях структур; построена иерархия взаимосвязей всех введенных и известных ранее эквивалентностных понятий; дана логическая характеризация рассмотренных эквивалентностей, а также показана их разрешимость для конечных структур;
  - введена и исследована модель структур событий с глобальным непрерывным временем, разработаны временные эквивалентные отношения в семантиках интерливинг/'истинный параллелизма' и линейное/ветвящееся время; установлены взаимосвязи между данными эквивалентностями на всем классе и некоторых подклассах рассматриваемой модели.
- Разработана и практически реализована концепция анализа корректности параллельных систем реального времени, представленных различными временными сетевыми моделями, с использованием аппарата темпоральных логик реального времени и временных эквивалентностных отношений:
  - предложены алгоритмы верификации систем реального времени, представленных сетевыми моделями с непрерывным временем. Временные сетевые свойства представляются в виде формул известной темпоральной логики реального времени $TCTL$. Продемонстрирована возможность применения техники частичных порядков для редукции числа анализи-

руемых сетевых состояний. Даны оценки сложности предложенных алгоритмов и доказана их корректность;
- с целью повышения эффективности верификационных алгоритмов введены и исследованы новые модели параллельных систем реального времени — временные сети с зонами и параметрические временные сети, а также предложены и оценены алгоритмы поведенческого анализа данных моделей;
- решены проблемы распознавания временных трассовой, тестовой и бисимуляционной эквивалентностей в контексте конечного варианта вновь разработанной временной семантической модели — временных процессов (сетевых процессов с глобальным непрерывным временем). Распознавание временной трассовой и временной тестовой эквивалентностей осуществляется с использованием техник графов регионов, детерминированных графов и сведения указанных эквивалентностей к соответствующим вариантам бисимуляционной эквивалентности на детерминированных графах временных процессов. Распознавание временных бисимуляционных эквивалентностей базируется на методах теории категорий (понятии открытых морфизмов). Предложены алгоритмы распознавания указанных временных эквивалентностей для дискретно-временных сетей Петри с использованием вновь введенного понятия временной экспансии раскрутки МасМиллана. Показана корректность предложенных алгоритмов и даны оценки их сложности;
- разработана и программно реализована экспериментальная система RT-MEC, поддерживающая методы спецификации, анализа, верификации параллельных систем и систем реального времени, представленных различными моделями временных сетей Петри. Средствами RT-MEC проведены эксперименты, демонстрирующие целесообразность и эффективность предложенных формальных методов.

Основные публикации по теме диссертации

1. Вирбицкайте И.Б. О программной реализации системы имитации потоковых вычислений // Тр. VI Всесоюз. шк.-сем. "Распределенная обработка информации". – Львов, 1987. – С. 20–22.
2. Вирбицкайте И. Б. Обработка последовательных фрагментов на

устройствах потокового типа // Тр. VII Всесоюз. шк-сем. "Распределенная обработка информации". – Львов, 1989. – С. 44–45.
3. Вирбицкайте И. Б. ЭВМ, управляемые потоками данных. – Новосибирск, 1989. – 60с. – (Препр./АН СССР. Сиб. отд-ние. ВЦ; № 822).
4. Virbitskaite I.B. Investigating Dataflow Networks with Token Colouring // Methods of Theoretical and Experimental Computer Science. – Novosibirsk, 1989. – P. 142–157.
5. Virbitskaite I.B. Behavioral Notions for Eager Data Flow Computing with I-structures // Current Topics in Informatics Systems Research. – Novosibirsk, 1991. – P. 81–92.
6. Virbitskaite I. The Relative Strength of Density and Crossing Properties for Event Structures // Proc. Intern. Conf. Young Computer Scientists. – Beijing, China, 1993. – P. 547–552.
7. Virbitskaite I. Observing Some Properties of Event Structures // Lect. Notes Comp. Sci. – 1993. – Vol. 735. – P. 229–250.
8. Virbitskaite I. Some Characteristics of Nondeterministic Processes // Parallel Processing Letters. – 1993. – Vol. 3, N 1. – P. 99–106.
9. Вальковский В.А., Вирбицкайте И.Б. Потоковые вычислительные системы // Системная информатика. – Новосибирск: Наука, 1993. – Т. 2. – С. 38–72.
10. Вирбицкайте И.Б., Вотинцева А.В., Шкляев Д.А. О семантических аспектах потоковых вычислений с цветными фишками // Программирование. – 1996. – № 3. – С. 17–35.
11. Вирбицкайте И.Б. О некоторых свойствах структур событий // Кибернетика и системный анализ. – 1997. – № 3. – С. 45–55.
12. Virbitskaite I., Votintseva A. Behavioural Characterizations of Partial Order Logics // Lect. Notes Comput. Sci. – 1997. – Vol. 1279. – P. 463–474.
13. Virbitskaite I. On the Semantics of Concurrency and Nondeterminism: Bisimulations and Temporal Logics // Electronic Notes in Theoretical Computer Science. – 1998. – Vol. 18.
14. Virbitskaite I. An Event Structure Model for Dataflow Computing // Computers and Artificial Intelligence. – 1999. – Vol. 18, N 1. – P. 73–93.
15. Вирбицкайте И.Б., Покозий Е.А. Использование техники частичных порядков для верификации временных сетей Петри // Программирование. – 1999. – № 1. – С. 28–41.

16. Вирбицкайте И.Б., Покозий Е.А. Метод параметрической верификации поведения временных сетей Петри // Программирование. – 1999. – № 4. – С. 16–29.
17. Virbitskaite I.B., Bystrov A.V. Implementing Model Checking and Equivalence Checking for Time Petri Tets by the RT-MEC Tool // Lect. Notes Comp. Sci. – 1999. – Vol. 1662. – P. 194–200.
18. Virbitskaite I.B., Pokozy E.A. Parametric Behaviour Analysis for Time Petri Nets // Lect. Notes Comp. Sci. – 1999. – Vol. 1662. – P 134–140.
19. Virbitskaite I.B., Pokozy E.A. A Partial Order Method for the Verification of Time Petri Nets // Lect. Notes Comp. Sci. – 1999. – Vol. 1684. – P. 547–558.
20. Вирбицкайте И.Б. Семантические модели в теории параллелизма. – Новосибирск: ИСИ СО РАН, 2000. – 196 с.
21. Вирбицкайте И.Б., Боженкова Е.Н. Исследование тестовой эквивалентности временных структур событий // Программирование. – 2000. – № 5. – С. 18–30.
22. Virbitskaite I.B. Towards Decision of Testing Equivalence for Time Petri Nets. – Oxford University Press. Logic Journal of IGPL. – 2000. – Vol. 8, N 6. (Proc. 7th Intern. Workshop on Logic, Lang., Inform. and Comput., 2000, Natal, Brazil). – P. 181–190.
23. Andreeva M.V., Bozhenkova E.N., Virbitskaite I.B. Analysis of Timed Concurrent Models Based on Testing Equivalence // Fundamenta Informaticae. – 2000. – Vol. 43, N 1–4. – P. 1–20.
24. I.B. Virbitskaite. A Verification Method for Timed Concurrent Systems Using Timing Zones // Proc. Internat. Conf. on Concurrency, Specification and Programming. – Berlin: Humboldt University, 2000. – P. 323–334.
25. I.B. Virbitskaite. Characterizing Time Net Processes Categorically // Lect. Notes Comp. Sci. – 2001. – Vol. 2127. – P. 128–141.
26. Moskaleva N.S., I.B. Virbitskaite. On the Category of Event Structures with Dense Time // Lect. Notes Comp. Sci. – 2001. – Vol. 2138. – P. 287–298.
27. I.B. Virbitskaite. An Observation Semantics for Timed Event Structures // Lect. Notes Comp. Sci. – 2001. – Vol. 2244. – P. 215–225.

Подписано в печать 14.11.2001г.
Формат бумаги 60×84 1/16 Объем 2 уч.-изд.л.
Тираж 100 экз.

НФ ООО ИПО "Эмари" РИЦ, 630090, г. Новосибирск, пр. Акад. Лаврентьева, 6