Российская академия наук
Сибирское отделение
Институт систем информатики
им. А.П.Ершова

На правах рукописи

Гаранина Наталья Олеговна

ВЕРИФИКАЦИЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ АФФИННОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ, ЛОГИК ЗНАНИЙ И ДЕЙСТВИЙ

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

Автореферат
диссертации на соискание ученой степени
кандидата физико-математических наук

Новосибирск, 2004

Работа выполнена в Институте систем информатики
Сибирского отделения Российской академии наук

Научный руководитель:	кандидат физико-математических наук Шилов Н.В.
Официальные оппоненты:	доктор физико-математических наук Пальчунов Д.Е. кандидат физико-математических наук Соколов В.А.
Ведущая организация:	Тверской государственный университет (г. Тверь)

Защита состоится 28 декабря 2004 года в 15 час. 00 мин. на заседании диссертационного совета К003.032.01 в Институте систем информатики им А. П. Ершова Сибирского отделения РАН по адресу:

630090, г.Новосибирск, пр. Лаврентьева, 6.

С диссертацией можно ознакомиться в читальном зале библиотеки ИСИ СО РАН (пр. Лаврентьева, 6).

Автореферат разослан "_____" ноября 2004 г.

Ученый секретарь
диссертационного совета
К003.032.01
к.ф.-м.н. Мурзин Ф.А.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность. На сегодняшний день все более явным становится разрыв между техническими возможностями и технологиями обработки и передачи информации. Главным образом, это происходит из-за отсутствия удовлетворительного для промышленности решения проблемы проверки правильности программных систем и логических схем вычислительных устройств. Методы имитационного моделирования и тестирования не обеспечивают исчерпывающего анализа всех возможных вариантов поведения систем, что особенно важно для критических приложений, таких как системы управления воздушным и дорожным движением, медицинская аппаратура, электронная коммерция, сети телефонных коммутаторов. Методы же *формальной верификации* дают полную характеризацию правильности работы программной системы.

Техника формальной верификации, получившая название *проверки на модели* (англ. model checking), является одним из наиболее перспективных и широко используемых подходов к решению проблемы автоматизации отладки и проверки правильности программ. Основными преимуществами этого метода являются его полная автоматизируемость и возможность с помощью него конструктивно исследовать нежелательные поведения системы, что особенно важно в процессе разработки сложных программных приложений. Ключевыми понятиями проверки моделей являются: формальный язык описания свойств программной системы (как правило, это программные логики — темпоральные или динамические), алгоритм проверки выполнения свойств в модели системы и структуры данных, кодирующих модель в процессе проверки.

Разнообразие программных логик, используемых для спецификации систем, объясняется не только тем, что разные логики описывают принципиально разные свойства моделей, но и различной сложностью проверки на модели каждой из них. Поэтому так важно исследовать теоретически алгоритмические свойства различных логик. Традиционно программные логики включают *динамические* логики, *временные логики* и их варианты с *неподвижными точками*. Однако во многих программных системах, в силу связи между информацией и действием, естественно возникает понятие знания. Что робот должен *знать*, чтобы открыть сейф, и откуда он *знает*, достаточно ли он *знает*, чтобы открыть его? Перед пересылкой очередного сообщения, *знает* ли отправитель, что получатель получил предыдущее? Когда база данных может ответить

на вопрос "Я *не знаю*"? Поэтому часто бывает, что правильность функционирования программной системы зависит от "знаний" ее компонент. Поэтому не так давно в семейство программных логик были добавлены *эпистемические логики*, в частности, пропозициональная логика знаний для n агентов PLK_n и пропозициональная логика общих знаний для n агентов PLC_n. Они позволяют специфицировать такие системы, в которых необходимо проверять утверждения, касающиеся знаний параллельных процессов, которых принято называть *агентами*. С помощью логик знаний особенно удобно описывать свойства систем, в которых действия распределенных параллельных процессов зависят от информации, которой они располагают. К таким программным системам относятся, коммуникационные протоколы, особенно в ненадежной среде, программы управления роботами, получающими информацию от окружающей среды и т.п. Проверка на модели систем, специфицированных эпистемическими логиками, позволяет исследовать в этих системах знания, основанные на неполной информации. Иными словами, можно, меняя доступность той или иной информации, проверять, как меняется у агента представление о мире и о других агентах. Например: достаточно ли процессу в системе с разделяемыми ресурсами наблюдать параметр занятости ресурса, чтобы знать, когда он освободится, или необходимо иметь доступ к локальной информации остальных процессов (например, к информации о состоянии вычислений). Однако особенно полезными оказываются *комбинации логик знаний* с темпоральными логиками или динамическими логиками действий с неподвижными точками, поскольку они позволяют описывать эволюцию знаний агентов во времени или их изменение в результате каких-либо действий. При рассмотрении временных аспектов знаний возникают системы, различающиеся "разумностью" агентов, действующих в системах. Например, часто рассматриваются *забывающие* агенты, которые не помнят историю развития событий в системе, а имеют лишь информацию о ее текущем состоянии. В противоположность им можно определить агентов с *абсолютной памятью*, различающих состояния системы, основываясь на запомненных ими истории данных. Кроме того, могут быть *синхронные* агенты, помнящие, "который час" и *асинхронные*, не знающие времени. Агенты, чьи знания не зависят от времени, называются *обычными* агентами. Свойства различных комбинированных логик в системах с разнообразными агентами изучаются в течении последних двадцати лет. Например, в 1986 г. Дж.Хальперном и М.Варди изучена задача разрешимости для

комбинаций временных логик LTL и CTL с логиками PLK_n и PLC_n в (а)синхронных системах как забывающих, так и с абсолютной памятью. В 1998 г. Р.ван дер Мейденом исследована задача проверки на модели формул PLC_n в (а)синхронных системах с абсолютной памятью. В 1999 г. Р.ван дер Мейденом и Н.В.Шиловым была изучена задача проверки на модели для комбинаций PLK_n и PLC_n с LTL в синхронных системах с абсолютной памятью. В этой работе были предложены древовидные структуры данных для проверки на модели логики линейного времени и знаний с ограниченной глубиной знаний. Однако перечисленные работы исследуют комбинации только темпоральных логик с логиками знаний. Комбинации же с динамической логикой с неподвижными точками позволяют выразить более широкий спектр свойств мультиагентных систем. Например, при проверке на модели свойства существования выигрышной стратегии, можно узнать, какой минимальной информацией должен располагать агент, чтобы выиграть в конечной игре. Что касается практической стороны, то в 2003 году в Австралии под руководством Р.ван дер Мейдена был разработан прототип системы проверки на моделях MCK, проверяющий модели, специфицированные формулами некоторых комбинаций пропозициональных логик знаний и времени. Его недостатком является то, что используются опять-таки только темпоральные комбинации логик знаний и почти не реализована проверка самых интересных систем, а именно — синхронных с агентами с абсолютной памятью.

Вторая составляющая проверки на модели — алгоритмы проверки выполнимости спецификации — представлена большим разнообразием методов. Особенно важны алгоритмы проверки на модели формул μ-исчисления, как наиболее выразительной из пропозициональных программных логик. Однако даже самые эффективные из них экспоненциальны относительно размера формулы, поэтому целесообразно рассматривать фрагменты μ-исчисления, имеющие полиномиальную сложность проверки на модели. В 1993 г. А. Эмерсон, С. Джатла и П. Систла впервые определили фрагмент μ-исчисления, который допускает полиномиальную проверку на конечных моделях.

Основной недостаток метода проверки на модели — это "комбинаторный взрыв" в пространстве состояний, который возникает, когда система состоит из компонентов, переходы в которых выполняются параллельно. В 1987 г. К.МакМиллан показал, что, используя символьное представление графа переходов как двоичных разрешающих диа-

грамм (OBDD) Бриана — структур данных для представления булевых функций — можно верифицировать очень сложные системы. Применяя алгоритм Кларка-Эмерсона проверки на модели формул CTL и новое представление графов переходов, можно провести верификацию некоторых систем, содержащих более 10^{20} состояний. Однако неявное представление в виде OBDD вполне подходит для моделирования последовательных схем и протоколов, состояния которых кодируются булевскими переменными, но для систем с целочисленными состояниями такое представление оказывается не совсем естественным. Для таких систем более эффективными оказываются представления, существенно использующие при кодировании тот факт, что элементами пространства состояний являются целые числа. В 1994 г. Б. Бугло и П. Волпер предложили *периодические множества* для представления множеств состояний. Основным недостатком такого представления является то, что оно не допускает полностью символической проверки на моделях. В 1999 г. Т. Бултан, Р. Гербер и В. Пух разработали систему для символической проверки бесконечных моделей, представляя множества состояний *формулами арифметики Пресбургера*. К достоинствам такого способа представления данных относится то, что можно проверять сколь угодно большие и даже бесконечные модели, но недостатком является тот факт, что сложность оперирования формулами Пресбургера вычислительно дорога: она экспоненциально зависит от размера формул.

Из вышесказанного следует, что теоретическое исследование алгоритмических свойств новых комбинаций логик знаний и действий, конструктивное определение новых фрагментов формул μ-исчисления, допускающих полиномиально сложную проверку на модели, выявление новых представлений данных для кодировки пространства состояний и переходов в распределенных системах и реализация метода проверки моделей сложных мультиагентных систем, специфицированных комбинациями логик знаний и действий, является актуальной задачей.

Цель диссертации состоит в разработке эффективных методов верификации моделей мультиагентных систем. Достижение цели связано с решением следующих задач:

— теоретическое исследование проблемы проверки моделей для новых, более выразительных, комбинаций логик знаний и действий;

— теоретическое исследование нового эффективного аппроксимационного алгоритма проверки на модели формул μ-исчисления;

— теоретическое исследование новых, более эффективных, символи-

ческих форматов представлений данных для логик действий и знаний;

— экспериментальная реализация новой системы проверки моделей для комбинированных логик знаний и действий на моделях.

Методы исследований в теоретической части используют аппарат полимодальных логик, логик знаний, времени и действий, разрешимые теории высших порядков, аппарат линейных диофантовых уравнений.

Научная новизна состоит в следующем.

• Определены новые комбинации пропозициональных логик знаний и действий. Для них исследованы их относительная выразительная сила, сложности задачи проверки на модели и разрешимости в мультиагентных системах общего вида и с асинхронными забывающими агентами. В синхронных системах с агентами с абсолютной памятью исследована задача проверки на модели.

• Предложен полиномиальный алгоритм проверки на модели для фрагмента μ-исчисления, вычисляющий верхнюю и нижнюю аппроксимацию семантики формулы.

• Для кодировки множеств состояний и переходов проверяемой модели предлагается использовать новые форматы представления данных, а именно: аффинные множества, векторно-аффинные множества и, для кодировки моделей, чьими состояниями являются деревья — векторно-аффинные деревья.

• По результатам теоретических исследований был выполнен машинный эксперимент: реализована программа Экзаменатор — прототип системы проверки моделей 1) с асинхронными забывающими агентами, специфицированных формулами μ-исчисления в комбинации с логикой знаний PLK_n; 2) с синхронными агентами с абсолютной памятью, специфицированных формулами логики CTL, расширенной действиями в комбинации с логикой знаний PLK_n.

Практическая ценность заключается в том, что в разработанном методе проверки моделей, специфицированных формулами комбинированных логик знаний и действий, имеются следующие достоинства: 1) за счет новых комбинаций логик можно описывать и проверять более богатый спектр свойств мультиагентных моделей; 2) предложенный полиномиальный аппроксимационный алгоритм позволяет эффективно проверить иные, чем раньше, формулы μ-исчисления, а значит и комбинированных логик знаний и действий; 3) предложенная кодировка состояний и переходов моделей позволяет эффективно проверять большие распределенные системы специального широкого класса.

Апробация работы. Основные научные результаты подробно обсуждались на объединенном семинаре ИСИ СО РАН и кафедры программирования НГУ "Теоретическое и экспериментальное программирование", докладывались на следующих научных конференциях и совещаниях:

1. *International Workshop on Fixed Points in Computer Science (FICS 2002), Copenhagen, Denmark, 2002;*
2. *4th International Conference of Computer-aided Technologies in Applied Mathematics, (ICAM-2002), Tomsk, Russia, 2002;*
3. *5th International Conference of Perspectives of System Informatics (PSI 2003), Novosibirsk, Russia, 2003;*
4. *Конференция-конкурс Технологии Microsoft в Информатике и Программировании, Новосибирск, Россия, 2004;*
5. *International Workshop on Concurrency, Specification and Programming (CSP 2004), Caputh, Germany, 2004.*

Публикации. По теме диссертации опубликовано 7 научных работ.

Структура работы. Диссертация состоит из введения, пяти глав, заключения, списка литературы из 53 наименований и приложения. Объем основной части работы — 139 страниц, приложения — 26 страниц. Работа включает 15 иллюстраций и 10 таблиц.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обсуждается актуальность работы, дан краткий обзор основных результатов и открытых проблем исследуемой области, сформулированы цели диссертации, описаны полученные результаты, отмечается их новизна. Введение также содержит краткий обзор структуры диссертации.

В **первой главе** даются основные определения, понятия, результаты и примеры, используемые в диссертации. В **разд. 1.1** определены синтаксис и семантика базовых пропозициональных логик, как частных случаев и расширений пропозициональной полимодальной логики наиболее общего вида, формулы которой строятся из символов пропозициональных констант и символов отношений с помощью стандартных булевских связок и модальностей возможности и необходимости. Семантика данных логик определена в терминах выполнимости на моделях Крипке. *Модель Крипке* включает множество состояний, интерпретации символов отношений и символов пропозициональных констант и может рассматриваться как помеченный ориентированный граф с вершинами и ребрами, помеченными множествами пропози-

циональных констант и символами отношений, соответственно. В этой главе определяются следующие логики. В пропозициональной *логике знаний* для n агентов PLK_n символы отношений являются символами агентов, занумерованных от 1 до n, причем интерпретация этих отношений *неразличимости* должна быть отношением эквивалентности. В пропозициональной *логике общих знаний* для n агентов PLC_n символы отношений являются группами символов агентов, т.е. подмножествами множества $\{1, \ldots, n\}$, а интерпретация этих отношений является рефлексивно-транзитивным замыканием отношений неразличимости агентов из группы, и, значит, тоже эквивалентностью. В элементарной пропозициональной *динамической логике EPDL* символы отношений — это символы действий с интерпретацией без каких-либо ограничений. Во *временной логике на деревьях с действиями Act*-CTL используются другие модальности: *в следующем состоянии*, *всегда*, *когда-нибудь* и *пока*. Так как это ветвящаяся логика, к синтаксису модальностей добавляются кванторы *на каждом пути* и *на некотором пути*, где под *путем* понимается последовательность состояний, получающаяся в результате повторений некоторого действия. *Act*-CTL с единственным действием равна известной логике CTL. *Пропозициональное μ-исчисление* μC — это EPDL, синтаксически расширенная конструкциями неподвижных точек формул, определяющих монотонное преобразование. Показано, что: PLC_n — расширение PLK_n; *Act*-CTL является расширением EPDL; μC — расширение *Act*-CTL. Определено понятие *абстракции* для полимодальных логик. Оно позволяет формализовать тот факт, что выполнимость некоторого множества формул (не обязательно тавтологий) сохраняется при переходе от конкретной модели к абстрактной. Оказывается полезным ввести новое понятие *аффинной модели* программной системы как конечной модели, состояния которой определяются целочисленными переменными. Кроме того, в описании модели не допускается перемножение этих переменных (умножение на константу допустимо), а все равенства, неравенства и переходы зависят от одной переменной.

В **разд. 1.2** описаны примеры, иллюстрирующие выразительные возможности комбинации логик знаний и действий. В **разд. 1.2.1** — задача о монетках, состоящая в определении фальшивой монеты среди настоящих посредством ограниченного числа взвешиваний на чашечных весах. Для этой головоломки построена ее формальная модель с единственным агентом, который “помнит” все свои действия и их результаты, т.е. обладает *абсолютной памятью*. Он вычисляет фальши-

вую монету, “анализируя” последовательность взвешиваний, и формула комбинированной логики знаний и действий PLK_1+PDL, выражает факт, что агент через заданное число взвешиваний будет точно знать, какая из монет фальшива. В **разд. 1.2.2** приведена похожая задача об угадывании числа, служащая для иллюстрации сложных формальные понятий синхронных мультиагентных систем с абсолютной памятью и примененная в качестве теста в экспериментальном исследовании. Сама задача состоит в угадывании натурального числа из определенного промежутка посредством выполнения ограниченного количества арифметических действий. Для этой головоломки также определена модель с единственным агентом с абсолютной памятью, вычисляющим задуманное число, и формула комбинированной логики знаний и действий PLK_1+PDL, выражающая факт, что агент через заданное количество арифметических действий будет точно знать, какое число задумано.

Во **второй главе** рассматриваются алгоритмические проблемы комбинированных логик действий и знаний в различных мультиагентных системах. Для мультиагентных систем общего вида исследуется: 1) сравнительная выразительная сила этих логик; 2) задача проверки на модели; 3) разрешимость. Для мультиагентных систем с асинхронными забывающими агентами исследуются те же задачи. Для мультиагентных систем с синхронными агентами с абсолютной памятью исследуется задача проверки на модели. Более подробно: в **разд. 2.1** на основе базовых логик определяются исследуемые *комбинированные логики* $\mu\mathrm{PLC}_n$, $\mu\mathrm{PLK}_n$, *Act*-CTL-C_n, *Act*-CTL-K_n, EPDL-C_n и EPDL-K_n как синтаксическая комбинация соответствующих динамических логик и логик знаний с прежней семантикой, определенной в мультиагентной *среде*, являющейся сочетанием моделей для логик знаний и динамических логик. В **утверждении 4** доказано, что сравнительная выразительная сила логик удовлетворяет следующим неравенствам: EPDL-$\mathrm{K}_n \leq$ EPDL-C_n, *Act*-CTL-$\mathrm{K}_n \leq$ *Act*-CTL-C_n, $\mu\mathrm{PLK}_n = \mu\mathrm{PLC}_n$, EPDL-$\mathrm{K}_n <$ *Act*-CTL-$\mathrm{K}_n < \mu\mathrm{PLK}_n$ и EPDL-$\mathrm{C}_n <$ *Act*-CTL-$\mathrm{C}_n < \mu\mathrm{PLC}_n$.

Алгоритмические проблемы для этих логик в средах, т.е. системах с агентами общего вида, исследуются в **разд. 2.2**. В **утверждении 5** оценивается сложность задачи проверки на модели: относительно размера модели и формулы она 1) линейна для формул логик EPDL-K_n, EPDL-C_n, *Act*-CTL-K_n, и *Act*-CTL-C_n и 2) экспоненциальна для формул логик $\mu\mathrm{PLK}_n$ и $\mu\mathrm{PLC}_n$. **Утверждение 6** о проблеме разрешимости: она $PSPACE$-полна для EPDL-K_n и $EXPTIME$-полна для EPDL-C_n, *Act*-

CTL-K$_n$, *Act*-CTL-C$_n$, μPLK$_n$, и μPLC$_n$. **Теорема 1** об алгоритмических свойствах изучаемых логик формально объединяет эти утверждения. **Утверждение 7** конструктивно доказывает возможность использования методов проверки на модели формул μ-исчисления для проверки формул комбинированных логик.

В **разд. 2.3** *асинхронные системы с забывающими агентами* определяются как модели, порождаемые мультиагентными средами, чьими состояниями являются все последовательности состояний порождающей среды, а агенты таких моделей "видят" только последнее состояние последовательностей, "забывая" предыдущие состояния. Заметим, что количество состояний в этих системах бесконечно. Асинхронные системы с забывающими агентами в определенном смысле сводятся к мультиагентным средам. Чтобы показать это, определяется отображение непустых конечных последовательностей состояний в последний элемент этой последовательности и обратное ему отношение. В **утверждении 8** показано, что формулы комбинированных логик выполняются на непустой конечной последовательности состояний в забывающей асинхронной среде тогда и только тогда, когда они выполняются в последнем состоянии последовательности в исходной среде. Это влечет **утверждение 9** о том, что асинхронные системы с забывающими агентами являются абстракцией мультиагентных сред относительно формул исследуемых комбинированных логик, т.е. все формулы выполнимые в некотором состоянии порождаемой асинхронной системы с забывающими агентами, выполняются в соответствующем состоянии порождающей системы. Отсюда следует **теорема 2** об эквивалентности алгоритмических свойств комбинированных логик в асинхронных средах с забывающими агентами соответствующим свойствам в мультиагентных средах. Следовательно, интересующая нас задача проверки на моделях формул этих логик в бесконечных забывающих асинхронных средах сводится к той же задаче в конечных мультиагентных средах.

В **разд. 2.4** определяются *синхронные системы с абсолютной памятью* как модели, порождаемые мультиагентными средами, чьими состояниями являются последовательности пар (состояние, действие) порождающей системы и агенты таких моделей "видят" всю последовательность целиком. Эти системы также имеют бесконечное множество состояний. Приведен пример такой модели на основе задачи об угадывании числа. Далее показаны симуляционные возможности синхронных систем с абсолютной памятью: **утверждение 10** доказывает,

что можно симулировать вычисления машин Тьюринга; **утверждение 11** показывает возможность симуляции выполнимости формул слабой логики второго порядка с единственной функцией следования $WS(1)S$. Эти симуляционные свойства синхронных систем с абсолютной памятью используются в исследовании задачи проверки комбинированных логик в таких системах.

Разд. 2.5 посвящен проверке на модели комбинированных логик в синхронных системах с абсолютной памятью. В **утверждении 12** доказано, что задача проверки на модели при всех $n > 1$: для EPDL-C$_n$ — $PSPACE$-полна; для *Act*-CTL-K$_n$ — разрешима с неэлементарной верхней и нижней границей; для *Act*-CTL-C$_n$, μPLK$_n$, и μPLC$_n$ — неразрешима.

В **разд. 2.6** проводится более детальное исследование задачи проверки на модели для формул логики *Act*-CTL-K$_n$ с ограниченной глубиной знаний. Целями этого исследования являются определение более точных границ задачи проверки на модели для данной логики, а также конструктивный подход к этому методу проверки. В связи с этим определяется *глубина знаний формулы* k как максимальная вложенность модальностей знаний. Новая модификация древовидных структур данных — *k-деревьев знаний* — порождаемых синхронными системами с абсолютной памятью, иллюстрируется на примере задачи об угадывании числа. В **утверждении 13** подсчитывается количество k-деревьев над системой и максимальное количество вершин в таких деревьях — они сравнимы с башней экспонент высоты k. Определяются *деревья знаний агентов на последовательности состояний* как деревья знаний, зависящие от всех состояний этой последовательности и переходов между ними. Эти деревья отражают знания, приобретенные агентами к последнему состоянию последовательности. Для k-деревьев определяется *функция обновления знаний* в результате действий. В **утверждении 14** обосновывается корректность функции обновления знаний, т.е. показано, что деревья знаний агентов на последовательности состояний могут быть получены последовательным применением функция обновления знаний к дереву знаний, соответствующему исходному элементу данной последовательности. В **утверждении 15** показано, что логики *Act*-CTL-K$_n$ и *Act*-CTL имеют одинаковую выразительную силу в синхронных системах с абсолютной памятью. С помощью функций обновления знаний определяется класс *ассоциированных* с синхронными системами с абсолютной памятью конечных моделей Крипке, базирую-

щихся на k-деревьях. Из **утверждения 17** следует возможность проверки формул *Act*-CTL-K$_n$ с семантикой в бесконечных синхронных системах с абсолютной памятью на ассоциированных с ними конечных моделях. **Утверждение 18**, вследствие учета размера и количества k-деревьев в ассоциированных моделях Крипке, более точно оценивает сложность задачи проверки на моделях для формул *Act*-CTL-K$_n$ и вместе с утверждением 12 приводит к **теореме 3** об оценке сложности задачи проверки на модели комбинированных логик в синхронных системах с абсолютной памятью.

Одним из главных выводов второй главы является то, что проверка мультиагентных систем, специфицированных формулами комбинированных логик знаний и действий, так или иначе, сводится к проверке формул μ-исчисления в конечных моделях. Поэтому в **третьей главе** предлагается аппроксимационный алгоритм для вычисления семантики формул нового фрагмента μ-исчисления, имеющий полиномиальную оценку сложности, и обосновывается его корректность. Сначала в **разд. 3.1** обсуждаются параметры сложности задачи проверки на модели формул μ-исчисления. Затем в **разд. 3.2** вводится специальная префиксная форма формулы μ-исчисления, приемлемая для алгоритма проверки, и в **утверждениях 19 и 20** показано, что всякую формулу μ-исчисления можно привести к такому виду. Далее в **разд. 3.3** — собственно алгоритм, полиномиальность которого основана на полиномиальной оценке вычислений независимых неподвижных точек, как показано в **утверждении 21** о корректности алгоритма. Там же утверждается, что этот алгоритм вычисляет верхнюю и нижнюю аппроксимации семантики формулы μ-исчисления. Следствием утверждения 21 является **теорема 4** о том, что совпадающие аппроксимации равны точной семантике формулы. Показано, что формула, выражающая справедливость, принадлежит новому фрагменту, но не фрагменту Эмерсона-Джатлы-Систлы, также имеющего полиномиальную сложность проверки на модели. Доказано важное **утверждение 22** о том, что данный алгоритм вычисляет точную семантику формул *Act*-CTL.

В **четвертой главе** предлагаются новые символические представления данных для алгоритмов символической проверки на моделях формул μ-исчисления и комбинированной логики действий и знаний *Act*-CTL-K$_n$ — *аффинные представления*. Также описаны методы манипуляции с ними, необходимые для символической проверки на моделях, а именно: объединение множеств, их пересечение и вычисление предусло-

вий действий, а так же проверка множеств на включение.

В **разд. 4.1** изложены синтаксис и семантика языка описания моделей, для которых возможны аффинные представления, а именно — аффинных моделей. В **подразд. 4.1.1** определяется синтаксис, а в **подразд. 4.1.2** — семантика языка и ассоциированная с ней модель Крипке, чья корректность относительно описания модели показана в **утверждении 24**. В **подразд. 4.1.3** описан переход от аффинных моделей с нечисловыми типами переменных к аффинным моделям с целочисленными переменными. В **подразд. 4.1.4** приведен простой пример описания аффинной модели игры в числа.

Разд. 4.2 посвящен описанию *ограниченных аффинных множеств*, которые являются множествами, определяемыми конечным набором *аффинных атомов* — линейных двучленов с целыми коэффициентами, каждый из которых определен на отрезке целых чисел. С их помощью можно кодировать аффинные модели с единственной переменной. В **подразд. 4.2.1** предлагается метод кодирования с помощью аффинных множеств пропозициональных констант и действий таких моделей и в **утверждении 25** доказана его корректность. В **подразд. 4.2.2** описаны алгоритмы манипуляции с аффинными множествами. Объединение аффинных множеств является простым объединением множеств. Для вычисления пересечения необходимо определить совпадающие значения аффинных множеств посредством решения линейных диофантовых уравнений. Вычисление предусловий детерминированных переходов сводится к вычислению значений аффинных множеств, из которых возможен детерминированный переход в данное аффинное множество, для чего также используются решения линейных диофантовых уравнений. Предусловия недетерминированных действий вычисляются относительно их контекста в проверяемой формуле с использованием предусловий детерминированных переходов, из которых они состоят. Корректность этих алгоритмов показана в **утверждении 26**, там же оценена их сложность. В **подразд. 4.2.3** изложены идеи оптимизации ограниченных аффинных множеств, необходимой, поскольку объединение множеств может привести к дублированию элементов. **Подразд. 4.2.4** содержит алгоритм проверки включения множеств и **утверждение 27** о его корректности и сложности.

В **разд. 4.3** представлены *векторно-аффинные множества*, обобщающие понятие аффинных множеств для возможности кодирования аффинных моделей с несколькими переменными. Здесь идея состоит в

том, чтобы каждое множество состояний модели описать набором *аффинных векторов*, компонентами которых являются аффинные атомы, соответствующие значениям переменных модели. Аналогично предыдущему разделу в **подразд. 4.3.1** предлагается метод кодирования с помощью векторно-аффинных множеств пропозициональных констант как множеств массивов аффинных атомов, и действий аффинных моделей с помощью *векторных действий* — как множеств массивов аффинных переходов. В **утверждении 28** доказывается корректность этого представления. В **подразд. 4.3.2** описаны алгоритмы манипуляции, необходимые для символической проверки с использованием векторно-аффинного представления. Поскольку векторно-аффинные множества расширяют понятие аффинного множества, а векторное действие — понятие аффинного действия, то эти алгоритмы основаны на аналогичных алгоритмах для аффинных множеств. В **утверждении 29** показана корректность алгоритмов и подсчитана их сложность. В **подразд. 4.3.3** изложен простой алгоритм оптимизации векторно-аффинных множеств с целью исключения дублирующих друг друга элементов представления, а **подразд. 4.3.4** содержит алгоритм проверки включения множеств и **утверждение 30** о его корректности и сложности.

В **разд. 4.4** на простом примере показана эффективность нового представления для моделей определенного класса по сравнению с популярным символическим представлением данных как упорядоченных бинарных разрешающих диаграмм (OBDD). Вычислены векторно-аффинная кодировка и BDD-кодировка системы переходов с 20 состояниями. В частности, векторно-аффинная кодировка переходов — это целочисленная таблица размера 2×4, а дерево BDD-кодировки переходов содержит 72 вершины и 144 ребра. Процесс трансляции модели в векторно-аффинную кодировку происходит существенно быстрее.

Разд. 4.5 посвящен аффинной проверке на моделях комбинированных логик. В **подразд. 4.5.1** синтаксис языка описания аффинных моделей дополнен конструкциями, позволяющими вводить агентов, связывая с ними множества наблюдаемых переменных модели. Здесь же определена семантика этих дополнений и среда, ассоциированная с описанием мультиагентной аффинной модели, чья корректность относительно описания среды показана в **утверждении 31**. Отметим, что из утверждения 7 следует, что свойства мультиагентных сред с обычными и забывающими (в силу теоремы 2) агентами, специфицированными в логике $\mu\mathrm{PLC}_N$ (и менее выразительных логиках), можно проверять,

используя алгоритмы проверки на модели для μ-исчисления. Поэтому в **подразд. 4.5.2** определяется *векторно-аффинный переход по знаниям*, представляющий отношения неразличимости обычных и забывающих агентов, с использованием понятия наблюдаемых переменных, а именно: переход по знаниям некоторого агента из состояния w возможен в те состояния, где значения наблюдаемых переменных совпадают с их значениями в w. В **утверждении 32** показана корректность этого представления. В **подразд. 4.5.3** рассмотрены новые представления данных для проверки формул *Act*-CTL-K$_n$ в синхронных системах с абсолютной памятью — *векторно-аффинные деревья*. Они являются обобщением векторно-аффинных множеств для кодировки моделей, чьими состояниями являются деревья знаний. Эти структуры — деревья с вершинами, помеченными аффинными векторами, представляющими множество состояний, и ребрами, помеченными агентами. В **подразд. 4.5.4** предлагается метод кодирования с помощью векторно-аффинных деревьев пропозициональных констант и действий моделей на деревьях, порожденных аффинными моделями с синхронными агентами с абсолютной памятью, и в **утверждении 33** доказывается его корректность. Векторно-аффинные деревья позволяют кодировать множества деревьев с мощностью порядка размера модели, при этом экспоненциально уменьшается количество вершин по сравнению с обычными деревьями. В **подразд. 4.5.5** описаны алгоритмы манипуляции, необходимые для символической проверки с использованием векторно-аффинного представления деревьев знаний. Поскольку векторно-аффинные деревья являются обобщением векторно-аффинных множеств, то эти алгоритмы основаны на аналогичных алгоритмах для векторно-аффинных множеств. В **утверждении 34** показана корректность алгоритмов и вычислена их сложность. В **подразд. 4.5.5** изложен простой алгоритм оптимизации аффинных деревьев с целью исключения дублирующих друг друга элементов представления, а **подразд. 4.5.6** содержит алгоритм проверки включения множеств аффинных деревьев и **утверждение 35** о его корректности и сложности.

В **пятой главе** содержится краткое описание прототипа системы Экзаменатор для проверки на моделях формул комбинированных логик. В **разд. 5.1** описаны входные данные программы: аффинные модели с единственным агентом, со свойствами, специфицированными формулами логики μC+PLK$_1$, и с указанием типа агента — забывающего или с абсолютной памятью. Также описана структура программы, состо-

ящая из следующих блоков: обработка входного файла, трансляция в аффинные структуры данных, манипуляция данными, алгоритм проверки и вывод. Выходными данными являются: константа $true$ ($false$), означающая, что спецификации удовлетворяют все (ни одно) состояния модели, либо множество состояний в которых формула спецификации выполняется. В **разд. 5.2** описаны интерфейс и параметры программы. В **разд. 5.3** приведен тест — задача об угадывании числа — на языке описания аффинных моделей и результаты тестирования для различных входных параметров задачи.

Основные результаты и выводы. В рамках диссертации были получены следующие результаты.

• В работе с теоретической точки зрения изучены выразительная сила, верхняя граница задачи проверки на модели и сложность разрешимости для пропозициональных программных логик EPDL-K, EPDL-C, *Act*-CTL-K, *Act*-CTL-C, μPLK и μPLC в средах общего вида, асинхронных забывающих средах и синхронных средах с абсолютной памятью. В частности, доказано, что 1) асинхронные среды с забывающими агентами являются абстракцией сред общего вида относительно множества всех формул вышеперечисленных логик; 2) задача проверки на модели разрешима в синхронных средах с абсолютной памятью для *Act*-CTL-K.

• Для формул фрагмента пропозиционального μ-исчисления предложен полиномиальный аппроксимационный алгоритм проверки на модели, вычисляющий верхнюю и нижнюю аппроксимацию семантики формул и доказана его корректность.

• Для кодировки множеств состояний и переходов проверяемой модели предложен новый эффективный формат представления данных, а именно: аффинные множества, векторно-аффинные множества и векторно-аффинные деревья. Последнее делает осуществимой проверку на модели синхронных мультиагентных систем с абсолютной памятью.

• Вышеизложенные результаты объединяются в методе аффинной проверки на модели для логик, представимых в $\mu\mathrm{PLC}_n$, в средах с асинхронными забывающими агентами и логик, представимых в *Act*-CTL-K_n, в средах с синхронными агентами с абсолютной памятью. На основе этого метода реализован прототип системы проверки моделей Экзаменатор и проведен эксперимент, показывающий эффективность метода.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. **Shilov N.V., Garanina N.O.** Combining Knowledge And Fixpoints — Novosibirsk, 2002. — 50 p. — (Prepr./ Sib.Div. of RAS. IIS; N 98).

2. **Shilov N.V., Garanina N.O.** Model Checking Knowledge And Fixpoints // Proc. 4th Int. Workshop on Fixed Points on Computer Science — Copenhagen, Denmark, 2002. — P. 25–39.

3. **Shilov N.V., Garanina N.O.** Model Checking Knowledge And Fixpoints // Тр. конф. Новые Информационные Технологии в Исследовании Сложных Структур — Вестник ТГУ. Материалы научных конференций, симпозиумов, школ, проводимых в ТГУ. — 2002. — N1(II) — С. 20–23.

4. **Shilov N.V., Garanina N.O.** A Polynomial Approximations for Model Checking // Proc. 5rd Int. Conf. Perspectives of System Informatics. — Berlin etc.: Springer-Verlag, 2003. — P. 395—400. — (Lect. Notes Comput. Sci.; 2890).

5. **Гаранина Н.О.** Аффинная проверка моделей программ // Тр. конф. Технологии Microsoft в Информатике и Программировании — Новосибирск, НГУ, 2004. — С. 94–96.

6. **Гаранина Н.О.** Аффинное представление данных для проверки моделей программ — Новосибирск, 2004. — 48 с. — (Препр./ Сиб. отдние. РАН. ИСИ; N 116)

7. **Shilov N.V., Garanina N.O., Kalinina N.A.** Model checking knowledge, actions and fixpoints // Proc. Int. Workshop on Concurrency, Specification and Programming — Caputh, Germany, 2004. — v.2, p.351–357.

Личный вклад автора. Результаты, касающиеся теоретических исследований комбинированных логик знаний и действий, а также алгоритма проверки фрагмента μ-исчисления, получены автором в сотрудничестве с научным руководителем Н.В.Шиловым. Вклад автора состоял в исследовании алгоритмических свойств асинхронных мультиагентных систем, адаптации определений и утверждений, относящихся к k-деревьям, для проверки на модели формул логики знаний и действий в синхронных системах с абсолютной памятью, а также в доказательстве корректности аппроксимационного алгоритма проверки на модели формул μ-исчисления. Теоретическое исследование аффинных представлений данных и реализация прототипа системы проверки мультиагентных систем выполнены автором полностью самостоятельно.

Гаранина Наталья Олеговна

ВЕРИФИКАЦИЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ АФФИННОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ, ЛОГИК ЗНАНИЙ И ДЕЙСТВИЙ

Подписано в печать 19.11.04 Объем 1,1 уч.-изд.л.
Формат бумаги 60×84 1/16 Тираж 100 экз.

ЗАО РИЦ "Прайс-курьер"
630090, г. Новосибирск, пр. Акад. Лаврентьева, 6, тел. (383-2) 34-22-02