САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Герасимов Александр Сергеевич

РАЗРАБОТКА И РЕАЛИЗАЦИЯ АЛГОРИТМА ПОИСКА ВЫВОДА В РАСШИРЕНИИ БЕСКОНЕЧНОЗНАЧНОЙ ПРЕДИКАТНОЙ ЛОГИКИ ЛУКАСЕВИЧА

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

АВТОРЕФЕРАТ

диссертации на соискание ученой степени кандидата физико-математических наук

Санкт-Петербург – 2007

Работа выполнена на кафедре информатики математико-механического факультета Санкт-Петербургского государственного университета.

Научный руководитель:	доктор физико-математических наук, профессор Косовский Николай Кириллович
Официальные оппоненты:	доктор физико-математических наук Оревков Владимир Павлович
	кандидат физико-математических наук Тишков Артем Валерьевич
Ведущая организация:	Санкт-Петербургский государственный политехнический университет

Защита диссертации состоится «26» апреля 2007 года в 15:30 на заседании диссертационного совета Д 212.232.51 по защите диссертаций на соискание ученой степени доктора наук при Санкт-Петербургском государственном университете по адресу: 198504, Санкт-Петербург, Старый Петергоф, Университетский пр., д. 28, математико-механический факультет Санкт-Петербургского государственного университета.

С диссертацией можно ознакомиться в Научной библиотеке Санкт-Петербургского государственного университета по адресу: 199034, Санкт-Петербург, Университетская наб., д. 7/9.

Автореферат разослан «14» марта 2007 года.

Ученый секретарь
диссертационного совета
доктор физико-математических наук Мартыненко Б. К.

Общая характеристика работы

Актуальность темы. Интерес к многозначным (в частности, бесконечнозначным) логикам объясняется их разнообразными применениями, включающими представление нечетких знаний и приблизительные рассуждения (см., например, [12, 14]).

Многозначная логика как область математической логики развивалась параллельно с нечеткой логикой, восходящей к Заде [18, 3]. Нечеткая логика Заде основана на теории нечетких множеств — множеств с размытыми границами; такое множество задается с помощью функции принадлежности, сопоставляющей элементу действительное число из отрезка $[0, 1]$. *Нечеткой логикой в широком смысле* (далее для краткости — нечеткой логикой) сейчас называют (см. [15]) дисциплину, использующую понятия теории нечетких множеств для разработки методов прикладных приблизительных рассуждений. Нечеткая логика используется в промышленных системах нечеткого контроля, например, в бытовых приборах. Однако с формально-логической точки зрения методы нечеткой логики не представляются корректно обоснованными [14].

В последнее десятилетие активно идет процесс формализации нечеткой логики (см. основополагающие труды [14, 12, 7]). В связи с этим *математической нечеткой логикой* (или *нечеткой логикой в узком смысле*) называют дисциплину, которая разрабатывает дедуктивные системы для нечеткой логики, рассматривая ее как многозначную логику, в стиле и со строгостью математической логики. Бесконечнозначная предикатная логика Лукасевича (ее описание может быть найдено, например, в [14]) является одной из основных многозначных логик, используемых для формализации нечеткой логики.

Процесс становления математической нечеткой логики еще далек от завершения в связи с тем, что нечеткая логика включает концепции, которые отсутствуют в многозначной логике. В частности, в нечеткой логике имеются лингвистические модификаторы «очень», «чрезвычайно», «вполне» и др. Заде [3] применяет возведение функции принадлежности в квадрат для представления модификатора «очень», что делает формализацию нечеткой логики трудной. Потому альтернативная и более простая формализация существенной черты нечеткой логики — модификаторов типа «очень» — является важным шагом в сближении математической нечеткой логики с нечеткой логикой в широком смысле; к тому же это расширяет область возможных применений математической нечеткой логики.

Необходимым условием для многих успешных применений какой-либо логики является наличие удобных методов поиска доказательств в этой логике с помощью компьютера. Для бесконечнозначной предикатной логики Лукасевича до этой работы были известны лишь исчисления гильбертовского типа (их можно найти, например, в [14, 12]), не подходящие для автоматического поиска доказательств.

Для бесконечнозначной пропозициональной логики Лукасевича известны разнообразные методы поиска доказательств, в том числе разработанные в течение последних нескольких лет, например: методы семантических таблиц [13, 17], секвенциальные исчисления [5, 10, 16]. Особо отметим секвенциальное исчисление для уровневой логики [5]. Логические связки уровневой логики позволяют записывать формулы логики Лукасевича. Для распознавания аксиом секвенциального исчисления уровневой логики применяются методы линейного программирования, и этот подход развивается в данной работе.

При попытках применения методов поиска доказательств в пропозициональной бесконечнозначной логике Лукасевича для вывода нечетких знаний ощущается ограниченность этих методов, поскольку они, в частности, не охватывают предикатную логику. Например, в [16] при представлении нечетких знаний приходится переводить предикатные формулы в пропозициональные, что, во-первых, можно осуществить только для конечных областей определения предикатов и, во-вторых, значительно удлиняет запись формул, представляющих исходные знания.

Итак, разработка удобных для поиска вывода исчислений для бесконечнозначной предикатной логики Лукасевича и реализация компьютерных программ для поиска вывода является перспективной исследовательской задачей. А возрастающий интерес к бесконечнозначной предикатной логике Лукасевича в связи с развитием математической нечеткой логики обусловливает потребность в автоматизации поиска доказательств для этой логики Лукасевича.

Цели работы. Расширение бесконечнозначной предикатной логики Лукасевича средствами для выражения модификаторов типа «очень», разработка исчисления для такого расширения и реализация алгоритма для автоматического поиска вывода в этом исчислении.

Основные результаты

1. Сформулировано секвенциальное исчисление для бесконечнозначной предикатной логики Лукасевича, расширенной модификаторами типа «очень», которые восходят к нечеткой логике Заде.

2. Исследованы свойства предложенного секвенциального исчисления, служащие теоретической основой для разработки алгоритма поиска вывода в этом исчислении.

3. Разработан алгоритм поиска вывода в предложенном секвенциальном исчислении. Доказаны свойства разработанного алгоритма, отражающие различные аспекты его корректности.

4. Алгоритм поиска вывода реализован в виде интерфейса прикладного программирования.

5. Уточнен алгоритм решения систем линейных двучленных неравенств, используемый для распознавания некоторых аксиом введенного секвенциального исчисления. Получена оценка временной сложности этого алгоритма в формальной вычислительной модели.

6. Алгоритм решения систем линейных двучленных неравенств реализован в виде интерфейса прикладного программирования.

Научная новизна. Все основные результаты диссертации являются новыми. Следует отметить, что до данной работы для бесконечнозначной предикатной логики Лукасевича были известны лишь исчисления гильбертовского типа, и не были разработаны ни теоретические основы, ни программные средства для автоматического доказательства в этой логике.

Теоретическая и практическая ценность. Предложенная логика может быть использована для представления нечетких знаний. Сформулированное секвенциальное исчисление может использоваться для доказательства не только во введенной логике, но и в бесконечнозначной предикатной логике Лукасевича, что значительно эффективнее, чем использование исчислений гильбертовского типа.

Реализованный интерфейс прикладного программирования (ИПП) для поиска вывода может применяться, например, в исследовательских целях для автоматического доказательства как в предложенной логике, так и в логике Лукасевича. Этот ИПП может служить ядром дедуктивной системы, основанной на любой из упомянутых логик.

Реализованный (также в виде ИПП) алгоритм решения систем линейных двучленных неравенств может использоваться для решения задач бóльших размеров, чем позволяют системы компьютерной алгебры, которые предоставляют алгоритмы решения более общих задач.

Апробация работы. Результаты работы были представлены на VIII и IX Общероссийских научных конференциях «Современная логика: проблемы теории, истории и применения в науке» (Санкт-Петербург, 2004 и 2006 годы); Межвузовском конкурсе-конференции студентов, аспирантов и молодых ученых Северо-Запада «Технологии Microsoft в теории и практике программирования» (Санкт-Петербург, 2005 год); Международной конференции «Устойчивость и процессы управления» (Санкт-Петербург, 2005 год); семинаре Санкт-Петербургского отделения Российской ассоциации искусственного интеллекта (Санкт-Петербург, 2006 год); XVI Международной школе-семинаре «Синтез и сложность управляющих систем» (Санкт-Петербург, 2006 год); Десятой национальной конференции по искусственному интеллекту с международным участием КИИ-06 (Обнинск, 2006 год).

Публикации. Основные результаты опубликованы в работах [$1'$ – $6'$].

Структура и объем диссертации. Диссертация состоит из 6 глав, списка литературы и 2 приложений. Диссертация изложена на 194 страницах машинописного текста. Основной текст диссертации занимает 168 страниц, приложения занимают 26 страниц. Список литературы содержит 82 наименования.

Содержание работы

Первая глава содержит аннотацию данной диссертационной работы, описание бесконечнозначной предикатной логики Лукасевича, обзор близких по теме исследования работ и обоснование актуальности направлений исследований, разрабатываемых в данной диссертации. Затем ставятся цели этой диссертации и излагается краткое содержание следующих глав.

Вторая глава посвящена описанию предлагаемой логики (обозначаемой Lq), секвенциального исчисления LqS для нее и свойств этого исчисления [$2'$, $1'$].

В первом разделе этой главы определяется язык логики Lq и его семантика. Неотъемлемой частью каждого предикатного символа является так называемый его *отрезок истинностных значений* $[a, b]$, где a, b — рациональные числа, $a < b$. *Термом* является предметная переменная и предметная константа. *Атомарной формулой* является любое рациональное число, пропозициональная переменная и предикатный символ с заключенным в скобки списком термов. *Формулами* логики Lq являются атомарные формулы, а также $(A\&B)$, $(A \vee B)$, $(A \prec B)$, $q \cdot A$, $\forall xA$, $\exists xA$, где A и B — формулы логики Lq, q — рациональное число, x — предметная переменная. Связки $\prec$ и $q\cdot$ носят названия *нечеткое неравенство* и *модератор* соответственно.

Для задания семантики языка логики Lq вводятся понятия *интерпретации* и *оценки* языка. Эти понятия аналогичны традиционным, за исключением того, что здесь интерпретация сопоставляет предикатному символу предикат, который действует в отрезок действительных чисел, являющийся отрезком истинностных значений этого предикатного символа (каждой пропозициональной переменной сопоставляется действительное число из такого отрезка). Если заданы интерпретация и оценка, то каждая формула A получает *истинностное значение* $[A]$, являющееся действительным числом, согласно следующим правилам: $[(A\&B)] = \min([A],[B])$, $[(A \vee B)] = \max([A],[B])$, $[(A \prec B)] = [B] - [A]$, $[q \cdot A] = q \cdot [A]$, $[\forall x A] = \inf_x [A]$, $[\exists x A] = \sup_x [A]$. Формула называется *общезначимой*, если истинностное значение этой формулы неотрицательно во всякой интерпретации при любой оценке.

Отметим, что, во-первых, модераторы реализуют увеличение или уменьшение истинностных значений формул, поэтому с помощью модераторов можно формализовать различные модификаторы нечеткой логики типа «очень». Во-вторых, утверждение о том, что во всякой интерпретации при любой оценке формула A принимает истинностные значения, не меньшие (соответственно, не большие) рационального числа r, эквивалентно тому, что формула $(r \prec A)$ (соответственно, $(A \prec r)$) общезначима. В-третьих, любую формулу бесконечнозначной предикатной логики Лукасевича можно представить в виде формулы логики Lq так, что во всякой интерпретации при любой оценке истинностные значения этих формул совпадают.

Приводится следующий пример формализации приблизительных рассуждений с помощью логики Lq. Из посылок (1) если предмет мал, то его трудно разглядеть, и (2) предмет z очень мал, следует, что (3) предмет z довольно трудно разглядеть. Введем предикат $P1[0,1](x)$, который сопоставляет предмету x действительное число из отрезка $[0,1]$, характеризующее степень малости этого предмета, и предикат $P2[0,1](y)$, аналогичным образом выражающий степень трудности разглядывания предмета y. Модификатор «очень» формализуем с помощью модератора $1/2\cdot$, «довольно» — с помощью модератора $2/3\cdot$. Указанное приблизительное рассуждение запишем в виде формулы логики Lq: $((\forall x(P1[0,1](x) \prec P2[0,1](x)) \ \& \ 1/2 \cdot P1[0,1](z)) \prec 2/3 \cdot P2[0,1](z))$. Таким образом, для обоснования этого приблизительного рассуждения достаточно доказать соответствующую ему формулу.

В конце первого раздела второй главы доказана теорема о том, что множество общезначимых формул логики Lq неперечислимо.

Во втором разделе второй главы сформулировано безантецедентное секвенциальное исчисление LqS для логики Lq, и определены сопутствующие понятия. *Секвенцией* называется конечный список формул логики Lq (*членов* этой секвенции), разделенных запятыми; некоторые формулы могут повторяться; порядок формул в списке не имеет значения. Секвенция называется *общезначимой*, если дизъюнкция всех ее членов общезначима (пустая секвенция в виде такой дизъюнкции представляется числом -1). Приводятся *правила вывода*, вводящие логические символы, кроме модератора перед атомарной формулой и нечеткого неравенства (формулы, состоящие из атомарных формул с модераторами и нечетких неравенств, обрабатываются при распознавании аксиом).

Далее определены аксиомы исчисления LqS. *Каноническая цепь неравенств (КЦН)* определяется следующим образом. Формулы вида P и $q \cdot P$ (где $q\cdot$ — модератор, P — атомарная формула) являются КЦН. Если I и J — КЦН, то $(I \prec J)$ является КЦН.

Пусть дана секвенция S. Удалим из S все члены, которые не являются КЦН; тогда полученная секвенция называется *базовой подсеквенцией* секвенции S. Секвенция называется *аксиомой*, если базовая подсеквенция этой секвенции общезначима.

Любой секвенции с непустой базовой подсеквенцией сопоставляется (с помощью приведенного в тексте диссертации алгоритма) система строгих и нестрогих линейных неравенств с рациональными коэффициентами и рациональнозначными переменными. Доказана теорема о том, что такая секвенция является аксиомой, если и только если соответствующая система линейных неравенств несовместна.

В третьем разделе второй главы устанавливаются некоторые свойства исчисления LqS. Перечислим наиболее важные из них.

1. Все правила вывода и обратные к ним сохраняют общезначимость секвенций. Исчисление является семантически обоснованным.

2. Исчисление непротиворечиво.

3. Безантецедентное секвенциальное исчисление для классической двузначной логики вкладывается в исчисление LqS.

4. Для логики Lq не существует полное и семантически обоснованное исчисление.

5. Исчисление LqS неразрешимо.

6. Исчисление LqS полно для пропозиционального фрагмента логики Lq.

7. Пропозициональный фрагмент исчисления LqS разрешим.

8. Исчисление LqS допускает минус-нормализацию вывода.

Поясним последнее свойство. При так называемом поиске вывода заданной секвенции S *снизу вверх* находят секвенции, из которых секвенция S получается по одному из правил вывода, и для каждой найденной секвенции, если она не является аксиомой, находят секвенции, из которых она получается по одному из правил вывода, и т.д. При этом по секвенции S и правилу вывода требуется подбирать секвенции, которые являются посылками применения этого правила вывода, причем секвенция S должна быть заключением этого применения (т.е. требуется осуществлять *контрприменение* правила вывода). При контрприменении правил вывода исчисления LqS, вводящих какой-либо выбранный логический символ, секвенции-посылки подбираются детерминированным образом, кроме того случая, когда осуществляется контрприменение так называемого *минус-правила.* Каждое из минус-правил исчисления LqS (типа правила введения квантора существования в сукцедент секвенции в генценовском исчислении для классической двузначной логики) допускает бесконечный перебор термов для подстановки в посылку при контрприменении.

Для классической двузначной логики первого порядка известны секвенциальные исчисления (см., например, [4], а также [6, 11]), в которых устранен бесконечный перебор термов для подстановки при контрприменениях минус-правил соответствующих секвенциальных исчислений. Такое устранение бесконечного перебора термов названо *минус-нормализацией* [6]. Однако упомянутые работы (и все известные нам другие работы) не содержат доказательств равнообъемности предложенных секвенциальных исчислений и какого-либо из традиционных секвенциальных исчислений для классической двузначной логики.

В настоящей работе формулируется ограничение на термы для подстановки (подставляемый терм — один из конечного числа термов, входящих в заключение минус-правила) и доказывается, что при этом не изменяется объем выводимых секвенций. (Тогда обоснование минус-нормализации для исчисления классической двузначной логики следует из свойства 3.)

В последнем, четвертом разделе второй главы формулируется подлогика $Lq2$ логики Lq и секвенциальное исчисление $Lq2S$ для этой подлогики. В формулах подлогики $Lq2$ использование логической связки $\prec$ ограничено так, что распознавание аксиомы исчисления $Lq2S$ сводится к проверке несовместности системы линейных неравенств, каждое из которых имеет не более двух членов. Для решения таких задач существует сильно полиномиальный алгоритм (такой алгоритм описан в пятой главе данной работы), тогда как для распознавания аксиом исчисления LqS известны только полиномиальные алгоритмы. Подлогика $Lq2$ оказывается достаточно выразительной для первоначального моделирования областей нечетких знаний.

В **третьей главе** описан алгоритм поиска вывода и доказаны его свойства.

При поиске вывода заданной секвенции в исчислении LqS снизу вверх естественным образом строится *дерево поиска вывода*, корнем которого является исходная секвенция, непосредственными потомками корня — секвенции, являющиеся посылками контрприменения некоторого правила вывода к корневой секвенции, и т.д. (считается, что это дерево растет от корня вверх). В тот момент, когда все секвенции-листья оказываются аксиомами, дерево поиска вывода становится *деревом вывода*, т.е. вывод найден.

Несмотря на то, что бесконечный перебор термов для подстановки при контрприменениях минус-правил может быть устранен, задача подбора термов, хотя и упростилась, все равно остается. Воспользуемся так называемым *методом метапеременных*: вместо подстановки конкретного терма при контрприменении минус-правила отложим выбор терма, подставив вместо него уникальную метапеременную. Тем самым вместо дерева поиска вывода строится так называемая *заготовка вывода*, содержащая метапеременные. Тогда в ходе поиска вывода время от времени нужно проверять, можно ли подобрать термы-значения для метапеременных так, чтобы превратить заготовку вывода в дерево вывода.

Предлагаемый алгоритм $Prove$ [3′] ищет вывод заданной секвенции, строя заготовку вывода. При контрприменении минус-правила и введении метапеременной с ней ассоциируется *подстановочное множество* — конечное множество термов, содержащее все термы, только которые и достаточно подставлять вместо этой метапеременной в соответствии с ограничением минус-нормализации. Тогда для того, чтобы подобрать значения для метапеременных, превращающие заготовку вывода в дерево вывода, производится *унификация* — конечный перебор термов из подстановочных множеств метапеременных, входящих в заготовку вывода, и при этом переборе выбираются значения метапеременных, при которых заготовка превращается в дерево вывода.

В первом разделе третьей главы обсуждается идея алгоритма и определяются используемые понятия.

Во втором разделе описаны шаги главного алгоритма $Prove$ и вспомогательных алгоритмов: алгоритма, проверяющего, является ли секвенция аксиомой, и алгоритма, проводящего унификацию. Также сформулированы требования к вспомогательному алгоритму, называемому *тактикой поиска вывода*: такой алгоритм по заготовке вывода сообщает, когда нужно провести унификацию, и выбирает (a) секвенцию-лист S, (b) правило вывода (R), контрприменение которого следует осуществить, и (c) вхождение логического символа в S, которое может быть введено в S применением правила (R). В конце второго раздела приводится пример поиска вывода с комментариями.

В третьем разделе доказаны свойства вспомогательных алгоритмов и главного алгоритма $Prove$. Итогом является следующая теорема.

Теорема. *Пусть алгоритм Prove использует любую тактику поиска вывода; S — секвенция, подаваемая на вход алгоритма Prove. Тогда верны следующие утверждения.*

(1) Если алгоритм Prove выдал ответ «выводима», то секвенция S выводима в исчислении LqS.

(2) Если алгоритм Prove выдал ответ «невыводима», то секвенция S невыводима в исчислении LqS.

(3) Пусть секвенция S не содержит кванторов. Тогда алгоритм Prove выдает ответ «выводима», если секвенция S выводима в исчисле-

нии LqS, и выдает ответ «невыводима», если секвенция S невыводима в исчислении LqS.

В конце третьего раздела обсуждается выбор тактики поиска вывода, и описывается алгоритм-тактика, который выбирает минус-правила для контрприменения «равномерно», давая равные возможности разным вхождениям кванторов участвовать в контрприменениях.

В **четвертой главе** описана программная реализация алгоритма поиска вывода [3′]. Этот алгоритм реализован на языке программирования Java в виде интерфейса прикладного программирования (ИПП), который предоставляет программный интерфейс для доступа к своим функциональным возможностям. В начале этой главы кратко описывается назначение основных классов, сгруппированных в пакеты, и приводятся диаграммы классов, изображающие иерархии логических символов, термов и формул.

Затем определяется политика разделения объектов в программном представлении формул (с помощью синтаксических деревьев с возможно разделяемыми листьями) и секвенций. Эта политика, с одной стороны, позволяет однозначно и эффективно идентифицировать вхождение неатомарной подформулы в формулу, а, с другой стороны, позволяет существенно экономить память при поиске вывода, допуская общие формулы в разных секвенциях.

Далее описываются основные компоненты реализации алгоритма поиска вывода. Отметим следующие ключевые особенности.

В этом ИПП тактика поиска вывода задается с помощью интерфейса `Tactics`. Главный алгоритм поиска вывода может использовать любую тактику, являющуюся реализацией этого интерфейса. (Такая архитектура соответствует образцу проектирования *стратегия* [8].)

Каждое правило вывода представлено объектом, который имеет метод, осуществляющий контрприменение этого правила. Главный алгоритм поиска вывода делегирует контрприменение правила вывода, выбранного тактикой, самому этому правилу. Таким образом, правила вывода могут быть легко изменены.

Каждая система линейных неравенств, получаемая при распознавании аксиом, проверяется на несовместность вспомогательным алгоритмом, описанным в пятой главе данной работы, если все неравенства системы двучленные, иначе — функцией `FindInstance` системы компьютерной алгебры Mathematica (посредством J/Link — инструментария, связывающего программу на Java с Mathematica).

В конце четвертой главы приведен систематический обзор предоставляемого программного интерфейса для поиска вывода.

Объем написанного программного кода, включая реализацию алгоритма решения систем линейных двучленных неравенств, составляет около 9000 строк.

Пятая глава посвящена алгоритму проверки совместности систем строгих и нестрогих линейных двучленных неравенств с целочисленными коэффициентами и рациональнозначными переменными, а также алгоритму решения таких систем (под алгоритмом решения системы понимается алгоритм, находящий хотя бы одно решение системы, если оно существует, и сообщающий о несовместности системы в противном случае). Эти алгоритмы основаны на методе исключений переменных и одновременном с исключением переменной удалении избыточных неравенств так, что количество неравенств в системе, содержащих любые две переменные, не превосходит заранее фиксированную константу.

Алгоритм называется *сильно полиномиальным* (см., например, [9]), если (a) он полиномиален по числу шагов при реализации на машине Тьюринга, и (b) число элементарных арифметических операций (сложение, вычитание, умножение, деление, сравнение), выполняемых им над рациональными числами, ограничено полиномом от числа целых чисел на входе.

Упомянутые алгоритмы были предложены в [2]; там же установлено условие (b) определения сильно полиномиального алгоритма.

В настоящей диссертационной работе осуществлена детализация этих алгоритмов, причем добавлены необходимые шаги (без которых описания алгоритмов из [2] были некорректны), и разработан вспомогательный алгоритм удаления избыточных неравенств (см. [5′, 6′]). Доказана корректность описанных в настоящей диссертации алгоритмов, т.е. алгоритм проверки совместности систем выдает ответ «совместна», если входная система совместна, и выдает ответ «несовместна», если входная система несовместна (аналогичное утверждение установлено и для алгоритма решения систем).

Далее определена вычислительная модель, близкая к равнодоступной адресной машине с логарифмическим весовым критерием (см. [1]), и получена полиномиальная оценка временной сложности описанных алгоритмов [6′]. Установлена полиномиальность числа шагов этих алгоритмов при реализации на машине Тьюринга, что позволяет также доказать их сильную полиномиальность.

В последнем разделе пятой главы описана реализация алгоритма на языке Java в виде интерфейса прикладного программирования [4′]. В этой реализации участвуют 2 алгоритма с похожими шагами: упомянутый алгоритм решения систем с удалением избыточных неравенств и алгоритм, основанный на обычном методе исключений переменных. Одна из основных задач объектно-ориентированного программирования — вычленение общего поведения с целью повторного использования кода — решена с помощью *шаблонного метода* [8].

Наконец, приводятся результаты экспериментов по сравнению производительности реализованного автором диссертации алгоритма решения систем и алгоритма, представленного функцией `FindInstance` системы компьютерной алгебры Mathematica. Эксперименты проводились на одном и том же персональном компьютере. На решение систем из нескольких тысяч неравенств реализованный алгоритм затрачивает несколько секунд, а Mathematica — несколько десятков минут. На решение систем из нескольких десятков тысяч неравенств реализованный алгоритм затрачивает до нескольких десятков секунд, а Mathematica не заканчивает работу за 12 часов. Эти результаты говорят о том, что для решения систем линейных двучленных неравенств реализованный алгоритм намного эффективнее алгоритма, используемого в системе компьютерной алгебры Mathematica.

Шестая глава содержит список основных результатов, полученных в данной работе.

В **приложении А** приводится исходный текст небольшой программы, использующей разработанный ИПП для поиска вывода. Приведенный в следующем разделе этого приложения протокол поиска вывода формулы получен с помощью этой программы. В последнем разделе приложения А описан пример формализации нечетких знаний, заданных предложениями естественного языка, с помощью логики Lq и вывода новых нечетких знаний из заданных.

Приложение Б содержит исходный текст небольшой программы, использующей разработанный ИПП для решения систем линейных двучленных неравенств, а также служащей для сравнения производительности реализованного автором диссертации алгоритма решения систем и алгоритма, представленного функцией `FindInstance` системы компьютерной алгебры Mathematica.

Список литературы

[1] *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. / Пер. с англ. — М.: Мир, 1979.

[2] *Давыдок Д. В.* О совместности систем двучленных линейных неравенств. // Косовский Н. К., Тишков А. В. Логики конечнозначных предикатов на основе неравенств. — СПб: Издательство С.-Петерб. университета, 2000. — С. 246–268.

[3] *Заде Л.* Понятие лингвистической переменной и его применение к принятию приближенных решений. / Пер. с англ. — М.: Мир, 1976.

[4] *Кангер С.* Упрощенный метод доказательства для элементарной логики. // Математическая теория логического вывода. / Под ред. А. В. Идельсона, Г. Е. Минца. — М.: Наука, 1967. — С. 200–207.

[5] *Косовский Н. К.* Уровневые логики. // Записки научных семинаров Петербургского отделения Математического института РАН. — Т. 220. — СПб: Наука, 1995. — С. 72–82.

[6] *Маслов С. Ю., Минц Г. Е.* Теория поиска вывода и обратный метод. // Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. — М.: Наука, 1983. — С. 291–314.

[7] *Новак В., Перфильева И., Мочкорж И.* Математические принципы нечеткой логики. / Пер. с англ. — М.: Физматлит, 2006.

[8] Приемы объектно-ориентированного проектирования. Паттерны проектирования. / Пер. с англ. / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. — СПб: Питер, 2001.

[9] *Chandru V., Rao M. R.* Linear programming. // Algorithms and theory of computation handbook. / Ed. by M. J. Atallah. — CRC Press, 1999.

[10] *Ciabattoni A., Fermüller C. G., Metcalfe G.* Uniform rules and dialogue games for fuzzy logics. // LPAR / Ed. by F. Baader, A. Voronkov. — Vol. 3452 of *Lecture Notes in Computer Science.* — Springer, 2004. — Pp. 496–510.

[11] *Degtyarev A., Voronkov A.* Equality reasoning in sequent-based calculi. // Handbook of Automated Reasoning. / Ed. by A. Robinson, A. Voronkov. — Elsevier, 2001. — Vol. 1. — Pp. 611–706.

[12] *Gottwald S.* A Treatise on Many-Valued Logics. — Baldock: Research Studies Press, 2001.

[13] *Hähnle R.* Many-valued logic and mixed integer programming. // *Annals of Mathematics and Artificial Intelligence.* — 1994. — Vol. 12, no. 3-4. — Pp. 231–263.

[14] *Hájek P.* Metamathematics of Fuzzy Logic. — Dordrecht: Kluwer Academic Publishers, 1998.

[15] *Hájek P.* What is mathematical fuzzy logic. // *Fuzzy Sets and Systems.* — 2006. — Vol. 157, no. 5. — Pp. 597–603.

[16] *Metcalfe G., Olivetti N., Gabbay D. M.* Łukasiewicz logic: from proof systems to logic programming. // *Logic Journal of the IGPL.* — 2005. — Vol. 13, no. 5. — Pp. 561–585.

[17] *Olivetti N.* Tableaux for Łukasiewicz infinite-valued logic. // *Studia Logica.* — 2003. — Vol. 73, no. 1. — Pp. 81–111.

[18] *Zadeh L. A.* Fuzzy logic and approximate reasoning. // *Synthese.* — 1975. — Vol. 30. — Pp. 407–428.

Основные работы автора по теме диссертации

[1′] *Герасимов А. С.* Бесконечнозначная предикатная логика со связкой для усиления утверждений. // Современная логика: проблемы теории, истории и применения в науке: Материалы IX Общероссийской научной конференции (Санкт-Петербург, 2006). — СПб: 2006. — С. 348–350.

[2′] *Герасимов А. С.* Предикатная логика на основе секвенциального исчисления, предназначенная для моделирования непрерывных шкал. // Десятая национальная конференция по искусственному интеллекту с международным участием КИИ-06 (Обнинск, 2006): Труды конференции. — М.: Физматлит, 2006. — С. 339–347.

[3′] *Герасимов А. С.* Программная реализация поиска доказательств в бесконечнозначной предикатной логике, основанной на линейных неравенствах. // Материалы XVI Международной школы-семинара «Синтез и сложность управляющих систем» (Санкт-Петербург, 2006). / Под ред. О. Б. Лупанова. — М.: Изд-во механико-математического факультета МГУ, 2006. — С. 30–35.

[4′] *Герасимов А. С.* Разработка ИПП для решения задачи определения совместности систем линейных двучленных неравенств. // Технологии Microsoft в теории и практике программирования: Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада. — СПб: Изд-во Политехн. ун-та, 2005. — С. 161–162.

[5′] *Герасимов А. С., Косовский Н. К.* Истинно полиномиальный алгоритм определения совместности систем линейных двучленных неравенств. // Устойчивость и процессы управления. Труды международной конференции (Санкт-Петербург, 2005). / Под ред. Д. А. Овсянникова, Л. А. Петросяна. — СПб: СПбГУ, НИИ ВМ и ПУ, ООО ВММ, 2005. — С. 779–785.

[6′] *Герасимов А. С., Косовский Н. К.* Оценка сложности истинно полиномиального алгоритма проверки совместности систем линейных двучленных неравенств. // *Вестник С.-Петербург. ун-та. Сер. 10.* — 2006. — № 2. — С. 16–21.