

*На правах рукописи*

Дехтярь Михаил Иосифович

**Семантика и анализ сложности алгоритмических  
проблем динамических систем и языков,  
использующих логическое программирование**

Специальность 05.13.17 — Теоретические основы информатики

**Автореферат**  
диссертации на соискание учёной степени  
доктора физико-математических наук.

Переславль-Залесский — 2009

Диссертация выполнена на кафедре информатики Тверского государственного университета.

ОФИЦИАЛЬНЫЕ ОППОНЕНТЫ: член-корреспондент РАН,  
доктор физико-математических наук,  
профессор А.Л.Семёнов

доктор физико-математических наук,  
профессор И.А. Ломазова

доктор физико-математических наук,  
доцент В.Б. Новосельцев

ВЕДУЩАЯ ОРГАНИЗАЦИЯ: Институт прикладной математики  
им. М.В.Келдыша РАН

Защита диссертации состоится 20 ноября 2009 г. в 13-00 на заседании диссертационного совета ДМ002.084.01 при Учреждении Российской академии наук Институте программных систем РАН, расположенном по адресу: 152020, Ярославская область, Переславский район, с. Веськово, ул. Петра Первого, д. 4а.

С диссертацией можно ознакомиться в библиотеке Учреждения Российской академии наук Института программных систем РАН

Автореферат разослан

Ученый секретарь  
диссертационного совета

Пономарева С.М.

## Общая характеристика работы

**Актуальность.** Диссертация посвящена исследованию семантических проблем и анализу сложности алгоритмических проблем в таких областях теоретической информатики как дедуктивные базы данных, мультиагентные системы, языки логического и функционального программирования. Многие из рассматриваемых в работе тем традиционно относятся также к области искусственного интеллекта, поскольку рассматриваемые системы — активные базы данных, динамические дедуктивные базы данных и мультиагентные системы — демонстрируют черты разумного поведения. Все указанные области имеют не очень длинную историю (впрочем, как и вся теоретическая информатика), достаточно активно развиваются в последние десятилетия и являются вполне актуальными и в настоящее время. Это подтверждается, в частности, все возрастающим числом научных журналов и международных конференций, тематика которых включает перечисленные выше области.

Появление логического программирования обычно связывают с именами Дж. Робинсона, предложившего в 1965г. правило логического вывода, получившее название метода резолюций, Р. Ковальского, показавшего в начале 70-х, как на этом методе можно построить язык программирования, М. ван Эмдена, А. Колмероз и Д. Уоррена, внесших в середине 70-х большой вклад в практическую реализацию языка Пролог. В те же годы методы автоматического вывода активно разрабатывались в ленинградский логической школе. Предложенный С.Ю.Масловым “обратный метод вывода” был, по-существу, эквивалентен методу резолюции.

Логические программы применяются в различных системах, основанных на знаниях. Они могут выступать в качестве языка запросов для реляционных баз данных, представлять интенциональные компоненты дедуктивных баз данных, использоваться для вывода заключений в экспертных системах. В настоящей работе присутствуют четыре подхода к использованию логического программирования: во 2-ой главе логические программы задают ограничения целостности для транзакций и состояний в полных (реляционных) и частичных базах данных, в 3-ей главе логические программы с обновлениями определяют поведение динамических дискретных систем — осуществляют изменения их состояний, в 4-ой главе они выступают в качестве основных компонент интел-

лектуальных агентов, отвечающих за определение возможных действий агентов. Наконец, в 5-ой главе изучаются логические программы с интервальными вероятностями, которые используются для представления и вывода неточных знаний о фактах и событиях.

Вопросы интеллектуального выполнения внешних запросов на обновление в базах данных (БД) и базах знаний (БЗ), непрерывно поддерживающих ограничения целостности (ОЦ), исследовались с начала 80-х годов для класса пропозициональных БЗ. Первые шаги по формализации обновлений БЗ были предприняты П. Гарденфорсом (Gärdenfors P.) и его коллегами, которые предложили некоторый набор постулатов, описывающих обновления БЗ при появлении новых знаний. Позже Х. Катсуно (H.Katsuno) и А. Мендельзон (A. Mendelzon) уточнили эти постулаты для пропозиционального случая. Их аксиомы неявно выражают важный принцип *минимальных изменений*, выполняемых в исходных моделях для получения результирующих. В каждом конкретном случае операторы обновления определяются с помощью некоторого явного критерия близости двух моделей. Такие критерии были предложены в работах Ю. Деяла (U.Dayal), К.Форбуса (Forbus K.) и других авторов. В большинстве работ расстояние между двумя моделями оценивалось через мощность их симметрической разности. В случае логики 1-го порядка ситуация с обновлениями БД и БЗ усложняется из-за возможности различных конструктивных интерпретаций отрицания. Целый ряд семантик отрицания был предложен в работах К. Апта (K. Apt), Н. Бидуа (N. Bidoit), ван Гельдера (van Gelder), Т. Пжимужинского (T. Przymusiński), М.Гельфанда и В. Лифшица и др. Большинство авторов сосредотачивались на обновлениях БЗ. В этой области можно выделить работы Х. Алфереса (J. Alferes) и Л.Перейры (L. Pereira), С. Чери (S. Ceri), Т. Пжимужинского и Х. Тернера (H. Turner), Т. Айтера (T. Eiter) и Г. Готлоба (G. Gottlob), Д. Лобо (J. Lobo) и др. В некоторых подходах при обновлениях изменяется не только БД, но и интенциональная компонента БЗ. Кроме того, общей чертой предлагаемых методов обновления БЗ является то, что в качестве возможных результатов рассматриваются только специальные (минимальные, wf-, совершенные, стабильные и др.) модели. Это вполне оправдано для систем с интенциональными знаниями. Однако этот подход не применим к БД, где ОЦ не должны изменяться и любая их модель является допустимой. Проблема наименьших допустимых обновлений для БД была

рассмотрена Н. Спиратосом (N. Spyrtatos) с соавторами для очень простого подкласса ограничений целостности – правил вида  $l \leftarrow l_1$ . В главе 2 эта проблема исследована в гораздо более общих случаях.

В области динамических (дедуктивных) баз данных с обновлениями (ДБД) внимание в первую очередь было уделено средствам задания обновлений, анализу их выразительных возможностей и сложности. Здесь можно отметить исследования С. Абитбуля (S. Abiteboul) и В. Виану (V. Vianu), А. Боннера (A. Bonner) и М. Кифера (M. Kifer), С. Манчанды (S. Manchanda) и Д. Уоррена (D. Warren) и др. При этом обновления, нарушающие ОЦ, как правило, считались недопустимыми. В главе 3 это ограничение снимается, изучаются свойства живучести ДБД, взаимодействующих со внешней средой, при этом каждая из взаимодействующих сторон может нарушать ОЦ.

Понятие интеллектуального агента сейчас является, пожалуй, основным в искусственном интеллекте (ИИ). Например, на нем построено все изложение в классическом учебнике по ИИ С. Рассела и П. Норвига. Исследования систем взаимодействующих интеллектуальных агентов занимают одно из ведущих мест в области ИИ и прикладного программирования. В первую очередь это объясняется тем, что мультиагентные системы (МА-системы) имеют широкую сферу применений, включающую такие разные области как управление бизнес-процессами, электронная торговля, Интернет-навигация, социальные и военные приложения и др. Такое разнообразие приложений привело к появлению многочисленных различных подходов к определению понятий агента и МА-системы (использующих, к тому же, разные уровни абстрагирования от конкретных систем). Среди авторов основных концепций и посвященных МА-системам монографий выделим М. Вулдриджа (M. Wooldridge) и Н. Дженнингса (N. Jennings), А. Рао (A. Rao) и М. Джорджефа (M. Georff), В.С. Субраманиана (V. S. Subrahmanian) с соавторами, И. Шохам (Y. Shoham) и К. Лейтон-Брауна (K. Leyton-Brown) Отметим также достаточно обширные обзоры на русском языке В.И. Городецкого и Д.А. Поспелова и монографию В.Б. Тарасова, в которых рассмотрены многочисленные предложенные модели, а также промышленные инструментальные средства, предназначенные для реализации МА-систем.

Рассматриваемая в диссертации проблема верификации МА-систем, относится к классу проблем, который под названием model checking (проверка на моделях программ) изучается уже ряд лет для абстракт-

ных систем (диаграмм) переходов и конкретных программных систем. Содержательно, они формулируются так: по спецификации динамической системы (программы) определить, выполняется ли для нее некоторое свойство, выраженное в некотором формальном языке (как правило, языке какой-либо временной логики). Началом исследований в этой области послужили работы Е. Кларка (E. Clarke), Е. Эмерсона (E. Emerson) и Дж. Сифакиса (J. Sifakis) в начале 80-х, которые были продолжены М. Варди (M. Vardi), П. Вольпером (P. Wolper), Д. Пеледом (D. Peled), О. Купферман (O. Kupferman) и др. Результаты этого этапа были подытожены в классической монографии Е. Кларка с соавторами<sup>1</sup>. Современное состояние области отражено в монографии К. Байера (C. Baier) и Ж.-П. Катоена (J.-P. Katoen)<sup>2</sup>. В частности, в нее вошли результаты К. Куркубетиса (C. Courcoubetis) и М. Янакакиса (M. Yannakakis) по алгоритмам верификации вероятностных систем, которые мы используем при верификации вероятностных МА-систем в параграфе 4.4. Среди отечественных исследований по анализу поведения и верификации программ и динамических систем выделим работы Н.В. Евтушенко, В.А. Захарова, И.А. Ломазовой, В.Н. Непомнящего, А.К. Петренко, В.М. Смелянского, В.А. Соколова, Н.В. Шилова.

Отметим, что в разных областях ИИ постоянно возрастает интерес к исследованию систем, включающих стохастические элементы. Одна из таких областей — вероятностные базы знаний. Для задания их интенциональной компоненты используются разные варианты вероятностного логического программирования. Они рассматривались в работах Д. Пула (D. Poole), и П. Хаддави (P. Haddawy), К. Барала (C. Baral), М. Гельфонда и Д. Раштона (J. Rushton). Их работы относятся к так называемому Баейсовскому подходу.

В ряде случаев бывает трудно точно оценить вероятность тех или иных фактов или событий. Тогда используют интервалы вероятностей, которые являются ограничениями для множества возможных истинных точечных вероятностных распределений. Несколько вариантов вероятностного логического программирования, основанное на Баейсовском подходе, определил и изучал Т. Лукашевич (T. Lukasiewicz). В его программах интервальное правило вида  $(H|B)[c_1, c_2]$  означает, что  $c_1 P(B) \leq P(H \wedge B) \leq c_2 P(B)$ . Другой подход, основанный на вероятностной вы-

<sup>1</sup>Clarke E.M., Grumberg O. and Peled D., Model checking, MIT Press, 2000. 314 p.

<sup>2</sup>Baier C., Katoen J.-P. Principles of model checking. MIT Press, 2008. 975 p.

полнимоности (PSAT) берет свое начало в работе Буля по логике и теории вероятностей<sup>3</sup> В ней он рассматривал конечное множество пропозициональных формул  $F_1, \dots, F_m$  над множеством атомов  $A_1, \dots, A_n$  и предполагал, что вместо знаний об истинности или ложности формул известны их вероятности  $Prob(F_i) = p_i, 1 \leq i \leq m$ . Буля интересовало можно ли так назначить вероятности  $Prob(A_j) = q_j, 1 \leq j \leq n$ , пропозициональным атомам, чтобы все утверждения о вероятностях формул  $F_1, \dots, F_m$  оказались верны. Этот подход естественным образом переносится на интервальные вероятности. Основанные на нем формализмы логических программ с интервальными вероятностями были предложены в работах Р.Энга (R.Ng) и В.С.Субраманиана (V.S.Subrahmanian), В. Лакшманана (V.Lakshmanan) и Ф.Садри (F. Sadri). Семантика интервальных вероятностных программ задается с помощью множества возможных миров, в которых для каждого события известно, произошло оно или нет. Вероятность события есть сумма вероятностей всех миров, в которых событие произошло. В работах Р.Энга и В.С.Субраманиана для рассматриваемого класса программ была также предложена семантика, основанная на вычислении минимальной неподвижной точки рекурсивного определения, задаваемого программой, и содержалось утверждение о совпадении этих двух семантик. В главе 5 это утверждение уточняется и дается точное описание семантики вероятностных логических программ.

Вообще, семантика рекурсивных определений вызывает постоянный интерес, поскольку рекурсия является одним из самых мощных средств разработки программ как в традиционных императивных языках программирования, так и в языках логического и функционального программирования, где она является основным таким средством. Многие результаты о рекурсивных программах собраны в отдельной главе из книги З. Манна (Z.Manna)<sup>4</sup>. В ней различные вычислительные стратегии оцениваются по отношению к вычислению семантики наименьшей неподвижной точки. В частности, показано (с ссылкой на диссертацию Морриса), что *правило самой левой-самой внутренней замены* (“вызов по значению”) не является правилом вычисления неподвижной точки для класса всех рекурсивных программ. В то же время это правило используется в операционной семантике некоторых языков программиро-

<sup>3</sup>G. Boole. *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*. Macmillan, London. 1854.

<sup>4</sup>Манна З. Теория неподвижной точки программ. // Кибернетический сборник. М., вып. 15, 1978. С.38–100.

вания. В частности, с его помощью определена семантика языка Рефал. Этот язык функционального программирования был создан В.Ф. Турчиным в 1966 году и успешно развивается уже более 40 лет им самим и группами его учеников в Москве, Нью-Йорке и Переславле-Залесском. Поэтому естественно возникла задача выделения класса рекурсивных программ, для которых правило вызова по значению вычисляет наименьшую неподвижную точку, и выяснения отношения Рефал-программ к этому классу. Еще одна проблема, связанная с языком Рефал, — это сложность выполнения одного шага Рефал-машиной.

Как мы уже отмечали, алгоритмические и сложностные проблемы, наряду с семантическими, занимают центральное место в нашей работе. Классическая теория алгоритмов занималась, в основном, классификацией алгоритмических проблем на разрешимые и неразрешимые, уделяя основное внимание последним. С появлением и развитием вычислительной техники алгоритмы заняли одно из важнейших мест как в практике вычислений, так и в науке о них — информатике (computer science). В выдержавшей уже три издания книге Д. Харела (D. Harel)<sup>5</sup> утверждается: “Алгоритмика — это больше, чем ветвь информатики. Она является ядром информатики и, со всей определенностью можно сказать, имеет важное значение для большинства наук, бизнеса и технологии.” Появившаяся в 60-х годах XX века теория сложности алгоритмов и вычислений — сосредоточилась на классификации и изучении сложности разрешимых проблем. Ее “окончательная” цель состоит в определении сложности любой корректно сформулированной задачи. Другие цели связаны с установлением и пониманием отношений между различными вычислительными феноменами. Наибольших успехов теория добилась как раз в достижении этих целей. Многие результаты по сложности вычислений приведены в известном учебнике К.Пападимитриоу (C.Papadimitriou) и вошли в недавние монографии Л.Хемаспандры (L.Nemaspaandra) и М. Огихары (M.Ogihara)– 2002г., Д.Козена (D.Kozen) – 2006г., О. Голдрейча (O.Goldreich) – 2008г.

Прежде всего нужно отметить выделение классов  $P$  и  $NP$  задач, разрешимых за полиномиальное время детерминированными и недетерминированными алгоритмами, соответственно. Класс  $P$  считается формализацией содержательного понятия “практически разрешимой зада-

---

<sup>5</sup>Harel D. Algorithmics: the spirit of computing. 3rd ed. Pearson Education Ltd., 2004. 536p.

чи”. Хотя вопрос о совпадении или различии этих классов “ $P = NP?$ ”<sup>6</sup> остается нерешенным, обнаружение в последние 38 лет многочисленных  $NP$ -полных проблем (в том числе и в данной работе) позволило связать между собой задачи в совершенно различных областях информатики и математики. Кроме того, для многих конкретных алгоритмических проблем разными авторами была установлена полнота в больших сложных классах  $PSPACE$ ,  $EXPTIME$  и др.

Еще одно направление в теории сложности — сложность алгоритмов или “информационная” сложность — получило начало в работах А.Н.Колмогорова, А.А.Маркова и их учеников<sup>7</sup>. Здесь изучаются размеры (длины) программ, решающих соответствующую алгоритмическую проблему, при отсутствии или наличии некоторых ограничений на сложность (время, память) вычислений. Сложностью  $M^f(A^x)$  аппроксимации начального отрезка длины  $x$  проблемы  $A$  мы называем длину самой короткой программы, вычисляющей  $A(y)$  при  $y \leq x$  за время  $\leq f(y)$ . Представленные в работе результаты о сложности аппроксимации были получены автором в конце 70-х – начале 80-х годов. За рубежом исследования связи между колмогоровской и вычислительной сложностью продолжались в работах Р. Бука (R.Book), Ю. Хартманиса (J.Hartmanis), Д. Лутца (J.Lutz) и др. В частности, недавно были обнаружены неожиданные связи между ограниченной колмогоровской сложностью и размерностью Хаусдорфа в геометрии фракталов. Среди отечественных авторов имеющих серьезные достижения в теории сложности алгоритмов и вычислений отметим: А.П. Бельтюкова, Н. К. Верещегина, Д.Ю. Григорьева, Ю.В. Матиясевича, А.А. Разборова, А.Л. Семенова, А.О. Слисенко.

**Цели работы.** Цели работы состояли в разработке теоретических основ создания программных систем для новых информационных технологий, исследовании информационных структур и в анализе основанных на знаниях динамических моделей взаимодействующих информационных процессов и алгоритмов верификации их поведения.

Для достижения этих целей в работе исследовались следующие проблемы.

---

<sup>6</sup>Формально этот вопрос был поставлен С. Куком в 1971г., но его содержательная постановка в терминах сложности доказательств была сделана К.Геделем в письме фон Нейману в 1956г.

<sup>7</sup>За рубежом независимо тот же подход был предложен Р.Соломоновым (R.Solomonoff) и Г.Чейтиным (G.Chaitin).

- 1) Проблема корректного выполнения обновлений полных (реляционных) и “частичных” баз данных с сохранением статических и динамических ограничений целостности.
- 2) Разработка моделей дискретных динамических систем, основанных на логических программах с обновлениями, и анализ устойчивого поведения таких систем во взаимодействии с внешней средой.
- 3) Разработка моделей детерминированных, недетерминированных и вероятностных мультиагентных систем и верификация динамических свойств поведения таких систем.
- 4) Исследование семантик языка дизъюнктивных логических программ с интервальными вероятностями и алгоритмов их вычисления.
- 5) Семантика и верификация программ языка функционального программирования Рефал и анализ сложности его вычислений.
- 6) Построение теории аппроксимации алгоритмических проблем на начальных отрезках и исследование сложности аппроксимации трудных проблем в различных сложностных классах.

**Методы исследования.** В диссертации используются методы таких разделов теоретической информатики как теория баз данных, логическое программирование, теория сложности алгоритмов и вычислений, искусственный интеллект, а также математических дисциплин: математической логики и теории алгоритмов.

**Научная новизна и основные положения, выносимые на защиту.** Все полученные в работе результаты являются новыми. Наиболее существенными из них являются следующие.

1. Установлены оценки сложности задач, связанных с проблемой наименьших достаточных изменений (НДИ) при выполнении обновлений БД и предложены эвристические алгоритмы для решения проблемы НДИ для статических и динамических ограничений целостности и алгоритмы для максимального расширения обновлений и упрощения ограничений целостности.
2. Определена формальная модель взаимодействия интеллектуального агента – динамической дедуктивной базы данных – с внешней средой и установлена сложность верификации свойств устойчивого поведения (перспективности, стабильности и гомеостатичности)

для различных классов таких систем, определяемых типом логической программы с обновлениями в качестве интеллектуальной компоненты системы и типом условия, выделяющего допустимые состояния.

3. Определены формальные модели детерминированных, недетерминированных и вероятностных мультиагентных (МА-) систем, базирующиеся на архитектуре IMPAST, и получены оценки сложности верификации свойств поведения МА-систем, выразимых средствами временных логик, для различных классов МА-систем, различающихся типами используемых в качестве интеллектуальной компоненты агентов логических программ, количеством агентов в системе, количеством различных видов сообщений и количеством параметров в описаниях состояний.
4. Ранее определенный класс логических ( $p$ -) программ с интервальными вероятностями расширен до класса дизъюнктивных логических ( $dp$ -) программ. Для обоих классов изучено соотношение между теоретико-модельной семантикой “возможных миров” и семантикой минимальной неподвижной точки. Доказана NP-полнота проблемы совместности (существования модели) и проблемы следования для  $p$ - и  $dp$ -программ. Получено явное описание множества моделей, задаваемых  $p$ - и  $dp$ -программами. Построены алгоритмы для вычисления теоретико-модельной семантики  $p$ - и  $dp$ -программ, использующие специальные эвристики для сокращения перебора.
5. Определен класс рекурсивных программ, для которых семантика вызова “по значению” совпадает с семантикой минимальной неподвижной точки и показано, что Рефал-программы входят в этот класс. Доказано, что задача синтаксического отождествления, решаемая на каждом шаге работы Рефал-машины является NP-полной и получены верхние оценки сложности этой задачи.
6. Исследована сложность аппроксимации трудных проблем более простыми на начальных отрезках входных данных. В частности, получены оценки сложности аппроксимации проблем, которые полны или трудны для различных детерминированных и недетерминированных сложностных классов и установлена связь между верхними

и нижними оценками сложности вычисления проблемы и сложностью ее аппроксимации.

**Теоретическая и практическая значимость.** Работа носит теоретический характер. Результаты о сложности задачи отождествления для языка Рефал могут быть использованы (и использовались автором) при разработке компиляторов этого языка. Алгоритмы выполнения “интеллектуальных” обновлений могут быть использованы при разработке новых СУБД. Результаты о высокой сложности задач анализа поведения динамических баз данных и мультиагентных систем с одной стороны указывают на тупики, с которыми могут столкнуться разработчики систем верификации, а с другой — подталкивают к созданию интерактивных программных средств для моделирования и анализа свойств таких систем (набросок требований к программной системе такого типа приведен в [?]). Проведенное исследование семантики языка вероятностных программ, введенного Энгом и Субраманианом, показало, что в качестве языка представления интенциональных знаний в вероятностной БД он чересчур сложен и нуждается в существенных упрощениях.

Материалы диссертации могут использоваться (и уже неоднократно использовались автором) при чтении специальных курсов студентам и аспирантам, обучающимся математическим основам информатики. Они также послужили основой при выполнении ряда выпускных и дипломных работ и магистерских диссертаций на кафедре информатики Тверского государственного университета.

**Апробация работы** Содержание диссертации докладывалось на многих отечественных и международных конференциях и симпозиумах. Среди них выделим следующие:

Всесоюзная конференция по математической логике ( Кишенев, 1976; Новосибирск, 1979; Новосибирск, 1984), 8-th Symp. on Mathematical Foundation of Computer Science ( Praga, 1979), Всесоюзная научная конф. “Проблемы совершенствования, тестирования, верификации и отладки программ” (Рига, 1986), 10th International Symp. on Logic Programming (USA, Ithaca, 1994), 12th International Conf. on Logic Programming (Japan, 1995), 2nd and 3-rd Int. A.P.Ershov Memorial Conference “Perspective of Systems Informatics” ( Новосибирск, 1996 и 1999), 4th International Symposium Logical Foundation of Computer Science’97, (Ярославль, 1997), 14th

International Conference on Logic Programming (London, 1997), 4th International Conference Logic Programming and Nonmonotonic Reasoning'97 (Dagstuhl Castle, Germany, 1997), Joint International Conf. and Symp. on Logic Programming (Manchester, 1998), Seventh International Colloquium on Numerical Analysis and Computer Science with Applications (Plovdiv, Bulgaria, 1998), 13th International Conference on Logic Programming, (New Mexico, USA, 1999), 5th International Conference, Logic Programming and Nonmonotonic Reasoning'99 (El Paso, USA, 1999), First International Symp. on Foundations of Information and Knowledge Systems (FoIKS 2000) (Burg, Germany, 2000), First Int. Conf. on Computational Logic, CL-2000 (London, 2000), Конференция, посвященная 90-летию со дня рождения Алексея Андреевича Ляпунова (Новосибирск, 2001), International Workshop "Logic and Complexity in Computer Science" (Creteil, France, 2001), 8th European Conference on Logics in Artificial Intelligence (Cosenza, Italy, 2002), IEEE International Conf. on Artificial Intelligence Systems. (Дивногорское, 2002 и 2003), Second St.Petersburg Days of Logic and Computability, (St.Petersburg, Russia, 2003), Первая Всероссийская научная конференция "Методы и средства обработки информации", (Москва, МГУ, 2003), I-ой, III-ий, IV-ый, V-ый Межд. научно-практический семинар (конференция) "Интегрированные модели и мягкие вычисления в искусственном интеллекте" (Коломна, 2003, 2005, 2007, 2009), IX-ая, X-ая и XI национальная конференции по искусственному интеллекту с международным участием (Тверь, 2004; Обнинск, 2006; Дубна, 2008), 20th International Conference on Logic Programming (San Malo, France, 2004), 8th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'05)(Cosenza, Italy, 2005).

Результаты диссертации также докладывались автором на семинарах в Новосибирском госуниверситете (1976-1981), Институте математики СОАН (1976-1981), Вычислительном центре СО АН (1978-1981), Тверском госуниверситете (1982-2008), University Paris12-Val de Marne (Paris, 1996), Tel-Aviv University (1998, 2004), University of Maryland (College Park, 1998), University of Nantes (1999, 2000, 2001, 2002, 2003), University of Kentucky (Lexington, 2002).

В разные годы работа поддерживалась грантом INTAS (Grant 94-2412) и грантами РФФИ 95-01-01321, 96-01-00395, 97-01-00973, 98-01-00204, 99-01-00374, 01-01-00278, 02-01-00652, 04-01-00015, 04-01-00565, 05-01-01006, 07-01-00637, 08-1-00241.

**Публикации.** Список публикаций по теме диссертации приведён в конце автореферата и включает 47 печатных работ, 7 из которых опубликованы в изданиях, входящих в список ВАК (они выделены полужирным шрифтом), и еще 2 работы [48,49] приняты в печать в таких изданиях. 12 работ опубликованы в томах серии Lecture Notes in Computer Science (Artificial Intelligence) издательства Springer.

**Структура и объем диссертации.** Диссертация состоит из введения, семи глав основной части и списка литературы. Общий объем работы — 383 стр. Список литературы содержит 190 наименований.

## Краткое содержание работы

Глава 1 носит вспомогательный характер. В ней собраны основные обозначения и определения, связанные со сложностными классами и логическими программами, которые широко используются в остальной части работы.

В главе 2 рассматривается проблема, которая связана с возможностями БД нового поколения интеллектуально выполнять внешние запросы на обновление. Системы управления такими базами данных можно рассматривать как интеллектуальные агенты, цель которых — выполнять получаемые из внешней среды приказы на обновление и при этом поддерживать корректность состояний базы данных и их транзакций (переходов из одного состояния в другое). Знания о корректности представляются некоторыми логическими средствами, задающими *ограничения целостности* — условия, которым должны удовлетворять состояния и переходы.

Основное содержание этой главы основано на работе автора [?], которая, в свою очередь, продолжает и расширяет результаты по этой проблеме, полученные вместе с соавторами в работах [?, ?, ?, ?, ?, ?, ?]. В этих работах указанная проблема изучалась для статических ограничений целостности, фильтруемых допустимых состояний БД. Некоторые результаты для них представлены в параграфе 2.4. Основное же внимание в этой главе уделяется более широкому классу — *динамическим ограничениям целостности (ДОЦ)*, которые выделяют допустимые транзакции, т.е. переходы из одного состояния БД в другое. Примерами динамических ограничений целостности являются, например,

условия: “зарплата не должна уменьшаться”, “на должность ведущего научного сотрудника можно назначать лишь того, кто работает в должности старшего научного сотрудника”, “принимать заказ на продажу  $N$  принтеров следует лишь тогда, когда их текущее количество на складе не меньше  $N + 2$ ” и т.п.

Более точно рассматриваемая задача формулируется следующим образом: по заданной логической программе  $\Phi$ , формализующей динамические ограничения целостности (ДОЦ), исходному состоянию БД  $I$  и запросу на обновление  $\Delta$ , включающему множество  $\Delta^+$  добавляемых к  $I$  фактов, множество  $\Delta^-$  фактов, которые нужно удалить из  $I$ , и множество  $\Delta^*$  пар фактов, в которых первый должен быть заменен на второй, требуется найти такое **минимальное изменение**  $I'$  состояния  $I$ , что при переходе от  $I$  к  $I'$  выполнено обновление  $\Delta$  и не нарушено ни одно из ограничений  $\Phi$  (т.е. гарантируется, что  $\Delta^+ \subseteq I'$ ,  $I' \cap \Delta^- = \emptyset$ , для всякой пары  $(a, b) \in \Delta^*$ , если  $a \in I$ , то  $a \notin I'$  и  $b \in I'$ , и  $\langle I, I' \rangle \models \Phi$ ).

Как мы уже отмечали, мы называем эту проблему *проблемой наименьших достаточных изменений* (проблемой *НДИ*). Поскольку результирующих состояний, удовлетворяющих указанным условиям, может быть несколько, мы также рассматриваем *проблему перечисления всех таких состояний* — *проблему ПНДИ*.

В качестве критерия близости состояний БД чаще всего в работах по обновлениям баз данных и знаний использовался критерий, основанный на симметрической разности состояний. В [?] мы предложили соединить этот критерий с критерием максимального пересечения. Согласно этому комбинированному критерию вначале минимизируется множество удаляемых из БД известных данных, а затем минимизируется добавление новых данных. Представляется, что это лучше соответствует идеологии баз данных.

Проблемы НДИ и ПНДИ рассматриваются для двух классов баз данных. Один из них — это обычные реляционные БД, в которых отрицательная информация представлена неявно: отрицание некоторого факта имеет место, если этот факт отсутствует в текущем состоянии БД. Такие базы мы называем *полными*, поскольку для каждого факта известно истинен он или ложен в данный момент. Другой класс — это БД с явным представлением отрицательной информации. Состояния таких БД содержат как позитивные факты, так и их отрицания. Некоторый факт

истинен, если он входит в текущее состояние БД, и ложен, если это состояние содержит его отрицание. Если же ни факт, ни его отрицание не входят в текущее состояние БД, то его значение является неизвестным. Поэтому базы данных из этого класса называются *частичными*.

В параграфе 2.2 приводятся определения основных понятий: логического языка и задаваемых в его терминах статических (ОЦ) и динамических (ДОЦ) ограничений целостности. Они представляются с помощью обобщенных логических программ. Определены понятия частичного и полного состояний БД, выполнения ДОЦ на переходе из одного состояния в другое. Обновления задаются множествами команд вида *ДОБАВИТЬ(факт)*, *УДАЛИТЬ(факт)* и *ЗАМЕНИТЬ(факт1, факт2)*. Они выполнены на переходе, если факты из команд *ДОБАВИТЬ* присутствуют в результирующем состоянии, факты из команд *УДАЛИТЬ* в нем отсутствуют и каждый *факт1* из команд *ЗАМЕНИТЬ*, присутствующий в исходном состоянии, заменен на соответствующий *факт2* – в заключительном.

В параграфе 2.3 определена совместность ДОЦ  $\Phi$  с обновлением  $\Delta$ , которая означает, что множество переходов  $Tr(\Phi, \Delta)$ , удовлетворяющих  $\Phi$ , на которых выполнено  $\Delta$ , содержит хотя бы один переход. Совместность же  $\Phi$  с  $\Delta$  для состояния БД  $I$  означает, что имеется хотя бы одно состояние  $I'$  такое, что переход  $\langle I, I' \rangle \in Tr(\Phi, \Delta)$ , т.е. можно так выполнить запрос на обновление  $\Delta$ , что ДОЦ  $\Phi$  будут выполнены на соответствующем переходе. Показано, что обе проблемы совместности являются NP-полными (теорема 2.1). На множестве переходов определяется *сильный оператор непосредственной выводимости*  $T_{\Phi}^{\epsilon}(\langle I, I' \rangle)$ , который является монотонным и имеет эффективно вычислимую неподвижную точку. Во 2-ом пункте определяется понятие *консервативного оператора обновления*  $\Psi_{\Phi, \Delta}$ , который для каждого состояния БД  $I$  такого, что  $\Delta$  совместно с  $\Phi$  для  $I$ , выдает ближайшее к  $I$  состояние  $\Psi_{\Phi, \Delta}(I) = I'$ , такое что переход  $\langle I, I' \rangle \in Tr(\Phi, \Delta)$ .

В параграфе 2.4 приводятся некоторые результаты для случая статических ОЦ. В частности, исследована сложность некоторых алгоритмических проблем разрешения, связанных с проблемой НДИ. Даже в простейших случаях они оказываются достаточно сложными. Так, например, если статические ОЦ заданы базисной дефинитной логической программой, то проблема включения литерала в результат-решение проблемы НДИ является coNP-полной проблемой для частичных БД и  $\Sigma_2^P$ -

полной проблемой для полных БД (теоремы 2.3 и 2.4). В этом же параграфе также рассматриваются и анализируются несколько алгоритмов для решения проблем НДИ и ПНДИ в статическом случае.

Результаты в параграфах 2.5 - 2.9 относятся к БД с динамическими ограничениями целостности. Параграф 2.5 посвящен проблемам НДИ и ПНДИ для полных БД. В его первом пункте приведен недетерминированный алгоритм для проблемы НДИ. В теореме 2.7 доказано, что эта проблема для реляционных БД является полной в классе сложности  $FNP//OptP[O(\log n)]$ . Во втором пункте предложен алгоритм ПНДИ, решающий соответствующую проблему за время линейное от размеров всех параметров и экспоненциальное от количества потенциально добавляемых и удаляемых фактов.

В параграфе 2.6 для частичных БД построен алгоритм ЧНДИ, который решает проблему НДИ за полиномиальное время (теорема 2.9). Предложен также алгоритм для решения проблемы перечисления всех минимальных корректных результатов обновлений (проблемы ПНДИ) и получены оценки его сложности, которые экспоненциально зависят от числа фактов, не попавших в обновление, и полиномиально — от размера ДОЦ (теорема 2.10).

Из этих результатов следует, что каждый новый явно добавляемый или удаляемый факт существенно уменьшает пространство поиска возможных решений и, следовательно, время их поиска. Каждое упрощение ДОЦ — удаление ограничения или литерала из его условия — также упрощает поиск решения, так как уменьшает время на проверку выполнения ДОЦ. Поэтому в параграфах 2.7 - 2.9 для поиска практического решения проблемы НДИ мы распространяем на динамические ограничения целостности метод предварительного корректного расширения исходного запроса на обновление и одновременной оптимизации ДОЦ, примененный в работах [?, ?, ?] для статических ОЦ. Такая оптимизация обновлений не зависит от исходного состояния БД и может быть проведена на этапе трансляции запроса на обновление. Идея метода состоит в том, что запрос на обновление  $\Delta$  можно последовательно и корректно расширить, итеративно “протаскивая” его через ДОЦ  $\Phi$ . Корректность такого расширения означает, что оно сохраняет неизменным множество допустимых для  $\Phi$  и  $\Delta$  переходов  $Tr(\Phi, \Delta)$ . Целью является получение после конечного числа итераций максимального корректного расширения  $\Delta'$  исходного запроса на обновление и наиболее про-

стого представления  $\Phi'$  ограничений целостности. Использование пары  $\Phi', \Delta'$  вместо  $\Phi, \Delta$  может существенно ускорить алгоритмы поиска НДИ. Результаты, представленные в параграфе 2.7 дают теоретическое обоснование предложенного подхода к оптимизации обновлений на этапе трансляции и имеют ясный практический смысл. В нем определены операторы, применение которых к исходным ДОЦ и обновлению доставляет оптимальную (по отношению к порядку на парах) пару вида *обновление/ДОЦ*. Эта пара состоит из максимально расширенного обновления и минимальных по размерам ДОЦ, эквивалентных исходной паре. В пункте 2.7.2 описаны два метода упрощения ДОЦ, а в пункте 2.7.3 для каждого из видов БД определен прямой и обратный операторы расширения обновлений. В параграфе 2.8 предложен алгоритм, который строит такую оптимальную пару для класса частичных БД за полиномиальное время (теоремы 2.12, 2.13). Существование аналогичного алгоритма для полных БД маловероятно, поскольку предложение 2.15 (2) показывает, что в таком случае выполнялось бы равенство  $P=NP$ . Поэтому в общем случае в параграфе 2.8 предлагается использовать для полных БД итерацию прямого и обратного операторов расширения обновлений, которая дает некоторую аппроксимацию оптимальной пары (*обновление/ДОЦ*). Более того, для одного практически интересного класса полных БД, а именно для полных БД, имеющих ДОЦ с позитивными телами правил, оптимальные пары вида (*наибольшее корректное обновление/ наиболее простые эквивалентные ДОЦ*) все же можно находить за полиномиальное время (теорема 2.17).

Глава 3 посвящена анализу поведение пары взаимодействующих агентов *динамическая дедуктивная база данных – внешняя среда* на протяжении произвольного конечного или даже потенциально бесконечного числа шагов. В параграфе 3.1 определяются некоторые достаточно общие рамки для классификации и анализа многих естественных свойств устойчивого поведения дискретных интерактивных систем. Дискретная природа такой системы  $S$  проявляется себя, по меньшей мере, в двух отношениях. Во-первых,  $S$  ведет себя пошагово, что позволяет фиксировать ее *состояния* до и после каждого шага. Во-вторых, состояния  $S$  могут, как правило, быть описаны с помощью конечного числа отношений на значениях параметров системы. Мы будем предполагать, что фиксировано некоторое пространство  $\Sigma$  состояний  $S$  и что действия  $S$  (т.е. переходы системы из одного состояния в другое) представляются с помощью

некоторого бинарного отношения  $\vdash_S$  на  $\Sigma$ . Состояния системы делятся на *допустимые* и *недопустимые*, в которых система не может функционировать должным образом. Это подразделение определяется некоторым свойством состояний  $\Phi$ , которое играет роль *ограничения целостности (ОЦ)*. (ОЦ, рассматриваемые в этой главе, в предыдущей назывались *статическими*.) Тот факт, что состояние  $\mathcal{E}$  допустимо, т.е. удовлетворяет ОЦ  $\Phi$ , будем обозначать как  $\mathcal{E} \models \Phi$ . Интерактивная природа поведения систем, которые мы рассматриваем, подразумевает существование *активной внешней среды* системы и возможности влияния этой среды на состояния системы в процессе ее эволюции. Действия внешней среды на состояния системы представляется с помощью отдельного бинарного отношения  $\xrightarrow{d}$  на множестве  $\Sigma$ , которое мы будем называть *возмущением*. В нашем подходе фундаментальное различие между системой и внешней средой определяется уровнем наших знаний о них. Предполагается, что теория, задающая поведение системы, является полностью известной. Что касается внешней среды, то она является недетерминированной — для нее можно лишь оценить “сверху” множество возможных возмущений и известно, как определить результат применения конкретного возмущения к состоянию системы. Но невозможно для данного состояния точно предсказать, какое именно из возможных возмущений будет к нему применено. Например, если рассматривать банк как интерактивную дискретную динамическую систему, то можно видеть, что имеется некоторый полный список услуг, предоставляемых банком клиентам, при выполнении каждой из которых некоторая транзакция (действие) изменяет состояние банка. Но невозможно точно предсказать, какие именно заказы на услуги (возмущения) появятся в тот или иной момент.

В этих терминах эволюция дискретной системы может быть представлена как конечная или бесконечная траектория, состоящая из ее действий и возмущений внешней среды. Для простоты мы, не уменьшая общности рассмотрений, будем предполагать, что действия системы и возмущения регулярно чередуются. Имеется два вида регулярных траекторий:

$$(1) \quad \omega = \mathcal{E}_0 \xrightarrow{d} \mathcal{E}_1^* \vdash_S \mathcal{E}_1 \xrightarrow{d} \mathcal{E}_2^* \vdash_S \mathcal{E}_2 \dots$$

Такие траектории имеют тип *возмущение-действие* и называются *да-траекториями*.

$$(2) \quad \omega = \mathcal{E}_0 \vdash_S \mathcal{E}_1^* \xrightarrow{d} \mathcal{E}_1 \vdash_S \mathcal{E}_2^* \xrightarrow{d} \mathcal{E}_2 \dots$$

Эти траектории имеют тип *действие-возмущение* и называются ад-траекториями.

Рассматриваются три вида свойств живучести системы. В первой ситуации конечная ад-траектория начинается в недопустимом состоянии и система совместно с внешней средой могут в конце концов могут достигнуть допустимое состояние. Исходное состояние  $\mathcal{E}_0$  в этом случае называется *перспективным*. Во второй ситуации ад-траектория стартует в допустимом состоянии  $\mathcal{E}_0$  и первое действие выполняет система. Ее действия могут приводить к недопустимым состояниям, которые возмущениями внешней среды переводятся в допустимые, т.е.  $\mathcal{E}_i \models \Phi$  для всех  $i \geq 0$ . Такие траектории называются *стабильными*. В третьей ситуации да-траектория стартует в допустимом состоянии  $\mathcal{E}_0$ , к которому применяется возмущение внешней среды. Если при этом допустимость нарушается, то система должна восстановить ее в результате своего действия, т.е. и здесь  $\mathcal{E}_i \models \Phi$  для всех  $i \geq 0$ . Такие траектории называются *гомеостатичными*.

С технической точки зрения свойство перспективности близко связано с хорошо известной в искусственном интеллекте проблемой планирования, которую можно сформулировать, как проблему существования плана (траектории), приводящего систему в заданное состояние или в состояние, удовлетворяющее некоторым заранее заданным условиям. Содержательный смысл других определений следующий. Вдоль гомеостатичной траектории система способна регулярно восстанавливать свойство  $\Phi$  в ответ на разрушающие его возмущения внешней среды  $d$ . Дуально, вдоль стабильной траектории действия среды  $d$  компенсируют возможные разрушающие  $\Phi$  действия системы.

Траектории каждого из указанных типов (ад- или да-), начинающиеся в фиксированном состоянии  $\mathcal{E}_0$ , образуют деревья траекторий с корнем  $\mathcal{E}_0$ :  $T_{da}(\mathcal{E}_0)$  и  $T_{ad}(\mathcal{E}_0)$ , соответственно. В терминах этих деревьев можно формализовать многие естественные свойства живучести интерактивных динамических систем. Пусть  $Q_1, Q_2 \in \{\forall, \exists\}$ . Пусть  $S$  – система, действия которой представляются отношением  $\vdash_S$ , функционирующая в среде, возмущения которой представляются отношением  $\xrightarrow{d}$ , и пусть  $\mathcal{E}_0$  – это некоторое состояние системы. Скажем, что система  $S$  является  $Q_1Q_2$ -гомеостатичной в состоянии  $\mathcal{E}_0$ , если в дереве траекторий  $T_{da}(\mathcal{E}_0)$  имеется  $Q_1Q_2$ -поддерево с тем же корнем, в кото-

ром все ветви являются бесконечными гомеостатичными траекториями. Система является  $Q_1Q_2$ -стабильной в  $\mathcal{E}_0$ , если в дереве траекторий  $T_{ad}(\mathcal{E}_0)$  имеется  $Q_1Q_2$ -поддерево с тем же корнем, в котором все ветви являются бесконечными стабильными траекториями.

В параграфе 3.2 определяются дедуктивные базы данных с обновлениями (ДБД), основой для формализации которых выступают логические программы с обновлениями. Эти программы представляют из себя конечные множества предложений вида  $Head \leftarrow Body$ , где голова правила  $Head$  — это интенциональный атом (запрос) а тело  $Body$  — это последовательность литералов, элементарных обновлений баз данных вида  $insert(Fact)$  или  $delete(Fact)$  ( $Fact$  — это экстенциональный атом), присваиваний вида  $X := e$  ( $X$  — переменная,  $e$  — арифметическое выражение) и арифметических ограничений вида  $e_1 < e_2$ . На эти программы накладываются условия стратифицированности, обеспечивающие конечную завершаемость выполнения вызова каждой цели. Операционная семантика логических программ с обновлениями состоит в недетерминированном выборе предложения для разрешения очередной цели (как в обычном логическом программировании) и в последовательном слева-направо разрешении подцелей в теле выбранного предложения (как в Прологе). Вызов каждой цели  $\leftarrow a$  программы  $\mathcal{P}$  задает отношение перехода  $\overset{a}{\vdash}_{\mathcal{P}}$  на состояниях системы. Дедуктивная база данных с обновлениями (ДБД) — это система  $\mathcal{B} = \langle \mathcal{P} \cup \{ \leftarrow a_1, \dots, \leftarrow a_n \}, \Phi \rangle$ , которая включает логическую программу с обновлениями  $\mathcal{P}$ , фиксированное множество целей  $\{ \leftarrow a_1, \dots, \leftarrow a_n \}$ , реализующих действия ДБД, и ОЦ  $\Phi$ , которые задают конструктивное представление некоторого свойства состояний БД. ДБД  $\mathcal{B}$  задает на множестве состояний БД отношение переходов  $\vdash_{\mathcal{B}} = \bigcup_{i=1}^n \overset{a_i}{\vdash}_{\mathcal{P}}$ , являющееся объединением отношений для отдельных целей.

Мы классифицируем логические программы, налагая ограничения на вид их предложений. Логическая программа является  $\mathcal{P}$  *позитивной*, если в ней нет отрицаний, она *расширяющая*, если обновление  $delete$  не используется в ее предложениях.  $\mathcal{P}$  называется *базисной*, если все ее предложения базисные. Она является *плоской*, если все термы в литералах ее предложений являются либо переменными, либо константами.  $\mathcal{P}$  называется *ветвящейся*, если она не рекурсивна.  $\mathcal{P}$  называется *продукционной*, если она определяет один интенциональный

предикат  $q/0$  и все ее предложения являются *продукциями*, т.е. имеют вид  $Con_1 \wedge \dots \wedge Con_m \implies Act_1, \dots, Act_n$ , где каждое “условие”  $Con_i$  — это экстенциональный литерал, а каждое “действие”  $Act_j$  является элементарным обновлением.

Рассматриваемые ограничения целостности принадлежат одному из следующих трех классов.

$IC_0$  :  $\Phi$  сохраняется при расширениях состояний и все минимальные расширения состояния, удовлетворяющие  $\Phi$ , можно эффективно перечислить.

$IC_1$  : проблема  $\mathcal{E} \models \Phi$  принадлежит классу **P** (т.е. разрешима за полиномиальное время).

$IC_2$  : проблема  $\mathcal{E} \models \Phi$  принадлежит классу **PSPACE** (т.е. разрешима с полиномиальной памятью).

В частности, класс  $IC_0$  содержит ОЦ, задаваемые базисными ДНФ, логическими программами без отрицаний, а также некоторые монотонные свойства графов (например, связность). Хорошо известные функциональные зависимости и многие свойства размеров состояний БД (например, четность) принадлежат классу  $IC_1$ . Класс  $IC_2$  содержит ОЦ, выражаемые формулами логики 1-го порядка и др.

Возмущения внешней среды задаются парой  $\delta = (\Delta^+, \Delta^-)$  конечных множеств атомов.  $\mathcal{E} \xrightarrow{\delta} \mathcal{E}'$  означает, что  $D^+ = \mathcal{E}' \setminus \mathcal{E} \subseteq \Delta^+$  и  $D^- = \mathcal{E} \setminus \mathcal{E}' \subseteq \Delta^-$ . Таким образом,  $\delta$ -возмущение задает весьма простые теоретико-множественные границы для возможных множеств изменений состояний БД, которые не зависят от самих состояний.

В параграфе 3.3 анализируется сложность проблемы перспективности. Теорема 3.1 содержит оценки сложности этой проблемы для различных подклассов ДБД. Они варьируются от линейной сложности для расширяющих позитивных продукционных ДБД с ОЦ из  $IC_0$ , до полноты в классе **EXSPACE** — для плоских продукционных ДБД с ОЦ из  $IC_2$  и для плоских стратифицированных ДБД. Если в продукциях допустить сложные термы, то проблема становится неразрешимой.

Параграф 3.4 посвящен анализу различных вариантов проблемы стабильности. Для  $\exists\exists$ -стабильности (теорема 3.2) и  $\forall\forall$ -стабильности они аналогичны оценкам сложности для перспективности. Сложность проблем  $\forall\exists$ -стабильности и  $\exists\forall$ -стабильности несколько выше (теоремы 3.3 и 3.4). В частности, для плоских стратифицированных ДБД эти проблемы являются **EXPEXPTIME**-полными. Но и эти проблемы разрешимы

за полиномиальное время для расширяющих позитивных продукционных ДБД с ОЦ из  $IC_0$ .

Сложность проблемы  $\forall\exists$ -гомеостатичности для различных классов ДБД рассматривается в параграфе 3.5. Свойство  $\forall\exists$ -гомеостатичности представляется одним из самых интересных с практической точки зрения. Его выполнение для системы и некоторого ее (текущего) состояния означает, что какими бы ни были возмущения внешней среды, у системы будет достаточно средств, чтобы их компенсировать и на каждом шаге возвращаться в допустимое состояние. Теорема 3.5 устанавливает сложность  $\forall\exists$ -гомеостатичности для различных классов ДБД. Она показывает, что для монотонных ОЦ эта проблема разрешима за полиномиальное время, но при допущении их немонотонности сложность возрастает – проблема становится *EXPTIME*-трудной. Однако, если каждый факт, удаляемый внешней средой, может быть ею возвращен в базу данных, то  $\forall\exists$ -гомеостатичность можно распознать в детерминированной полиномиальной памяти (теорема 3.6).

Параграф 3.6 посвящен исследованию проблемы тотальной гомеостатичности, т.е. проверке гомеостатичности системы на всех допустимых состояниях. Рассматривается самый интересный вариант гомеостатичности —  $\forall\exists$ -гомеостатичность. Приводимые далее результаты показывают, что проверка  $\forall\exists$ -гомеостатичности на всех состояниях оказывается существенно более простой, чем на отдельном состоянии. Теорема 3.7 утверждает, что в классе продукционных ДБД, не использующих отрицаний в программах, и с ОЦ из класса  $IC_0$  проблема тотальной  $\forall\exists$ -гомеостатичности эффективно разрешима. Заметим, что обновления БД, при работе таких систем не обязательно монотонны, так как действия их продукций могут включать удаления фактов. Теорема 3.8 оценивает сложность тотальной  $\forall\exists$ -гомеостатичности для более широкого класса ветвящихся программ без отрицаний. Эта проблема разрешима за полиномиальное время при условии, что размер интенциональной сигнатуры системы ограничен некоторой константой. Без этого ограничения она является *co-NP*-полной. Технически самый сложный результат в этом параграфе теорема 3.10 относится к самому большому из рассматриваемых классов ДБД – классу  $DATALOG^u$  всех плоских стратифицированных программ. В ней доказано, что тотальная  $\forall\exists$ -гомеостатичность для этого класса является *NEXPTIME*-полной проблемой.

Глава 4 посвящена анализу поведения мультиагентных систем (МА-систем), в которых все агенты имеют одинаковую архитектуру, т.е. “равноправны” по возможностям получения и передачи информации. Для таких систем изучаются не только специальные свойства устойчивого поведения (как в предыдущей главе), но и произвольные свойства, выражимые средствами временных логик. В параграфе 4.1 перечисляются основные свойства интеллектуальных агентов и мультиагентных систем, сформулированные разными авторами. Приведены особенности архитектуры IMPAST, разработанной интернациональной группой в университете Мериленда,<sup>8</sup> на которой основана рассматриваемая нами модель МА-систем, и приводится точное описание этой модели. Кратко определены временные логики, используемые для описания свойств поведения МА-систем. Перечислены основные параметры, по которым классифицируются МА-системы. К ним относится детерминированный или недетерминированный способ выбора действий агентами системы, отсутствие или наличие переменных и отрицаний в логических программах, управляющих выбором допустимых действий, допустимость удалений из баз фактов, размерность предикатов, количество разных видов сообщений в системе, число агентов в системе. В параграфе 4.2 приводится небольшой пример, иллюстрирующий возможности рассматриваемых нами МА-систем. Это один из вариантов известного в литературе примера распределения ресурсов типа “Белоснежка и семь гномов”. Параграф 4.3 посвящен анализу поведения детерминированных МА-систем. Его первый пункт содержит описание алгоритма верификации детерминированных МА-систем, который служит основой для получения верхних оценок сложности. Во втором — представлены оценки сложности проблемы МА-BEHAVIOR для ряда классов детерминированных МА-систем (теорема 4.1) Они приведены в следующей таблице ???. Ее первые три столбца определяют классы рассматриваемых МА-систем. Столбцы “Баз.” и “Расш.” позволяют различить базисные и небазисные системы, а также расширяющие (без удалений) системы и системы с удалениями. В столбце “Параметры” указаны некоторые ограничения на параметры архитектуры МА-систем, приводящие к меньшей сложности верификации. В частности,  $N$  обозначает размер всего описания системы,  $m$  — число ее агентов,  $k$  — это максимальная размерность атомов действий и сообщений (в пропозициональном случае

---

<sup>8</sup>Subrahmanian V. S., Bonatti P., Dix J., et al., *Heterogeneous agent Systems*, MIT Press, 2000. 514p.

Таблица 1: Сложность верификации детерминированных МА-систем

Баз.	Расш.	Параметры	Логика	Сложность
Да	Да	позитивные	LTL	P
		$m^2 * r = O(\log N)$	LTL	P
		фикс. $m \geq 2$	LTL	PSPACE
		фикс. $r \geq 1$	LTL	PSPACE
	Нет		FLTL	PSPACE
Нет	Да	позит., фикс. $k$	LTL	P
		позитивные	LTL	EXPTIME
	Нет	фикс. $k$	FLTL	PSPACE
			FLTL	EXPSPACE

$k = 0$ ),  $r$  – это число различных сигналов (т.е. сообщений). В столбце "Логика" указан логический язык, в котором формулируются свойства поведения соответствующего класса МА-систем.

В параграфе 4.4 анализируется поведение недетерминированных МА-систем (теорема 4.2). Сложность верификации для них, естественно, оказывается выше и варьируется от NP-полноты до двойной экспоненты.

В параграфе 4.5 рассмотрено поведение вероятностных МА-систем. Источников неопределенности два: почтовая подсистема, которая может задерживать и даже терять сообщения, и неоднозначность результатов выполнения действий. В пунктах 2 и 3 определены синтаксис и семантика рассматриваемых систем и показано, как вероятностной МА-системе сопоставляется соответствующая цепь Маркова. В пункте 4 приведен полиномиальный алгоритм вычисления вероятностей перехода этой Марковской цепи. Это дало возможность в пункте 5 использовать известные результаты о сложности верификации Марковских цепей для поручения оценок сложности проблемы МА-BEHAVIOR для вероятностных МА-систем (теоремы 4.4, 4.5). В частности, существует алгоритм, который проверяет выполнимость FLTL-формулы  $F$  на состоянии  $S$  базисной вероятностной МАС  $\mathbf{A}$  в памяти, полиномиальной от  $|\mathbf{A}|$  и  $|F|$ . А вероятность  $p_{\mathbf{A}}(S_0, F)$  формулы  $F$  в состоянии  $S_0$  для такой же системы можно вычислить за время, экспоненциально зависящее от  $|\mathbf{A}|$  и  $|F|$ .

Глава 5 посвящена логическим программам с интервальными вероятностями. Формализмы логических программ с интервальными вероятностями, рассматриваемые в ней, были предложены в работах Р.Энга

и В.С.Субраманиана <sup>9</sup>. В параграфе 5.2 мы расширяем их, допуская дизъюнкции в головах правил. Дизъюнктивные логические программы с интервальными вероятностями (*dp*-программы) состоят из конечного числа предложений вида

$$G_1 : \nu_1 \vee \dots \vee G_k : \nu_k \longleftarrow F_1 : \mu_1 \wedge \dots \wedge F_n : \mu_n$$

где  $G_i$  и  $F_j$  — логические формулы, а  $\nu_i$  и  $\mu_j$  — подинтервалы отрезка  $[0, 1]$ . Содержательно, такое предложение означает, что если вероятность каждой формулы  $F_j$  лежит в интервале  $\mu_j$ , то вероятность хотя бы одной из формул  $G_i$  лежит в интервале  $\nu_i$ . Программы без дизъюнкций в головах назовем *p*-программами. Программа называется простой, если все формулы в ее предложениях являются атомами. В пункте 5.2.2 описана семантика “возможных миров” для этого класса программ, в которой вероятность формулы равна сумме вероятностей миров, в которых эта формула истинна. Моделями программы являются такие распределения вероятностей формул, на которых выполнены все ее предложения. В пункте 5.2.3 доказывается NP-полнота проблемы совместности (существования модели) и проблемы следования для *dp*-программ (здесь, в отличие от большинства NP-полных проблем, основную трудность представляет получение верхней оценки)(теоремы 5.1, 5.2). В указанных работах Р.Энга и В.С.Субраманиана для *p*-программ была определена семантика неподвижной точки. Эта семантика сопоставляла каждому атому программы (интерпретируемому, как случайное событие) интервал возможных вероятностных значений и утверждалось, что она совпадает с семантикой возможных миров. В пункте 5.2.4 устанавливается, что на самом деле семантика неподвижной точки недостаточна для описания всех моделей программ (предложения 5.3, 5.4, 5.5). В параграфе 5.3 получено полное описание множества точечных вероятностей как семантики возможных миров для разных классов вероятностных логических программ. В частности, мы решаем проблему точного описания семантики интервальных вероятностных программ как для класса *p*-программ, так и для более широкого класса *dp*-программ. В теоремах 5.3 и 5.4 и их следствиях приводится явное

---

<sup>9</sup>Ng R. and Subrahmanian V.S. Probabilistic Logic Programming. // Information and Computation, 101, 2, 1993. P.150–201. Ng R. and Subrahmanian V.S. A Semantical Framework for Supporting Subjective and Conditional Probabilities in Deductive Databases.// Journal of Automated Reasoning, 10, 2, 1993. P. 191–235.

описание множества точечных моделей, задаваемых  $p$ -программами и  $dp$ -программами как объединения конечного числа  $N$ -мерных параллелепипедов и  $N$ -мерных многогранников, соответственно ( $N$  - число различных фактов в программе). Описан алгоритм непосредственного построения множеств моделей. Предложен также алгоритм GenModT, использующий для построения множества моделей  $dp$ -программы ряд эвристик, позволяющих во многих случаях существенно уменьшить объем перебора. В параграфе 5.4 изучается семантика простых  $dp$ -программ. Устанавливаются необходимые и достаточные условия для того, чтобы теоретико-модельная семантика простой  $dp$ -программы совпадала с ее семантикой неподвижной точки (теорема 5.7).

Глава 6 посвящена двум теоретическим вопросам, связанным с языком функционального программирования РЕФАЛ (РЕкурсивных Функций АЛгоритмический язык), который был создан в 1966 году В.Ф. Турчиным в качестве метаязыка для описания семантики других языков. В результате появления достаточно эффективных реализаций на ЭВМ, он стал находить практическое применение в качестве языка программирования. Как язык практического программирования он ориентирован на символьные преобразования. В настоящее время основными диалектами языка являются РЕФАЛ-2 (1970-е), РЕФАЛ-5 (1985) и РЕФАЛ+ (1990), отличающиеся друг от друга деталями синтаксиса и набором дополнительных средств, расширяющих первоначальный вариант.

Первый из рассматриваемых вопросов связан с семантикой языка. Традиционно семантика Рефала определена в его описаниях операционно в терминах работы специальной Рефал-машины над основной памятью, называемой полем зрения. Вычисление Рефал-машины основано на правиле самой левой-самой внутренней замены (“вызов по значению”). Как мы выше отмечали, это правило не является правилом вычисления минимальной неподвижной точки, которая представляется естественной семантикой всякой системы рекурсивных определений. В параграфе 6.1 мы выделяем подкласс рекурсивных программ – программы с естественно расширенными базисными функциями, определяющие унарные неконстантные функции, и доказываем в теореме 6.1, что для этого подкласса вызов по значению является правилом вычисления минимальной неподвижной точки. В предложении 6.1 приведены примеры, показывающие, что при нарушении любого из условий теоремы 6.1 ее заключение будет неверно, т.е. все они являются необходимыми

для совпадения семантики вызова по значению с семантикой неподвижной точки. В пункте 6.2.3 определено естественное вложение Рефал-программ в класс рекурсивных определений, описанный в теореме 6.1 (предложение 6.3). Это позволяет заключить, что стратегия вычисления Рефала приводит к вычислению минимальной неподвижной точки (теорема 6.2). В пункте 6.2.4 показано, как, используя различные варианты индукции по неподвижной точке, этот результат можно применить для верификации (доказательства свойств) Рефал-программ.

Параграф 6.3 посвящен оценке сложности одного шага Рефал-машины. На каждом шаге Рефал-машина определяет имя вычисляемой функции  $F$ , ее аргумент  $w$ , а затем, используя определение функции в программе, вычисляет значение  $u = F(w)$  и заменяет им вызов  $F(w)$  в поле зрения. Если определение  $F$ ,  $w$  и замена выполняются весьма эффективно, то при вычислении значения  $F(w)$  Рефал-машине приходится (вообще говоря, не один раз) решать задачу отождествления некоторого образца  $v$  аргумента  $F$  с  $w$ . В этой задаче аргумент  $w$  представляет из себя объектное выражение — правильную скобочную последовательность, включающую символы некоторого конечного алфавита и числа, а в образец  $v$ , кроме того, могут входить переменные. Теорема 6.3 утверждает, что задача синтаксического отождествления, решаемая на каждом шаге работы Рефал-машины, является  $NP$ -полной. Нижняя оценка в этой теореме следует из представляющей самостоятельный интерес леммы 6.1, в которой доказано, что уже для 2-х символьного алфавита задача разрешимости уравнений в словах вида  $v = w$ , в которых переменные входят лишь в левую часть  $w$ , является  $NP$ -полной<sup>10</sup>. Более детальный анализ причин сложности задачи синтаксического отождествления позволил установить для нее верхнюю оценку времени  $O((3n^2/8)^m(k+n))$ , где  $k$  — длина образца,  $n$  — длина аргумента и  $m$  — число повторяющихся переменных в образце (предложение 6.3). В частности, если ни одна переменная не повторяется, то это дает линейную оценку сложности, а при малом их числе сложность полиномиальна.

В предыдущих главах было показано, что многие алгоритмические проблемы требуют для своего решения экспоненциального или сверхэкспоненциального времени и, следовательно, в общем случае практически

---

<sup>10</sup>Работа автора [?] с этим результатом опубликована в 1983г. При подготовке диссертации я обнаружил ссылку на более раннюю работу Энглин (Angluin) 1980г., посвященную вопросам индуктивного вывода, в которой получен аналогичный результат.

неразрешимы. Для их эффективного решения использовались два известных подхода:

- 1) уточнение формулировки проблемы, сокращение ее области определения, выделение легко разрешимых подпроблем;
- 2) использование эвристических алгоритмов, которые обеспечивают достаточно эффективное и достаточно точное, в том или ином смысле, решение проблемы.

В главе 7 рассматривается еще один подход к эффективному решению трудных проблем — *аппроксимируемость на начальных отрезках*. Он основан на следующем соображении: если для некоторой проблемы нет алгоритма, который эффективно решал бы ее на всех аргументах, то можно попытаться аппроксимировать ее просто решаемыми проблемами, решения которых совпадают с решениями исходной проблемы для достаточно большого начального отрезка аргументов. При этом оценка практической возможности такой аппроксимации зависит от размера аппроксимирующих алгоритмов как функции от длины отрезков, на которых они дают верные результаты. Так, очевидный “табличный” алгоритм, который можно предложить для быстрого разрешения начального отрезка любой проблемы, имеет экспоненциальный размер от длины аргументов и поэтому не может быть признан эффективным. С другой стороны, существуют сколь угодно сложные проблемы, для которых объем программ, распознающих за полиномиальное время их начальные отрезки, является медленно растущей функцией от длин отрезков ([?]). Последовательность длин программ, эффективно распознающих начальные отрезки разрешимой проблемы, образует сложностной спектр соответствующей проблемы. Основное внимание в главе 7 уделено аппроксимации “естественных” проблем, полных или трудных в различных сложностных классах. В параграфе 7.2 фиксируются свойства используемой нумерации машин Тьюринга (МТ) и определена сложность  $M^f(A^x)$  аппроксимации начального отрезка длины  $x$  проблемы  $A$ , как длина самой короткой программы МТ, распознающей  $A(y)$  при  $y \leq x$  за время  $\leq f(y)$ . В параграфе 7.3 устанавливается основной технический результат (теорема 7.1): для всякой границы времени  $f$  и неубывающей орф  $\alpha$ , принимающей все натуральные значения, существует разрешимая проблема  $A$  для которой  $M^f(A^x)$  совпадает (по порядку) с  $\alpha(x)$ . Емкостная и временная сложности проблемы  $A$  оцениваются через значения  $\alpha$  и  $f$ . В качестве следствий устанавли-

ливаются наличие в различных сложностных классах трудно аппроксимируемых проблем. Результаты затем распространяются и на сложность недетерминированных аппроксимаций (теорема 7.2). Связь между аппроксимируемостью и сводимостями с ограниченной сложностью устанавливается в параграфе 7.4. Теорема 7.3 является формализацией следующего содержательного тезиса: если трудно аппроксимируемая проблема  $A$  сводится с небольшой сложностью (например, за полиномиальное время) к проблеме  $B$ , то и проблема  $B$  является трудно аппроксимируемой. Вместе с результатами параграфа 7.3 это позволяет в параграфе 7.5 получить нижние оценки сложности аппроксимации для проблем, которые являются полными и трудными в различных сложностных классах (теоремы 7.4, 7.5, 7.6). В параграфе 7.6 устанавливаются нижние оценки сложности недетерминированной аппроксимируемости (теорема 7.7) и в качестве следствий выводится сверхполиномиальная (экспоненциальная) сложность логических схем, реализующих начальные отрезки проблем, которые являются трудными для классов EXPTIME (EXPSPACE). Для некоторых разрешимых логических теорий устанавливаются нижние оценки объема аксиоматических систем, в которых все теоремы ограниченной длины имеют “короткие” доказательства (следствия 7.7.1 и 7.1.2). В частности, если в некоторой системе аксиом любая теорема арифметики Пресбургера длины  $\leq n$  имеет доказательство длины  $\leq 2^{\sqrt{n}}$ , то размер этой системы не меньше  $c^n$  для некоторой константы  $c > 1$ , не зависящей от  $n$ . Завершается параграф оценками сложности аппроксимации начальных отрезков “понятных”, “красивых” теорем трудных теорий (теорема 7.8). Здесь завышенные понятия понимаются, как формулы, имеющие небольшую сложность порождения (Колмогоровскую сложность), поскольку строки с большой Колмогоровской сложностью естественно считать “случайными”, “запутанными” и т.п. В параграфе 7.7 устанавливается связь между полиномиальной аппроксимируемостью и некоторыми свойствами плотности полных и трудных проблем (лемма 7.5). В качестве следствий из результатов предыдущих параграфов получены, а в некоторых случаях и усилены, результаты о плотности полных проблем и сводимости к “редким” множествам из работ П. Бермана (P. Bergman), Н. Линч (N. Lynch) и Р. Соловея (R. Solovay) (теорема 7.9, предложения 7.3, 7.4, 7.5).

Параграф 7.8 посвящен установлению связей между границами вычислительной сложности проблемы и ее аппроксимируемостью. Основ-

ной здесь является лемма 7.6 об ускорении вычислений проблем с достаточно медленно растущей сложностью аппроксимации. Теорема 7.10 использует эту лемму и позволяет определить по верхней границе  $h(x)$  вычислительной сложности проблемы  $A$  и нижней границе  $g(x)$  такую функцию  $r_{g,h}(x)$ , что для сложности аппроксимации  $f(x) \leq g(x)/|x|$  и для бесконечного числа значений  $x$  имеет место неравенство  $M_s^f(A^x) \geq \lfloor \log_2 r_{h,g}(x) \rfloor$ . Показано, как этот результат можно перенести на временную сложность и приведены примеры функций  $r_{g,h}(x)$  для различных конкретных  $h(x)$  и  $g(x)$ . Теорема 7.11 дает примеры проблем, для которых нижняя оценка сложности аппроксимации из теоремы 7.10 является почти оптимальной.

### Список публикаций автора по теме диссертации

- [1] Дехтярь М.И. О сложности аппроксимации рекурсивных множеств // *Electronische Informatik und Kibernetik*, Akademie Verlag, Berlin, V. 12, N 3, 1976. P. 115–122.
- [2] Дехтярь М.И. О полиномиальной аппроксимируемости и сводимости // Четвертая Всесоюзная конференция по математической логике, Кишнев, 1976. С.41.
- [3] Dekhtyar M.I. Complexity spectra of recursive sets and approximability of initial segments of complete problems // *Electronische Informatik und Kibernetik*, V.15, N 1/2, Akademie Verlag, Berlin, 1979. P.11–32.
- [4] Dekhtyar M.I. Bounds on computational complexity and approximability of recursive sets // 8-th Symp. MFCS, *Lecture Notes in Computer Science*, V. 74, Springer, 1979. P.277–283.
- [5] Дехтярь М.И. Об аппроксимируемости и ускоряемости вычислений рекурсивных множеств // Пятая Всесоюзная конференция по математической логике. Тезисы докладов. Новосибирск, 1979. С.40–41.
- [6] Дехтярь М.И. О сложности некоторых подтеорий сложных теорий // *Семиотика и информатика*, вып. 12, М.: ВИНТИ, 1979. С.144–147.
- [7] Дехтярь М.И. О сводимости к "редким" множествам и плотности NP-полных задач // *Автоматы, алгоритмы, языки. Сб. трудов. Калининский государственный университет, Калинин*, 1982. С.42–52.

- [8] Дехтярь М.И. Замечание о сложности задачи синтаксического отождествления для языка рекурсивных функций // Математическая логика, математическая лингвистика и теория алгоритмов. Сб. трудов. Калининский государственный университет, Калинин, 1983. С.27–31.
- [9] Дехтярь М.И. Сложность решения одного класса уравнений в словах // Седьмая Всесоюзная конференция по математической логике. Тезисы докладов. Новосибирск, 1984. С.58.
- [10] Дехтярь М.И. О семантике Рефал-программ // Сложностные проблемы математической логики. Сб. трудов. Калининский государственный университет, Калинин, 1985. С.36–41.
- [11] Дехтярь М.И. О семантике и доказательстве свойств Рефал-программ // Всесоюзная научная конф. Проблемы совершенствования, тестирования, верификации и отладки программ. Тезисы докл., т.1, Рига, 1986. С. 102–103.
- [12] Дехтярь М.И., Диковский А.Я. **Динамические дедуктивные базы данных // Известия РАН, Техническая кибернетика, N 5, 1994. С.55–66.** (личный вклад 6С.)
- [13] Dekhtyar M.I., Dikovsky A.Ja., On Stable Behavior of Dynamic Deductive Data Bases // Proc.of the 10th International Symp. on Logic Programming, Ithaca, USA. The MIT Press, 1994. P.677.
- [14] Dekhtyar M. I., Dikovsky A. Ja., Dynamic Deductive Data Bases with Steady Behavior // Proc. of the 12th International Conf. on Logic Programming, (L. Sterling Ed.), The MIT Press, 1995. P.183–197. (личный вклад 7С.)
- [15] Дехтярь М.И., Диковский А.Я. ЕЕ-стабильность и перспективность поведения динамических дедуктивных баз данных. ИПМ им. М.В.Келдыша РАН, Препринт N 116, 1995. 34 с. (личный вклад 17С.)
- [16] Dekhtyar M. I., Dikovsky A. Ja. Properties of steady behavior of dynamic deductive data Bases. Part I. ЕЕ-stability and Promise // Universite Paris XII , Technical Report 95-07, 1995. 26p. (личный вклад 13С.)

- [17] Dekhtyar M. I., Dikovsky A. Ja. Properties of steady behavior of dynamic deductive data bases. Part II. Homeostaticity. Universite Paris XII, Technical Report 96-09, 1996. 15p. (личный вклад 7С.)
- [18] **Дехтярь М.И., Диковский А.Я. Анализ поведения дискретных динамических систем средствами логического программирования // Программирование, N 3, 1996. С.3–16. (личный вклад 7С.)**
- [19] Dekhtyar M. I., Dikovsky A. Ja. On Homeostatic Behavior of Dynamic Deductive Data Bases // Proc. 2nd Int. A.P.Ershov Memorial Conference "Perspective of Systems Informatics", Lecture Notes in Computer Science, Springer, V. 1181, 1996. P.420–432. (личный вклад 6С.)
- [20] Dekhtyar M. I., Dikovsky A. Ja., Recognition of deductive data base stability // Logic. Foundation of Computer Science - 4th International Symposium LFCS'97, Yaroslavl, Lecture Notes in Computer Science, V. 1234, Springer, 1997. P.67–77. (личный вклад 6С.)
- [21] Dekhtyar M. I., Dikovsky A. Ja. Total homeostaticity and integrity constraints restorability // Proc. of the 14th International Conference on Logic Programming. The MIT Press, Cambridge, Massachusetts, London, England, 1997. P.241–255. (личный вклад 7С.)
- [22] Dekhtyar M., Dikovsky A., Spyratos N. On Conservative Enforced Updates // Proceedings of 4th International Conference, LPNMR'97. Dagstuhl Castle, Germany, Lecture Notes in Computer Science, V. 1265, Springer, 1997. P.244–257. (личный вклад 5С.)
- [23] **Дехтярь М.И., Диковский А.Я., Спиратос Н. Восстановление ограничений целостности за счет наименьших достаточных изменений // Программирование. N 2. 1998. С.27–37. (личный вклад 4С.)**
- [24] Dekhtyar M.I., Dikovsky A.Ja., Spiratos N. On Logically Justified Updates // Proc. of the 1998 Joint International Conf. and Symp. on Logic Programming. Manchester-1998. The MIT Press, 1998. P.250–264. (личный вклад 5С.)

- [25] M.I. Dekhtyar, A. Ja. Dikovsky, N. Spyrtatos. Incremental Expansion of Database Updates through Integrity Constraints // Proc of JFPLC'99, Lyon (France):Hermes Science Publications, 1999, P.189–204. (личный вклад 6С.)
- [26] Dekhtyar M., Dikovsky A., Dudakov S., Spyrtatos N. Monotone Expansion of Updates in Logical Databases // Proc. of 5th International Conference LPNMR'99. Lecture Notes in Artificial Intelligence, V. 1730, Springer, 1999. P.132–147. (личный вклад 6С.)
- [27] Dekhtyar M. I., Dikovsky A. Ja., Valiev M. K. Applying temporal logic to analysis of behavior of cooperating logic programs // Proc. of 3rd Int. A.P.Ershov Memorial Conf. PSI'99, Novosibirsk, July, 1999. Lecture Notes in Computer Science, V. 1755, Springer, 2000. P.228–234. (личный вклад 2С.)
- [28] Dekhtyar M., Dikovsky A., Dudakov S., Spyrtatos N. Maximal Expansions of Database Updates // Foundations of Information and Knowledge Systems, FoIKS 2000. Lecture Notes in Computer Science, V. 1762, Springer, 2000, P.72–87. (личный вклад 4С.)
- [29] Dekhtyar M., Dikovsky A., Dudakov S. On Complexity of Updates Through Integrity Constraints // Proc. of the First Int. Conf. on Computational Logic (CL 2000), Lecture Notes in Artificial Intelligence, V. 1861, Springer, 2000. P.867–881. (личный вклад 5С.)
- [30] **Dekhtyar M., Dikovsky A., Dudakov S., Spyrtatos N. Maximal State Independent Approximations to Minimal Real Change // Annals of Mathematics and Artificial Intelligence. V. 33, Kluwer Academic Publishers, 2001. P.157–204. (личный вклад 10С.)**
- [31] Валиев М.К., Дехтярь М.И., Диковский А.Я. О сложности поведения систем взаимодействующих агентов// Труды конференции, посвященной 90-летию со дня рождения Алексея Андреевича Ляпунова, Новосибирск, Академгородок, 8 - 11 октября 2001г., С.18-28. (личный вклад 4С.)
- [32] Dekhtyar M. I., Dikovsky A. Ja., Valiev M. K. On Behavior of Interacting Agents // Research report N 02.01, March 2002, IRIN, Universite de Nantes, 36 p. (личный вклад 12С.)

- [33] Dekhtyar M. I., Dikovskiy A. Ja., Valiev M. K. Complexity of Multi-Agent Systems Behavior // Proc. of 8th European Conf. on Logics in Artificial Intelligence, JELIA 2002. Cosenza, Italy, Lecture Notes in Artificial Intelligence, V. 2424, Springer, 2002. P.125–136. (личный вклад 4С.)
- [34] Dekhtyar M. I., Dikovskiy A. Ja. Valiev M. K. Checking Multi-Agent Systems Behavior Properties // Proc. of IEEE Internat. Conf. on Artificial Intelligence Systems. (ICAIS2002). September 5-10, 2002, Divnomorskoe, Russia, 2002. P.308–313. (личный вклад 2С.)
- [35] Дехтярь М.И. Обновления баз данных при динамических ограничениях целостности // "Системная информатика", N 8, Сб. научных трудов под ред. И.В.Поттосина и Ф.Г.Марчука. Новосибирск : Наука. 2002. С.72–142. (личный вклад 4С.)
- [36] Валиев М.К., Дехтярь М.И., Диковский А.Я. О сложности верификации систем взаимодействующих интеллектуальных агентов // Труды международных конференций "Интеллектуальные системы"(IEEE AIS'03) и "Интеллектуальный САПР"( CAD2003), Дивноморское, 2-8 сентября 2003, М.: Физматлит, 2003, С. 475–481. (личный вклад 2С.)
- [37] Валиев М.К., Дехтярь М.И., Диковский А.Я. О сложности верификации динамических свойств многоагентных систем // Труды Первой Всероссийской научной конференции "Методы и средства обработки информации", Московский гос. университет, 2003, С. 329–335. (личный вклад 2С.)
- [38] Dekhtyar M., Dikovskiy A., and Valiev M. On feasible cases of checking multi-agent Systems Behavior // **Theoretical Computer Science, Elsevier Science, V. 303, N. 1, 2003. P.63–81.** (личный вклад 6С.)
- [39] Dekhtyar A., Dekhtyar M.I. Possible Worlds Semantics for Probabilistic Logic Programs // Proc., International Conference on Logic Programming (ICLP)'2004, Lecture Notes in Computer Science, V. 3132, Springer, 2004. P.137–148. (личный вклад 6С.)
- [40] Дехтярь А.М., Дехтярь М.И. О семантике простых логических программ с интервальными вероятностями // Труды Девятой наци-

ональной конференции по искусственному интеллекту с международным участием (КИИ-2004), Том 1, Тверь, М.: Физматлит, 2004. С.254–262. (личный вклад 4С.)

- [41] Валиев М.К., Дехтярь М.И., Диковский А.Я., Китаев Е. Л., Скороходов А.П. Верификация динамических свойств многоагентных систем // Труды III-го Межд. научно-практического семинара "Интегрированные модели и мягкие вычисления в искусственном интеллекте", М.: Физматлит, 2005. С.69–75. (личный вклад 2С.)
- [42] Dekhtyar A., Dekhtyar M.I. Revisiting the Semantics of Interval Probabilistic Logic Programs // Proc. 8th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'05), Lecture Notes in Artificial Intelligence, V. 3662, Springer, 2005. P.330–342. (личный вклад 6С.)
- [43] Dekhtyar M.I., Dikovsky A.Ja., Valiev M.K. **On complexity of verification of interacting agent's behavior** // **Annals of Pure and Applied Logic, V. 141, Elsevier, 2006. P.336–362.** (личный вклад 9С.)
- [44] Валиев М.К., Дехтярь М.И., Диковский А.Я. О свойствах многоагентных систем с вероятностными каналами связи // Труды IV-ой Межд. научно-практической конференции "Интегрированные модели и мягкие вычисления в искусственном интеллекте", М.: Физматлит, Коломна, 2007. С.119–126. (личный вклад 3С.)
- [45] Dekhtyar M.I., Dikovsky A.Ja., Valiev M.K. Temporal Verification of Probabilistic Multi-Agent Systems // Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday, Lecture Notes in Computer Science, V. 4800, Springer, 2008. P.256–265. (личный вклад 3С.)
- [46] Дехтярь А.М., Дехтярь М.И. О семантике дизъюнктивных логических программ с интервальными вероятностями // Труды XI Национальной конференции по искусственному интеллекту с международным участием. Дубна, М.: "LENAND", 2008. С.240–248. (личный вклад 4С.)
- [47] **М.К. Валиев, М.И. Дехтярь. Вероятностные мультиагентные системы: семантика и верификация // Вестник Твер-**

ского государственного университета, серия "Прикладная математика". 35 (95), 2008. С.9–22. (личный вклад 7С.)

[48] Dekhtyar A., Dekhtyar M.I. The Theory of Interval Probabilistic Logic Programs // Annals of Math. and Art. Intel., 2009. 32р. (принято в печать). (личный вклад 16С.) <http://www.springerlink.com/content/8n82m5637028489v/?p=5a24db81e09a4234907d44a4de6e066e&pi=18>

[49] Валиев М.К., Дехтярь М.И., Диковский А.Я. Системы агентов, управляемых логическими программами: сложность верификации // Программирование, 5, 2009. 22с. (принято в печать) (личный вклад 8С.)





