На правах рукописи

НОВОСЕЛЬЦЕВ Виталий Борисович

ФОРМАЛЬНАЯ ТЕОРИЯ СТРУКТУРНЫХ МОДЕЛЕЙ ОПИСАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ И МЕТОДЫ УСТАНОВЛЕНИЯ ВЫВОДИМОСТИ

Специальность 05.13.01 – «*Системный анализ, управление и обработка информации* (*в отраслях информатики, вычислительной техники и автоматизации*)»

АВТОРЕФЕРАТ

диссертации на соискание ученой степени доктора физико-математических наук

Томск – 2006

Работа выполнена на кафедре оптимизации систем управления Томского политехнического университета

Официальные оппоненты: ***ЖДАНОВ Александр Аркадьевич***, доктор физико-математических наук, профессор

МАТРОСОВА Анжела Юрьевна, доктор технических наук, профессор

ТИМЧЕНКО Сергей Викторович, доктор физико-математических наук, профессор

Ведущая организация: ***Институт математики им.С.Л. Соболева*** *Сибирского отделения Российской Академии Наук*

Защита состоится «21» декабря 2006 г. в **10**^{30} часов на заседании диссертационного совета **Д212.267.12** в Томском государственном университете по адресу:
634050, Томск, пр. Ленина, 36

С диссертацией можно ознакомиться в научной библиотеке Томского государственного университета

Автореферат разослан «_____» _____ 2006 г.

Ученый секретарь диссертационного совета,
д.т.н., профессор

В.И.Смагин

1. Общая характеристика диссертации

Актуальность темы. В диссертации рассматриваются проблемы автоматического установления выводимости предложений в специальных формальных теориях. Важным моментом при этом является использование *обратных методов* (в стиле С.Ю.Маслова[1]) вывода в рамках достаточно выразительных, но разрешимых исчислений. Разработанная методология применима при реализации программных систем, моделирующих процессы проведения доказательных рассуждений человеком, традиционно называемых системами управления знаниями. В работе предложен целостный подход, обеспечивающий построение строгих, адекватных и, в то же время, полных и эффективных формализмов работы со знаниями для широкого круга прикладных областей. При проведении исследований существенно используются естественно определяемые в пространствах вывода отношения порядка, что, как правило, обеспечивает высокую эффективность используемых стратегий и алгоритмов.

Одной из важнейших проблем современных *компьютерных наук* (*computer science*) является разработка инструментальных средств, максимально сокращающих объем специальных навыков, необходимых при использовании информационных технологий. Отметим три наиболее значимых, на наш взгляд, области, для которых «интеллектуализация» является чрезвычайно важной. По степени вовлеченности пользователей на *первое* место, несомненно, следует поставить *Всемирную паутину* (*World Wide Web*) с ее гигантскими информационными ресурсами и сопутствующими проблемами менеджмента данных, с быстрорастущим количеством сетевых сервисов и недостатками механизмов реализации этих служб, с необходимостью управления сложной структурой Паутины и т.д. – Все это вызывает растущие потребности в интеллектуальных средствах планирования, поиска и управления для Паутины.

Второй можно упомянуть нишу прикладных информационных комплексов. Имеются в виду системы, использующие нетривиальный логико-математический аппарат для получения результата (расчета, прогноза, классификации), необходимого конечному пользователю. Наконец, *третья* из важных областей практической деятельности относится, главным образом, к науке или инженерии. Речь идет о (*возможном*) желании прикладного пользователя *самостоятельно* и *за короткий срок* создать необходимый ему программный продукт. – Здесь одним из наиболее перспективных подходов является разработка систем с *автоматическим синтезом программ*.

---

[1] ***Маслов С.Ю.*** *Обратный метод установления выводимости в классическом исчислении предикатов // ДАН СССР. – 1964. – вып.159, №1. – С. 17-20*

Наиболее распространенными при разработке программных оболочек управления и обработки знаний являются дедуктивные системы. Такие системы моделируют проведение формально обоснованных, доказательных рассуждений, характерных для человека. Проблема построения доказательства традиционно понимается как проблема поиска вывода формулы логико-математического языка в соответствующем исчислении. Имеются многочисленные работы, связанные как со стандартными логиками (их разрешимыми фрагментами), так и рассматривающие неклассические исчисления (интуиционистские, нечеткие, дескриптивные и т.д.). В то же время ощущается явная нехватка работ *комплексного* характера, предлагающих формализации отмеченной проблемы для *классов* (пусть и родственных) различных подходов к представлению и обработке знаний. Кроме того, далеко не всегда надлежащее внимание оказывается вопросам алгоритмической эффективности.

Наконец, еще одно важное соображение. Большая часть работ, связанных с проблематикой установления выводимости, ориентированы на так называемые *целеориентированные* методы доказательства. – В них построение вывода осуществляется от целевого утверждения к аксиомам, что имеет как положительные, так и отрицательные стороны. *Обратные* по отношению к целеориентированным методы естественнее и не нуждаются в применении сложных механизмов возврата при неудаче (бектрекинге), хотя и весьма чувствительны к объему пространства вывода. Вопросы повышения эффективности обратных методов исследованы достаточно мало, что также подтверждает полезность приведенных в диссертации результатов.

Цель работы. Диссертация посвящена созданию комплексного подхода, обеспечивающего создание строгих, достаточно выразительных и, в то же время, эффективных механизмов логического вывода на базе обратного метода для классов формальных теорий. Существенным при построении процедур распознавания, анализа и планирования вывода является учет и использование порождаемых формализмами отношений порядка, что, как правило, обеспечивает высокую эффективность.

При реализации объявленной программы мы опираемся на известные фундаментальные и частные результаты, ссылки на которые всегда явно отмечены. Проведенные исследования, традиционно относят к автоматическому доказательству теорем на базе дедуктивного подхода. Предложенные логические системы детально исследуются – для них доказываются теоремы полноты и строго формулируются стратегии и алгоритмы построения вывода. Для алгоритмов также показывается корректность, и приводятся сложностные оценки.

Методы исследования. В работе использованы положения математической логики, теории множеств, теории алгебраических систем, теории алгоритмов, теории графов.

Научная новизна полученных результатов определяется следующими положениями:

- построена формальная теория структурных рекурсивных импликативных моделей ***SR***, для нее показаны свойства эффективной распознаваемости и эффективной разрешимости, а также приведены варианты прикладного использования;
- для исчисления ***SR*** сформулирован базовый обратный алгоритм установления выводимости предложения, показана корректность алгоритма и определены его сложностные характеристики;
- предложена бесповторная стратегия вывода, учитывающая структуру модели и естественный порядок на атрибутах;
- сформулирован и обоснован подход к установлению выводимости формул для систем классов ***KT*** и ***KT45*** (***S5***) (модальных логик знания и автоэпистемической);
- исследованы алгоритмические аспекты проблемы установления выводимости для предложенного подхода, на основании чего предложены строго обоснованные стратегии сокращения пространства поиска вывода.

Практическая значимость. Все теоретические построения, приведенные в диссертации, формально строго обоснованы. Полученные результаты частично использовались в ряде завершенных прикладных проектах, связанных с обработкой знаний, и применяются в руководимых автором текущих исследованиях и разработках, в том числе, аспирантских. В перспективе предложенный подход может быть использован для исследования экономических моделей с конкурентным взаимодействием, децентрализованных комплексов управления, распределенных информационных систем различного назначения.

Апробация работы. Результаты работы докладывались в профильных семинарах ВМиК МГУ, ММФ СПбГУ, ФПМиК и ММФ ТГУ, НПП «Элеси», АВТФ ТПУ, а также на ряде научных форумов различного уровня, среди которых можно отметить следующие:

- II Всероссийская конференция по прикладной логике – Новосибирск, ИМ СО РАН, 1988,

- IX Международная школа ПП-91 «Программное обеспечение управления и ИИ» – Иркутск, ИрВЦ СО РАН, 1991,
- Международная конференция «Параллельные вычисления-91» – Киев, ИК АН Украины, 1991,
- Международная конференция «Логика и семантическое программирование-1992» – Новосибирск, ИМ СО РАН, 1992,
- 5th International Symp. on Applied Logic – Tallinn, 1996,
- Международная конференция «Всесибирские чтения по математике и механике. Томск 1997» – Томск, ТГУ, 1997,
- 8th Korean-Russian International Symposium on Science and Technology – KORUS 2004, Томск, ТПУ,
- 8th International Conference on APPLIED MATHEMATICS Puerto De La Cruz,Tenerife, Spain, 2005,
- 9th Korean-Russian International Symposium on Science and Technology – KORUS 2005, Новосибирск, НГТУ.

Результаты, выносимые на защиту:

1. *Формальная теория структурных рекурсивных импликативных моделей* ***SR*** *для описания прикладных областей.*
2. *Эффективные обратные стратегии и алгоритмы установления выводимости формул в теории* ***SR.***
3. *Формальный подход к установлению выводимости формул для модальных систем классов* ***KT*** *и* ***KT45*** (***S5***) *с использованием обратного метода.*
4. *Строго обоснованные стратегии сокращения пространства поиска вывода в модальных системах класса* ***KT****.*

Объем и структура диссертации. Диссертация включает введение, пять глав, заключение и список литературы; общий объем составляет 207 страниц, включая рисунки.

2. СОДЕРЖАНИЕ ДИССЕРТАЦИИ

Во введении обоснована актуальность реализуемой тематики, определены предмет и цели исследования, приведены содержательные формулировки сопутствующих задач, а также указаны возможные варианты использования полученных результатов.

Глава 1

Здесь систематизированы распространенные подходы и методы, применяемые в области обработки знаний. Приведен аналитический обзор работ из наиболее значимых (по мнению автора) разделов указанной области, включающих, главным образом, дедуктивные системы и механизмы построения заключений в них.

Большинство используемых в настоящее время алгоритмов поиска вывода основано на теореме Эрбрана, которая формулирует необходимые и достаточные условия выводимости формулы для классической логики. При использовании этой теоремы происходит, как правило, порождение большого числа избыточных дизъюнктов. Для сокращения количества этих избыточностей используются разнообразные подходы. Наиболее известным и получившим широкое распространение является метод резолюций[1], предложенный Дж. Робинсоном. Этот метод относят к так называемым *прямым* или *целеориентированным* методам, используя его, доказательство строят, продвигаясь от цели к аксиомам.

Практически, одновременно с методом резолюций и независимо от Дж. Робинсона ленинградским логиком С.Ю. Масловым был предложен *обратный* метод поиска вывода – обратный по отношению к целеориентированным. Метод Маслова дает возможность, направленно, опираясь на внутренние свойства доказываемой формулы, ограничить перебор, возникающий при непосредственном применении теоремы Эрбрана, и получить вывод цели, двигаясь «от аксиом».

Существует целый ряд работ, связанных с проблематикой менеджмента знаний и использующих иные парадигмы и методы манипулирования когнитивными структурами с использованием логических построений. – В первой главе обсуждаются наиболее популярные из них, определяются преимущества и подчеркиваются недостатки.

---

[1] ***Robinson J.A.*** *On automatic deduction // Rice Univ. Studies. – 1964. – V.50. – №1*

Глава 2

В этой главе с учетом проведенного анализа предпринимается попытка построения формализма, призванного обеспечить возможность создания строгих и адекватных моделей для широкого круга предметных областей (**ПО**). Для достижения этой цели отталкиваются от хорошо разработанной теории вычислительных моделей[1] и используют современные подходы и достижения. Определяется язык теории структурных (С-) моделей, выделяются существенные для дальнейшего изложения классы моделей, определяется положение выделенных классов в существующей иерархии и строго задается понятие интерпретации модели.

Язык теории. Основные определения

Фиксируется сигнатура.

Определение 2.1. Четверку $\Sigma = (\mathbb{A},\mathbb{F},\mathbb{P},\mathbb{D})$, где $\mathbb{A}$, $\mathbb{F}$, $\mathbb{P}$ и $\mathbb{D}$ – не более чем счетные множества (элементарных) имен объектов, функциональных символов, селекторных символов и имен схем, соответственно, назовем сигнатурой. Считается, что выделено непустое конечное подмножество $\mathbb{E}\subset\mathbb{D}$ имен примитивных или первичных схем[2].

Элементы множества $\mathbb{A}$ называются *именами объектов* или *объектами,* если это не приводит к двусмысленности. Связь объекта a со схемой S отражается записью $S(a)$. Если объект a связан со схемой S и $S\in\mathbb{E}$, стандартная запись $S(a)$ заменяется записью a^S либо просто a, когда ссылка на схему не важна или очевидна из контекста. – Истинностное значение синонимов a^S и $S(a)$ будем определять следующим образом: $a^S=\boldsymbol{И}$, тогда и только тогда, когда $a|_{\mathbf{I}}\in S|_{\mathbf{I}}$, где $a|_{\mathbf{I}}$ и $S|_{\mathbf{I}}$ обозначают реализации (на интерпретации **I**) величины и схемы соответственно.

Определение 2.2. Выражение вида

$$f{:}a_1,\ldots,a_n \to a_0, \tag{2.1}$$

где a_i, $i=0,\ldots,n$ – имена объектов, называется функциональной связью (**ФС**). В (2.1) f – это имя **ФС**, a_i – аргументы ($i=1,\ldots,n$), a_0 – результат **ФС**.

---

[1] ***Тыугу Э.Х.*** *Решение задач на вычислительных моделях // ЖВМиМФ. – 1970. – Т.10. – №3*

[2] *Фиксация множества* $\mathbb{E}\subset\mathbb{D}$ *для конкретной* **ПО** *не является статической. При необходимости, можно сделать доопределение «вниз» и детализировать рассматриваемую* **ПО** *соответствующей модификацией множества* $\mathbb{E}$. *Требование конечности множества* $\mathbb{E}$ *при этом не нарушается.*

Неформально **ФС** трактуется как возможность вычисления значения атрибута a_0 по набору значений атрибутов $a_1,\ldots,a_n$ применением реализации отображения f.

Имена объектов формируются из элементов множества $\mathbb{A}$.

Определение 2.3. Пусть $a\in\mathbb{A}$, α – имя объекта, тогда a, $\alpha.a$ и $a.\alpha$ также являются именами объектов. В записи вида $\alpha.a$ α называется префиксом. *Длина* произвольного имени α определяется числом вхождений элементов множества $\mathbb{A}$ (с учетом повторений), из которых сформировано имя.

Одним из базовых является понятие (непервичной) схемы объекта:

Определение 2.4. Схема S объекта r определяется выражением вида

$$\begin{aligned} S(r) = r.(&S_{01}(a_{01}),\ldots, S_{0N0}(a_{0N0}),\\ &\textit{if}\quad p_1 \supset S_{11}(a_{11}),\ldots,S_{1N1}(a_{1N1})\ \square\ldots\square \\ &\qquad p_k \supset S_{k1}(a_{k1}),\ldots,S_{kNk}(a_{kNk})\ \textit{fi}\ \mid\ \textit{filter}\ ), \end{aligned} \tag{2.2}$$

где $S\in\mathbb{D}\backslash\mathbb{E}$, $r\in\mathbb{A}$. Для всех возможных значений индексов i, j, $S_{ij}\in\mathbb{D}$ – *собственные подсхемы* схемы S, $a_{ij}\in\mathbb{A}$ – ее *собственные величины*, $p_i\in\mathbb{P}$ – (возможно, параметризованные) выбирающие *селекторы*. В правой части (2.2) r называется *префиксом* схемы, участок до вертикальной черты – *заголовком*, фрагмент *if...fi* – *вариантной* его *частью*, а остальная часть заголовка – *постоянной частью*. Символ «$\square$» заимствован из нотации «охраняемых команд» Дейкстры и, в некотором смысле, соответствует традиционному «*else_if*». Иероглиф *filter* скрывает список *собственных* **ФС** схемы S. Считается, что префикс r может «проноситься» в скобочный фрагмент, так что $r.(S(a),\ldots)=_{def}(r.S(a),\ldots)=_{def}(S(r.a),\ldots)$. Аналогично префиксируются имена селекторов, отображений и атрибутов **ФС** из набора *filter*.

Селекторы на интерпретации **I** получают истинностные (шкальные) значения и образуют полную систему, т.е. $\forall i,j,\ i\neq j,\ p_i|_{\mathbf{I}}\&p_j|_{\mathbf{I}}=\textit{Л}$ и $p_1|_{\mathbf{I}}\vee\ldots\vee p_k|_{\mathbf{I}}=\textit{И}$. Синтаксическая конструкция (2.2) на интерпретации **I** определяет размеченное отношение в смысле Дейкстры.

Определение 2.5. Пусть $S(r)=(\ldots S_{ij}(r.a_{ij})\ldots)$ – схема. Если $S_{ij}\in\mathbb{E}$, то $r.a_{ij}$ – имя величины схемы S. Если $S_{ij}\in\mathbb{D}\backslash\mathbb{E}$ и α – имя величины схемы S_{ij}, то $r.a_{ij}.\alpha$ – имя величины схемы S. Ниже имена величин называются атрибутами.

Определение 2.6. Пусть S – схема. **ФС** $f{:}\alpha_1,\ldots,\alpha_n\rightarrow\alpha_0$ из *filter* называется допустимой для схемы S, если и только если $\alpha_0,\alpha_1,\ldots,\alpha_n$ – атрибуты схемы S. Схема называется синтаксически правильной, если *filter* содержит только допустимые **ФС**.

В дальнейшем считается, что **ФС**, содержащиеся в наборе *filter* схемы S, отвечают следующим требованиям:

- **ФС** содержит по крайней мере один собственный атрибут схемы;

- в наборе *filter* нет **ФС**, в которые вовлечены атрибуты из разных альтернативных ветвей схемы ***S*** или атрибуты из альтернативных частей её подсхемы;
- длина любого атрибута, связанного **ФС**, не превышает двух.

Определение 2.7. Факт наличия условия p у атрибута α отмечается записью α/p, а соответствующее выражение называется *у-атрибутом*.

Определим понятие структурной импликативной модели.

Определение 2.8. Структурной (С-) моделью называется конечная совокупность схем $M=(S_1,\ldots,S_m)$, где каждая S_i, $i=1,\ldots,n$, является элементарной или задана в соответствие с определением 2.4.

Определение 2.9. С-модель M является (синтаксически) замкнутой, если и только если для каждого её элемента $S_i \in M$ схемы, встречающиеся в определении S_i, являются элементарными либо определены в описании M.

Описывая С-модель для предметной области, мы строим специальную (прикладную) теорию, в которой можно решать различные алгоритмические проблемы. Моделируемые **ПО** не обязательно носят «расчетный» характер – необходимым требованием, как отмечалось выше, является *реализуемость* **ФС**. Постановка задачи в С-модели имеет непроцедурный характер, – в ней указываются лишь исходные и искомые атрибуты некоторой схемы.

Определение 2.10. Задачей в С-модели M называется тройка $T=(A_0,X_0,S)$, где A_0 и X_0 – наборы имён, соответственно, исходных и искомых величин, а S – схема С-модели M**,** в которой определены эти имена.

При исследовании свойств С-моделей и алгоритмов синтеза программ в них используется понятие развёртки схемы S. Оно определяется через развёртку схемы на атрибуте.

Определение 2.11. Пусть $S(r)=r.(\ldots S_{ij}(a_{ij}),\ldots|filter)$ – схема, и $S_{ij} \notin \mathbb{E}$. Развёрткой схемы S на атрибуте a_{ij} называется выражение, получающееся в результате (*а*) подстановки в заголовок S на место $S_{ij}(a_{ij})$ заголовка схемы S_{ij} и (*б*) присоединения к набору *filter* схемы S всех **ФС** схемы S_{ij}.

Все добавленные в схему S в результате такого действия атрибуты, имена селекторов и отображений модифицируются префиксом $r.a_{ij}$.

Определение 2.12. Под полной развёрткой S С-модели понимается объект, полученный из схемы S, у которого в результате последовательности развёрток на атрибутах в заголовке остаются только атрибуты, связанные с элементарными схемами.

Полная развёртка, вообще говоря, не является схемой в смысле определения 2.4, однако понятие схемы нетрудно переопределить с тем, чтобы оно оставалось корректным для развёртки на атрибуте и полной развёртки. Для этого необходимо в выражении (2.2) разрешить появление

нескольких *if*...*fi*-участков и потребовать, чтобы единственным видом их пересечения было строгое вложение.

При построении С-модели допускаются возможность рекурсивных определений схем. Ясно, что полная развёртка рекурсивной схемы не определена – процесс её построения не завершается. Во *втором* параграфе показывается, что некоторые естественные ограничения позволяют выделить класс С-моделей, допускающий рекурсивные схемы, в котором проблема доказательства *эффективно* решается, т.е. полная развертка схемы не требуется. Выделенный подкласс вкладывается в канонический класс детерминированных моделей Диковского[1], но является существенно более привлекательным, чем родительский, в смысле алгоритмических характеристик.

Определение 2.13. Схему S вида (2.2) называют детерминированной, если она не содержит вариантной части либо её вариантная часть имеет двухальтернативную форму $if\ p \supset ...;\ ...\ fi$.

Здесь и ниже, если не оговаривается противное, полагают, что совокупность схем, составляющих С-модель, является упорядоченной.

Определение 2.14. С-модель $M=(S_1,...,S_m)$ называется нерекурсивной детерминированной (НДС-) моделью, если выполнены следующие условия:

- M – синтаксически замкнута;
- $S_1,...,S_m$ – детерминированные схемы;
- любая схема S_i модели определяется только через элементарные схемы либо через схемы, которые предшествуют S_i в модели M.

Определение 2.15. Детерминированная схема, у которой для любых двух **ФС**, совпадающих по правой части (результату), аргументы принадлежат *разным* альтернативным ветвям, называется строго детерминированной.

Теорема 2.1 подтверждает включение класса НДС в класс Диковского.

Теорема 2.1. Полная развёртка любой схемы S НДС-модели M определяет модель Диковского, если M составлена только из строго детерминированных схем.

Замечание. А.Я. Диковским показано, что при выборе семантики, в которой не используется внешняя управляющая память для задания условий в детерминированных уравнениях, проблема синтеза в Д-вычислительных моделях является ***NP***-полной. В НДС-моделях алгоритмы, решающие проблему синтеза, обладают полиномиальными характеристиками. Теорема

---

[1] ***Диковский А.Я.*** *Детерминированные вычислительные модели // Техническая кибернетика. – 1984. – Т.33. – №5. – С. 68-74.*

2.1 выделяет в классе моделей Диковского «хороший» в алгоритмическом плане подкласс моделей.

В третьем параграфе вводится понятие рекурсивных детерминированных (РД) С-моделей и доказывается свойство эффективной распознаваемости.

Определение 2.16. С-модель $M=(S_1,\ldots,S_m)$ называется рекурсивной детерминированной (РДС) моделью, если выполнены следующие требования:

1) M – синтаксически замкнута;
2) $S_1, \ldots, S_m$ – детерминированные схемы;
3) при построении развёртки любой схемы S_i модели M выражения вида $\alpha.S_i(\ldots)$, где α – различные префиксы, появляются только в одной из двух альтернативных ветвей S_i. При этом выражения $\alpha.S_i(\ldots)$ определяют рекурсивные вхождения исходной схемы, ветвь, в которой имеются рекурсивные вхождения, называется рекурсивной ветвью, а сама S_i – рекурсивной схемой.

Конструктивность определения 2.16 обеспечивается теоремой.

Теорема 2.2. Принадлежность С-модели $M=(S_1,\ldots,S_m)$ классу РДС устанавливается конструктивно за конечное время, пропорциональное $O(|M|^4)=C\times m^4$.

Далее вводится отношение частичного порядка на атрибутах модели, а также понятия достижимости и допустимости атрибута. Достижимость атрибута определяется (*а*) наличием в развёртке функциональной связи, результатом которой является этот атрибут, (*б*) достижимостью всех аргументов такой **ФС** и (*в*) непротиворечивостью условий, при которых в процессе планирования достигнуты аргументы, с условиями допустимости результата.

Определение 2.17. Если в развёртке схемы S имеется фрагмент

$$\ldots\ if\ \ \alpha.p \supset \ldots\ if\ \alpha.\beta.q \supset \ldots fi \ldots fi,$$

то $\alpha.\beta.q$ и дополнительное условие $\alpha.\beta.\text{-}q$ называются более сильными условиями по сравнению с $\alpha.p$, этот факт отмечается записью $\alpha.\beta.q \rhd \alpha.p$ (символ «-» используется как знак принадлежности имени дополняющего селектора, а не обозначение операции). Если $\beta\in\mathbb{A}$, то $\alpha.\beta.q$ и $\alpha.\beta.\text{-}q$ называются непосредственно более сильными условиями, чем $\alpha.p$, записывается это в форме $\alpha.\beta.q \blacktriangleright \alpha.p$.

Определение 2.18. Пусть имеется рекурсивная схема S. Рассмотрим фрагмент её развёртки: $(\ldots S_{ij}(a_{ij})\ldots if\ldots\alpha.S_{ij}(a_{ij})\ldots fi\ldots)$. Атрибут $\alpha.a_{ij}$ будем называть следующим за a_{ij} и обозначать его $n^k_\alpha(a_{ij})$, где k – длина префикса α.

Введённые отношения существенно используются при построении и анализе алгоритмов установления выводимости.

Наконец, в *заключительном параграфе* фиксируются и обсуждаются понятия интерпретации С-модели, принадлежности кортежа типу и предложения вычислимости.

Определение 2.19. Интерпретация **I** С-модели M задаёт:

- для каждой схемы $S\in\mathbb{E}$ непустое множество (первичный тип) $S|_{\mathbf{I}}$;
- для каждого $f\in\mathbb{F}$ – отображение $f|_{\mathbf{I}}:S_1|_{\mathbf{I}}\times\ldots\times S_n|_{\mathbf{I}}\to S_0|_{\mathbf{I}}$;
- для каждого $p\in\mathbb{P}$ – булевское отображение $p|_{\mathbf{I}}:S_1|_{\mathbf{I}}\times\ldots\times S_n|_{\mathbf{I}}\to\{\boldsymbol{И},\boldsymbol{Л}\}$.

Определение 2.20. Рассмотрим M=(…S…). Пусть $A\leftrightharpoons\{a_0/p_{a0},\ldots,a_n/p_{an}\}$ – множество атрибутов развертки S, $\varphi\leftrightharpoons f{:}b_1/p_{b0},\ldots,b_k/p_{bk}\to b_0/p_{b0}$ – функциональная связь этой же развёртки ($b_i/p_{bi}\in A$, i=0,…,k), а r – некоторый кортеж, определяемый заголовком схемы S. Говорят, что $\varphi|_{\mathbf{I}}$ имеет смысл для кортежа r, если $p_{bi}|_{\mathbf{I}}=\boldsymbol{И}$, i=0,…,k. Кортеж r принадлежит типу $S|_{\mathbf{I}}$, если на r выполнены все функциональные соотношения, которых имеют смысл для него.

Определение 2.21. Выражение вида $F : A \to X$, где F – (программный) терм в сигнатуре $\boldsymbol{\Sigma}$, построенный по правилам, приведенным в *третьей* главе, $\boldsymbol{A}$ и $\boldsymbol{X}$ – непустые наборы y-атрибутов, называется *предложением вычислимости* (**ПВ**). Наборы A и X называются аргументами и результатами **ПВ** соответственно.

Для определенного выше понятия нарушается классическая трактовка функциональности, однако традиционные приемы («**ПВ** *есть объединение* **ФС***, таких что…*») позволяют не выходить за рамки классики.

Глава 3

В главе описывается исчисление, в рамках которого осуществляется построение программного терма-решения для задачи T=(A_0,X_0,S), когда S является схемой РДС-модели. Показывается полнота и непротиворечивость теории. Предлагаются нецелеориентированные (обратные) стратегия и алгоритм решения задачи T, для алгоритма исследуется вопрос корректности, и устанавливаются сложностные оценки.

В рамках построенного исчисления, исходя из описания задачи T, формулируется и доказывается теорема (существования) $S \vdash_M (\exists F)\ F{:}A_0 \to X_0$, после чего из доказательства извлекается программный терм-решение F.

Определение 3.1. Пусть M=(…S…) – РДС-модель, пусть также $a_1/p_1,\ldots,a_n/p_n$, $x_1/q_1,\ldots,x_m/q_m$ – y-атрибуты развёртки S, $\Psi \leftrightharpoons F{:}a_1/p_1,\ldots,a_n/p_n \to x_1/q_1,\ldots,x_m/q_m$ – предложение вычислимости, а r – кортеж, определяемый заголовком развёртки при некоторой интерпретации **I**. Говорят, что $\Psi|_{\mathbf{I}}$ имеет смысл для кортежа r, если $p_i|_{\mathbf{I}}=\boldsymbol{И}$, i=1,…,n и $q_j|_{\mathbf{I}}=\boldsymbol{И}$, j=1,…,m.

Определение 3.2. Говорят, что **ПВ** $\Psi \leftrightharpoons F{:}A \to X$ удовлетворяет схеме S, если при любой интерпретации **I** тип $S|_{\mathbf{I}}$ не содержит двух кортежей, для которых $\Psi|_{\mathbf{I}}$ имеет смысл, таких что их компоненты совпадают для всех атрибутов множества A, но не совпадают по крайней мере для одного атрибута множества X. При этом для произвольного кортежа r, который принадлежит отношению $S|_{\mathbf{I}}$, результатом применения $F|_{\mathbf{I}}$ к компонентам r, именованным атрибутами множества A, является набор компонент, именованных атрибутами множества X. Все **ФС** схемы удовлетворяют ей по определению.

В теории имеется три сорта термов, которые строятся в сигнатуре Σ согласно следующему определению.

Определение 3.3. (1) Термы первого сорта (a-термы) представляют собой упорядоченные списки атрибутов (атрибуты понимаются в смысле второй главы). Они порождаются элементами $\mathbb{A}$. (2) Термы второго сорта (y-термы) строятся из термов первого сорта и предикатных символов из $\mathbb{P}$ согласно следующей рекурсивной процедуре. Пусть p – селектор, реализуемый n-местной булевской функцией, $p \in \mathbb{P}$; A – n-элементный терм первого сорта, такой, что его i-й атрибут связан с той же схемой, что и i-й аргумент p; P и Q – термы второго сорта; α – некоторый атрибут; «-» и «&» – символы операций дополнения и конкатенации (конъюнкции), соответственно. Тогда перечисленные выражения являются термами второго сорта: $\alpha.p(A)$, $-P$, $P\&Q$. Других термов второго сорта нет. (3) Наконец, термы третьего сорта (программные) строятся из термов всех сортов по следующим правилам. Пусть P – терм второго сорта; f, g, h – функциональные символы (элементы множества $\mathbb{F}$); F_1, F_2, F_3 – термы третьего сорта и имеется вхождение g в F_3; α – некоторый атрибут; $F_3[h/g]$ – обозначение замены вхождения g на h в терме F_3. Тогда $\alpha.h{:}\ldots \to \ldots$ – терм третьего сорта; $F_1;F_2$ – терм (оператор композиции); *if* P *then* F_1 *else* F_2 *fi* – терм (оператор ветвления); h= *if* P *then* F_1 *else* $F_3[h/g]$ *fi* – терм (оператор рекурсии). Других термов третьего сорта нет.

В исчисление входит одна схема аксиом и четыре правила вывода.

(0) *схема аксиом*: $\vdash \boldsymbol{N}:\boldsymbol{A} \to \boldsymbol{A}$ ($\boldsymbol{N}$ – тождественное отображение)

(1) $$\frac{\vdash \boldsymbol{F}:\boldsymbol{A}\to\boldsymbol{X},\boldsymbol{Z}}{\vdash \boldsymbol{F}:\boldsymbol{A}\to\boldsymbol{X}}$$ (*правило сужения*)

(2) $$\frac{\vdash \boldsymbol{F}:\boldsymbol{A}\to\boldsymbol{X},\boldsymbol{Z}/\boldsymbol{P} \quad \vdash \boldsymbol{f}:\boldsymbol{Z}\to\boldsymbol{x}/\boldsymbol{p}}{\vdash \boldsymbol{F};\boldsymbol{f}:\ \boldsymbol{A},\boldsymbol{Z}/\boldsymbol{P}\to\boldsymbol{X},\boldsymbol{Z},\boldsymbol{x}/\boldsymbol{P}\&\boldsymbol{p}}$$ (*правило суперпозиции*)

здесь $\boldsymbol{P}$ – конъюнкция условий *достижимости* атрибутов из $\boldsymbol{Z}$, а $\boldsymbol{p}$ – условие *допустимости* $\boldsymbol{x}$ (напомним, что условие достижимости – это условие, при котором в процессе построения доказательства обеспечивается достижимость некоторого атрибута, а условие допустимости – это условие, при котором атрибут имеет смысл в РДС-модели).

(3) $$\frac{\vdash \boldsymbol{F}_1:\boldsymbol{A}\to\boldsymbol{X},\boldsymbol{x}/\boldsymbol{Q}\&\boldsymbol{p} \quad \vdash \boldsymbol{F}_2:\boldsymbol{A}\to\boldsymbol{X},\boldsymbol{x}/\boldsymbol{Q}\&\text{-}\boldsymbol{p}}{\vdash \boldsymbol{if}\ \boldsymbol{p}\ \boldsymbol{then}\ \boldsymbol{F}_1\ \boldsymbol{else}\ \boldsymbol{F}_2\ \boldsymbol{fi}:\boldsymbol{A}\to\boldsymbol{X},\boldsymbol{x}/\boldsymbol{Q}}$$ (*правило ветвления*)

(4) $$\frac{\vdash \boldsymbol{F}_1:\boldsymbol{A}\to\boldsymbol{X},\boldsymbol{x}/\boldsymbol{Q}\&\boldsymbol{p} \quad \boldsymbol{g}_1:\boldsymbol{n}_{\alpha_1}^{k_1}(\boldsymbol{A})\to\boldsymbol{n}_{\alpha_1}^{k_1}(\boldsymbol{X}),\ldots,\boldsymbol{g}_s:\boldsymbol{n}_{\alpha_s}^{k_s}(\boldsymbol{A})\to\boldsymbol{n}_{\alpha_s}^{k_s}(\boldsymbol{X}) \vdash \boldsymbol{F}_2:\boldsymbol{A}\to\boldsymbol{X},\boldsymbol{x}/\boldsymbol{Q}\&-\boldsymbol{p}}{\boldsymbol{h}=\boldsymbol{if}\ \boldsymbol{p}\ \boldsymbol{then}\ \boldsymbol{F}_1\ \boldsymbol{else}\ \boldsymbol{F}_2[\boldsymbol{h}/\boldsymbol{g}_1]\ldots[\boldsymbol{h}/\boldsymbol{g}_s]\ \boldsymbol{fi}:\boldsymbol{A}\to\boldsymbol{X}/\boldsymbol{Q}}$$ (*правило введения рекурсии*)

здесь $X/P \leftrightharpoons x_1/P,\ldots,x_m/P$, n^k_α – символ введённого выше отношения порядка.

Схема аксиом позволяет формировать «начальные» для вывода **ПВ** предложения вычислимости, отталкиваясь от заданных атрибутов из формулировки задачи *T*. Правило сужения (1) применяется, если необходимо исключить из рассмотрения часть атрибутов, достижимость которых доказана, например, при использовании правила (4).

Правило суперпозиции (композиции) (2) понимается как обычно: если показана достижимость *Z* и известно, что по *Z* вычислим *x*, то можно считать, что показана достижимость *x* с учётом условий, при которых достижимы атрибуты из *Z*. Правило ветвления (3) позволяет ослабить условие при атрибутах: если показана достижимость *x* при условии *Q*&*p* (терм F_1) и одновременно при условии *Q*&-*p* (терм F_2), то **ПВ**, у которого программный терм построен с помощью оператора ветвления, обеспечивает достижимость *x*, с элиминированием условий *p* и -*p*.

Правило введения рекурсии (4) применимо только для случая рекурсивной схемы. Установить рекурсивные схемы модели позволяет процедура, описанная в теореме 2.2. Правило означает, что если (*а*) показана достижимость *X*/*Q*&*p* по *A* посредством терма F_1 и (*б*) из предположения о достижимости $n^{k1}{}_{\alpha 1}(X),\ldots,n^{ks}{}_{\alpha s}(X)$ следует достижимость *X*/*Q*&-*p*, то

достижимо X/Q (при элиминированных p и $-p$) посредством применения рекурсивного программного терма

$$h=if\ p\ then\ F_1\ else\ F_2[h/g_1]...[h/g_s]\ fi:\ A \rightarrow X/Q.$$

Построенную теорию назовём исчислением ***SR*** (*structural recursive calculi*).

Далее исследуются важные свойства исчисления ***SR***. Показывается, что правила ***SR*** для некоторой схемы РДС-модели позволяют получить все **ПВ**, которые удовлетворяют этой схеме (полнота исчисления) и что, используя правила ***SR***, можно построить только **ПВ**, удовлетворяющие схеме (корректность ***SR***).

Теорема 3.1. Исчисление ***SR*** является корректным относительно понятия **ПВ**, удовлетворяющего схеме ***S***.

Доказательство. Утверждение теоремы означает следующее: если **ПВ** $\mathbf{\Phi}\equiv F{:}A\rightarrow X$ выведено с помощью правил ***SR*** на основании информации об исходной схеме S модели M, то, независимо от интерпретации **I**, для любых двух кортежей типа $S|_{\mathbf{I}}$, таких что $\mathbf{\Phi}|_{\mathbf{I}}$ имеет для них смысл, компоненты, соответствующие атрибутам A и X в этих кортежах, совпадают и связаны соотношением, задаваемым $\mathbf{\Phi}|_{\mathbf{I}}$. Доказательство строится конструктивно – рассматривается (произвольная) схема и все правила поочередно и показывается, что правила позволяют выводить из **ПВ**, удовлетворяющих исходной схеме, только те **ПВ**, которые также ей удовлетворяют. ■

При исследовании полноты исчисления ***SR*** необходимо показать, что если некоторое **ПВ** не может быть выведено из информации о схеме S по правилам ***SR***, то это **ПВ** не удовлетворяет схеме S. Вводится понятие замыкания набора атрибутов относительно схемы S, которое необходимо для доказательства теоремы полноты исчисления.

Определение 3.4. Пусть $S(r) = r.(A,X, \ldots \mid filter)$ и пусть A – некоторый набор атрибутов схемы S. Замыканием A^+ для A относительно функциональных связей схемы S называют объединение всех атрибутов X таких, что **ПВ** $F{:}A\rightarrow X$ может быть построено с использованием правил ***SR***.

Важное свойство замыкания A^+ определяется следующей леммой.

Лемма*. **ПВ** $F{:}A\rightarrow X$ выводимо по правилам ***SR***, если и только если $X \subseteq A^+$.

В заключение *первого параграфа*, с использованием *леммы**, формулируется и доказывается *теорема полноты*.

Теорема 3.2. Исчисление ***SR*** является полным относительно понятия **ПВ**, удовлетворяющего схеме.

Во втором параграфе предлагаются и исследуются стратегии и алгоритм установления выводимости в исчислении ***SR***. Алгоритм использует некоторую ограниченную стратегию поиска решения задачи $T=(A_0,X_0,S)$.

Входной информацией для алгоритма является описание модели *M* и формулировка задачи *T*. Доказательство теоремы существования заключается в управляемом стратегией последовательном применении правил ***SR*** к аксиомам и ранее полученным утверждениям. Аксиомами выступают **ФС** из описания схемы модели *M*. Поиск доказательства ведётся от данных к целям.

Алгоритм, включающий и стратегию вывода, описывается с использованием псевдокода, структуры управления которого традиционны для языков программирования высокого уровня. При описании алгоритма используются следующие обозначения и соглашения:

- C_k – множество *y*-атрибутов исходной схемы, достижимость которых установлена в текущий момент;
- *по*-аксиома – *проблемно-ориентированная* или *предметная* аксиома – **ФС** из (развёртки) исходной схемы;
- вход в подсхему – наличие в C_k y-атрибутов, которые принадлежат подсхеме текущей схемы;
- A_0 и *S* берутся из формулировки задачи *T*.

Описание алгоритма$^+$:

«установить i=0, C_0=A_0»;
«поднять флаг “*продолжать доказательство*”»;
«текущей схемой объявить *S*»;
while «поднят флаг “*продолжать доказательство*”» ***do***
  if «имеется *по*-аксиома схемы, аргументы которой входят в C_i» ***then***
    «применить правило (2), и, если возможно, правило (3)»;
    «переопределить C_{i+1}: к C_i добавить новые y-атрибуты либо изменить условия достижимости определённых ранее»;
    «установить i=i+1»;
    «если определился вход в подсхему, запомнить её»
  else if «имеется подсхема, в которую определён вход» ***then***
    «объявить её текущей схемой»
  else if «текущая – рекурсивная схема» ***then***
    «применить правило (4); установить i=i+1»;
    «переопределить C_{i+1}: к C_i добавить новые y-атрибуты либо изменить условия достижимости определённых ранее»
  else «объявить текущей внешнюю схему»
  fi fi fi;
  if («текущая – исходная схема *S*» ***and***
    «нет *по*-аксиом, аргументы которых входят в C_i») ***then***
    «опустить флаг “*продолжать доказательство*”» ***fi***
od.

Приведенный алгоритм$^+$ по своей сути строит замыкание A_0^+, учитывая это, корректность алгоритма$^+$ определяется следующей теоремой.

Теорема 3.3. Приведённый алгоритм$^+$ корректно вычисляет $A_0{}^+$.

Завершается *второй параграф* формулировкой и обсуждением элементов стратегии, связанных с использованием правила введения рекурсии, более точно – эффективных (линейных по длине исходного описания схемы) способов определения наборов аргументов (A) и результатов (X) искомого *рекурсивного* **ПВ**.

В *следующем параграфе* приводятся соображения по практической реализации предложенного алгоритма вывода, и устанавливается, что полный алгоритм (с включением правила введения рекурсии) *максимальной теоретической* оценкой имеет $\boldsymbol{O}(|M|^3)$, а реально является не более, чем квадратичным по длине исходного описания модели M.

В *последнем* параграфе главы суммируются выводы. Представленный подход к установлению выводимости формул является весьма привлекательным для реализации широкого круга прикладных когнитивных систем, учитывая показанные свойства полноты, эффективной распознаваемости и эффективной разрешимости.

Теория РДС-моделей может служить формальным базисом объектно-ориентированного подхода. – Свойства иерархичности, инкапсуляции и наследования естественным образом присущи представленной теории. Более того, малозначимое в рамках диссертационного исследования замечание о *динамичности множества первичных типов* в контексте ООП является довольно существенным – оно позволяет выстраивать иерархию понятий в двух направлениях, обеспечивая, по мере необходимости, как *агрегацию*, так и *декомпозицию*.

Наконец, опираясь на механизм *схем* теории РДС-моделей, можно ввести понятие «реализуемых предикатов» и на этой основе строить когнитивные системы, обладающие свойствами теорий высших порядков, но свободные от их недостатков.

Глава 4

В этой и заключительной главах обсуждаются варианты обратного метода установления выводимости для модальных теорий[1]. Приведенные результаты относятся к логике знания ***KT***, хотя используемый подход был с успехом использован автором и его учениками и для других модальных теорий (см., например, [20]).

---

[1] ***Фейс Р.*** *Модальная логика. – М.:Наука. – 1974.*

Необходимость таких исследований определяется недостаточностью существующих информационных технологий для анализа слабо-формализованных предметных областей, а также предметных областей с модальными связями. Популярные *нечеткие логики* далеко не всегда уместны и адекватны, а работа с модальностями в рамках традиционных подходов, как правило, реализуется введением *эклектичных* вероятностных механизмов, снижающих степень уверенности в корректности реализованного на таком базисе программного обеспечения. Модальные теории, в свою очередь, естественным образом включают понятия *необходимости*, *возможности*, *персонифицированности* и т.д.

Логика ***KT*** является более богатой системой по сравнению с минимальной модальной логикой ***K*** (за счет добавления аксиомы ***T***: $\Box A \supset A$) и интересна тем, что аксиома ***T*** отвечает свойству рефлексивности бинарного отношения *R структуры* модальной логики. Логика ***KT*** имеет большое значение при реализации систем обработки знаний, когда модальный оператор □ принимает трактовку «известно» (схема ***T*** как раз и требует, чтобы то, что *известно,* являлось *истинным*).

В *первом параграфе* строго вводятся основные определения и базовые исчисления, необходимые для реализации приведенного ниже плана установления выводимости, опирающегося на обратный метод.

Реализация плана осуществляется в четыре этапа:
I. Построение базовых исчислений для ***KT*** – *прямого* секвенциального исчисления $\boldsymbol{KT}_{\mathrm{seq}}$ и настроенного на *анализируемую* формулу **Ф** *обратного* исчисления $\boldsymbol{KT}^{\boldsymbol{\Phi}}_{\mathrm{inv}}$ (в последнем случае формулами $\boldsymbol{KT}^{\boldsymbol{\Phi}}_{\mathrm{inv}}$ являются все подформулы исходной **Ф**).
II. Введение исчислений *путей* $\boldsymbol{KT}^{\boldsymbol{\Phi}}_{\mathrm{path}}$ и $\boldsymbol{KT}^{\boldsymbol{\Phi}}_{\mathrm{IP}}$, кодирующих вывод (связывающих подформулу с примененным к ней в процессе вывода правилом) в базовых исчислениях $\boldsymbol{KT}_{\mathrm{seq}}$ и $\boldsymbol{KT}^{\Phi}_{\mathrm{inv}}$, соответственно.
III. Замена вывода исходной **Ф** формулы в ***KT*** на опровержение отрицания **Ф** в $\boldsymbol{KT}^{\Phi}_{\mathrm{IP}}$ (из соображений технического удобства) с применением стратегий сокращения пространства вывода.
IV. Отображение полученного в $\boldsymbol{KT}^{\boldsymbol{\Phi}}_{\mathrm{IP}}$ вывода в исходную систему.

Пусть **Ф** – формула логики ***KT***. Для анализа **Ф** удобнее использовать не саму систему ***KT***, а эквивалентное ей исчисление секвенций $\boldsymbol{KT}_{\mathrm{seq}}$. Справедлива *теорема*[1] (полноты $\boldsymbol{KT}_{\mathrm{seq}}$): **Ф** невыполнима в ***KT*** тогда и только тогда, когда существует опровержение **Ф** в $\boldsymbol{KT}_{\mathrm{seq}}$. Из упомянутой работы

---

[1] ***Fitting M.*** *Proof methods for modal and intuitionistic logics // Synthesis Library. Reidel Pub. Comp. – 1983. – V.169.*

заимствована и подходящая версия исчисления секвенций $\boldsymbol{KT}_{\text{seq}}$ (p – пропозициональная переменная):

Аксиомы: $\mathbf{\Gamma}, p, \sim p$.

Правила вывода:

$$\frac{\mathbf{\Gamma}, A, B}{\mathbf{\Gamma}, A\wedge B}\ (\wedge);\quad \frac{\mathbf{\Gamma}, A \quad \mathbf{\Gamma}, B}{\mathbf{\Gamma}, A\vee B}\ (\vee);\quad \frac{\mathbf{\Gamma}, A}{\Box\mathbf{\Gamma}, \Diamond A, \mathbf{\Delta}}\ (\Diamond);\quad \frac{\mathbf{\Gamma}, A}{\mathbf{\Gamma}, \Box A}\ (\Box).$$

В рамках обратного метода, поиск опровержения переносится в инверсное исчисление $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{inv}}$ (формулами $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{inv}}$ являются все подформулы исходной $\mathbf{\Phi}$, что и определяет настройку на формулу). Исчисление $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{inv}}$ приводится ниже:

Аксиомы: $p, \sim p$

Правила вывода:

$$\frac{\mathbf{\Gamma}, A, A}{\mathbf{\Gamma}, A}\ (\text{C});\quad \frac{\mathbf{\Gamma}, A \quad \mathbf{\Delta}, B}{\mathbf{\Gamma}, \mathbf{\Delta}, A\vee B}\ (\vee);\quad \frac{\mathbf{\Gamma}, A}{\mathbf{\Gamma}, A\wedge B}\ (\wedge_l);\quad \frac{\mathbf{\Gamma}, B}{\mathbf{\Gamma}, A\wedge B}\ (\wedge_r);$$

$$\frac{\mathbf{\Gamma}, A}{\Box\mathbf{\Gamma}, \Diamond A}\ (\Diamond);\quad \frac{\mathbf{\Gamma}}{\Box\mathbf{\Gamma}, \Diamond A}\ (\Diamond);\quad \frac{\mathbf{\Gamma}, A}{\mathbf{\Gamma}, \Box A}\ (\Box).$$

В общем случае неясно, как в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{inv}}$ находить опровержение произвольной секвенции. Для доказательства полноты $\boldsymbol{KT}^{\Phi}_{\text{inv}}$ доказывается *лемма подсеквенциальности*, которая позволяет переносить найденное в $\boldsymbol{KT}_{\text{seq}}$ опровержение $\mathbf{\Phi}$ в исчисление $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{inv}}$ (индукцией по длине вывода $\mathbf{\Gamma}$ в $\boldsymbol{KT}_{\text{seq}}$).

Лемма (*подсеквенциальности*). Пусть $\mathbf{\Phi}$ – формула $\boldsymbol{KT}$, $\mathbf{\Gamma}$ – секвенция, состоящая из подформул $\mathbf{\Phi}$ и имеющая опровержение в $\boldsymbol{KT}_{\text{seq}}$, тогда существует секвенция $\mathbf{\Delta}$ такая, что $\mathbf{\Delta} \dot{\subseteq} \mathbf{\Gamma}$, и $\mathbf{\Delta}$ имеет опровержение в $\boldsymbol{KT}^{\Phi}_{\text{inv}}$.

Теорема 4.1 (*полноты* $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{inv}}$). Формула $\mathbf{\Phi}$ системы $\boldsymbol{KT}$ является невыполнимой тогда и только тогда, когда она имеет опровержение в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{inv}}$.

Во *втором* параграфе в стиле А.А.Воронкова[1] для заданной формулы $\mathbf{\Phi}$ строится исчисление путей $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$ [14]. Поиск вывода в этом исчислении технически прост, а дерево вывода пустого пути представляет каркас доказательства $\mathbf{\Phi}$ в $\boldsymbol{KT}_{\text{seq}}$.

Определение 4.1. Пусть C – формула $\boldsymbol{KT}_{\text{seq}}$, C_1 и C_2 – ее подформулы. Путем в $\mathbf{\Phi}$ или $\mathbf{\Phi}$-путем будем называть любую конечную последовательность символов $\wedge_l$, $\wedge_r$, $\vee_l$, $\vee_r$, $\Box$, $\Diamond$, которая удовлетворяет следующим правилам:

- Пустой путь (элемент) ε есть $\mathbf{\Phi}$-путь.
- Пусть π есть $\mathbf{\Phi}$-путь, тогда

---

[1] ***Voronkov A.*** *How to optimize proof-search in modal logics: new methods of proving redundancy criteria for sequent calculi // ACM Transactions and Computational logic. – 2001. – V. 1. – pp. 1-35*

- если C имеет вид $C_1 \wedge C_2$, то $\pi\wedge_l$ и $\pi\wedge_r$ есть **Ф**-пути ($\wedge$-путь),
- если C имеет вид $C_1 \vee C_2$, тогда $\pi\vee_l$ и $\pi\vee_r$ есть **Ф**-пути ($\vee$-путь),
- если C имеет вид $\Box C_1$, тогда $\pi\Box$ есть **Ф**-путь ($\Box$-путь),
- если C имеет вид $\Diamond C_1$, тогда $\pi\Diamond$ есть **Ф**-путь ($\Diamond$-путь).

Подформула для **Ф** на пути π, обозначаемая $\mathbf{\Phi}|_\pi$, определяется традиционно. Исчисление путей $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$ имеет вид:

Аксиомы: $\mathbf{\Gamma}$, π_1, π_2

Правила вывода:

$$\frac{\mathbf{\Gamma}, \pi\wedge_l, \pi\wedge_r}{\mathbf{\Gamma}, \mathbf{\Pi}, \pi}\ (\wedge);\quad \frac{\mathbf{\Gamma}, \pi\vee_l \quad \Gamma, \pi\vee_r}{\mathbf{\Pi}, \pi}\ (\vee);\quad \frac{\mathbf{\Pi}\Box, \pi\Diamond}{\mathbf{\Gamma}, \pi}\ (\Diamond);\quad \frac{\mathbf{\Pi}\Box, \pi\Box}{\mathbf{\Gamma}, \pi}\ (\Box).$$

Все *пути*, входящие в *секвенции* $\mathbf{\Pi}=\pi_1,\ldots,\pi_n$ и $\mathbf{\Pi}\Box=\pi_1\Box,\ldots,\pi_n\Box$ *являются* **Ф**-*путями.*

Определение 4.2. *Образом* секвенции пути или дерева вывода в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path.}}$ называется дерево, полученное из первоначальных формул заменой каждого пути π на $\mathbf{\Phi}|_\pi$.

Для доказательства полноты $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$ используется соответствующая

Лемма (*бимоделирования для* $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$). (1) Пусть D – дерево вывода в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$, тогда образом D является дерево вывода **Ф** в $\boldsymbol{KT}_{\text{seq}}$. (2) Пусть D' – дерево вывода секвенции $A_1,\ldots,A_n$ в $\boldsymbol{KT}_{\text{seq}}$ и $\pi_1,\ldots,\pi_n$ – такие, что $\mathbf{\Phi}|_{\pi i}=A_i\ \forall i=1,\ldots,n$. Тогда существует дерево D для $\pi_1,\ldots,\pi_n$ в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$ такое, что D' является образом дерева D. (3) Пункты (1) и (2) справедливы, если везде «дерево вывода» заменить на «опровержение».

Теорема 4.2 (*полноты для* $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$). Формула **Ф** логики $\boldsymbol{KT}$ невыполнима тогда и только тогда, когда пустой путь ε имеет опровержение в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path.}}$.

В *третьем* параграфе собраны результаты, связанные с анализом избыточностей и полноты $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{IP}}$.

Определим *обратное* исчисление путей $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{IP}}$. Пусть $\pi_1,\ldots,\pi_n$ – **Ф**-пути. $\mathbf{\Pi} = \pi_1,\ldots,\pi_n$, $\mathbf{\Pi}\Box = \pi_1\Box,\ldots,\pi_n\Box$ и $\mathbf{\Gamma}$ – последовательности путей. Тогда аксиомами $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{IP}}$ являются любые формулы вида: π_1, π_2, где $p=\mathbf{\Phi}|_{\pi 1}$, $\sim p=\mathbf{\Phi}|_{\pi 2}$, для некоторой пропозициональной переменной p.

Правила вывода:

$$\frac{\mathbf{\Gamma}, \pi\wedge_l}{\mathbf{\Gamma}, \pi}\ (\wedge_l);\quad \frac{\mathbf{\Gamma}, \pi\wedge_r}{\mathbf{\Gamma}, \pi}\ (\wedge_r);\quad \frac{\mathbf{\Gamma}, \pi\vee_l \quad \mathbf{\Delta}, \pi\vee_r}{\mathbf{\Gamma}, \mathbf{\Delta}, \pi}\ (\vee);\quad \frac{\mathbf{\Gamma}, \pi, \pi}{\mathbf{\Gamma}, \pi}\ (\text{C});$$

$$\frac{\mathbf{\Gamma}\Box}{\mathbf{\Gamma}, \pi}\ (\Diamond+);\quad \frac{\mathbf{\Pi}\Box, \pi\Diamond}{\mathbf{\Pi}, \pi}\ (\Diamond);\quad \frac{\mathbf{\Pi}\Box, \pi\Box}{\mathbf{\Pi}, \pi}\ (\Box)$$

Заключительный параграф главы содержит резюмирующие выводы. В целях создания приемлемых условий программной реализации алгоритма

обратного метода для модальных систем построены специальные исчисления путей: прямое и обратное. Исчисления позволяют проследить последовательность вывода с использованием соответствующего (построенного) бинарного дерева, которое наглядно отображает шаги вывода проверяемой формулы из заданных аксиом. Достоверность и корректность планирования при этом опираются на доказанные в главе леммы и теоремы.

Предлагаемый в главе механизм вывода является достаточно естественным и наглядным для человека и допускает контроль со стороны прикладного эксперта или технического специалиста. Он (механизм) также хорошо приспособлен для практических реализаций, в частности, машин вывода в различных системах принятия решения, экспертных и других когнитивных комплексах. Естественность и наглядность становятся весьма существенными, поскольку составной частью любой *KBS* по сложившейся традиции должна быть мощная подсистема, объясняющая найденные решения.

Глава 5

В пятой главе приведены утверждения, описывающие алгоритмически полезные свойства множества *всех* возможных деревьев вывода в исчислениях путей. Подчеркнем, что главной целью является ограничение *пространства* поиска опровержения некоторым (собственным) подмножеством деревьев вывода, что достигается за счет введения специального упорядочения на множестве **Ф**-путей.

Сначала рассматриваются свойства, наиболее слабо ограничивающие пространство поиска с помощью упорядочения. Классический метод резолюций упорядочивает литеры в дизъюнктах и требует, чтобы правило резолюций применялось только тогда, когда наибольшие литералы в обоих дизъюнктах разрешимы. Введем подобные ограничения на построение деревьев вывода для логической системы ***KT***. Преобразуем классическое упорядочивание литер на модальный случай. В классическом исчислении возможно использовать любое упорядочение на подформулах **Ф**, которое принимает во внимание префиксное отношение. Непосредственный перенос подобного на модальные системы невозможен, поскольку не каждое упорядочение на путях сохраняет полноту. – Рассмотрим вывод:

$$\frac{\wedge_l\Box,\ \wedge_r\wedge_l\Box\vee_l,\ \wedge_r\wedge_r\wedge_l\Diamond \qquad \wedge_l\Box,\ \wedge_r\wedge_l\Box\vee_r,\ \wedge_r\wedge_r\wedge_l\Diamond}{\wedge_l\Box,\ \wedge_r\wedge_l\Box,\ \wedge_r\wedge_r\wedge_l\Diamond}\ (\vee).$$

Существенно то, что любой ($\vee$)-вывод применяется выше правила ($\Diamond$), потому что правило ($\Diamond$) не применимо к верхним секвенциям. Тем не менее, если определено упорядочение на путях, где $\wedge_r\wedge_l\Box\vee_l$ является наименьшим в первой посылке, то возможность применения ($\vee$) первым будет исключена, и

доказательство не будет найдено. Очевидно, что понятие **Ф**-упорядочения в модальных логиках является более сложным – определим его.

Определение 5.1. Пути назовем братьями, если один имеет вид $\pi\wedge_l$, а другой – $\pi\wedge_r$, либо $\pi\vee_r$ и $\pi\vee_l$.

Обозначения. *Везде ниже символом* $\ast$ *будем обозначать* $\wedge$ *или* $\vee$*; символом* $*$ *–* r *или* l*; символом* $\Box$ *–* $\Box$ *или* $\Diamond$*. Запись* $\pi'\sqsubseteq\pi$ *означает «*π' *есть префикс* π*».*

Определение 5.2. Для заданной формулы **Ф** назовем **Ф**-упорядочением любое отношение полного порядка $\succ$ на множестве всех **Ф**-путей, удовлетворяющее условиям:

1. $\pi_1\succ\pi_2$, всякий раз когда
   a) модальная длина π_1 строго больше модальной длины π_2 или
   b) π_1 и π_2 имеют одинаковую модальную длину, последний символ π_1 – $\ast_*$, а последний символ π_2 – $\Box$, или
   c) π_1 и π_2 имеют одинаковую модальную длину и $\pi_2\sqsubseteq\pi_1$, или
   d) π_1 и π_2 имеют одинаковую модальную длину, последний символ π_1 – $\ast_*$, последний символ π_2 – $\ast_*$, и при этом неверны оба утверждения $\pi_2\sqsubseteq\pi_1$ и $\pi_1\sqsubseteq\pi_2$, но π_1 имеет большую обычную длину, чем π_2.
2. Не существует пути между двумя братьями, то есть не существует **Ф**-путей π_1, π_2, π_3 таких, что $\pi_1\succ\pi_2\succ\pi_3$ и π_1, π_3 – братья.

Содержательно, отношение $\succ$ позволяет управлять порядком применения правил – сначала правила применяются к формулам, большим относительно $\succ$. Помимо этого отношение требует, чтобы заключение любого правила было меньше, чем любая его предпосылка в мультимножественном упорядочении. Условие (1a) гарантирует, что заключение меньше посылки при применении правил ($\Diamond$) или ($\Diamond$+). Условие (1b) введено для того, чтобы применение ($\Diamond$) или ($\Diamond$+) к секвенции, содержащей путь $\pi\ast_*$ не дало неполное исчисление. Условия (1c-d) и (2) являются не только техническими и служат для облегчения доказательств утверждений этого параграфа, но и однозначно определяют любые два пути по отношению к порядку $\succ$, что важно в плане реализации.

Обозначение. *Будем использовать запись* $\pi_1\succeq\pi_2$*, если* $\pi_1\succ\pi_2$ *или* $\pi_1=\pi_2$*.*

Для классической логики полнота метода резолюций с упорядочением доказывается чисто семантически. В случае модальной системы ***KT*** необходимо показать, что стратегия выбора наибольшей формулы (или пути) в дизъюнкте не конфликтует с критериями избыточности, рассмотренными

ранее. Доказательство полноты проводим в два этапа. На первом доказываем свойства деревьев вывода в $\boldsymbol{KT}^{\Phi}_{\mathrm{path}}$, а на втором переносим их в обратное исчисление $\boldsymbol{KT}^{\Phi}_{\mathrm{IP}}$, используя соответствующий вариант *леммы подсеквенциальности*.

При доказательстве полноты появляются трудности, связанные с тем, что требование упорядочения формулируется в терминах предпосылок вывода, в то время как доказательство полноты для $\boldsymbol{KT}^{\Phi}_{\mathrm{path}}$ использует следствия. Это несколько осложняет введение порядка на путях. Приведем вариант алгоритма, который упорядочивает **Ф**-пути.

Алгоритм, работает с секвенциями из множества путей, эти секвенции обозначаются $S_n \succ S_{n-1} \succ \ldots \succ S_0$. Содержательно запись означает, что для любого $\pi \in S_i$ и $\pi' \in S_{i-1}$ выполнено $\pi \succ \pi'$. Пути, принадлежащие одинаковым S_i, еще не упорядочены, но будут упорядочены позднее. На каждом шаге выбирается некоторое множество S_i в секвенции, содержащее два или более членов и заменяется двумя или более множествами $S_{i1} \succ \ldots \succ S_{ik}$ такими, что $S_{i1} \cup \ldots \cup S_{ik} = S_i$. Алгоритм заканчивает работу, когда каждое множество содержит только один элемент.

Алгоритм$^{\rightarrow}$ упорядочения.

1. *Устанавливаем* S_i, *как множество путей в* **Ф** *модальной длины i.*
2. *Для всех* S_i, *исключая последнее, выполняем:*
   3. *выбираем все пути* $\pi_1, \ldots, \pi_n$ *в* S_i *, заканчивающиеся* $\Diamond$;
   4. *разбиваем* S_i *на* $S_i \setminus \{\pi_1, \ldots, \pi_n\} \succ \{\pi_1\} \succ \ldots \succ \{\pi_n\}$;
   5. *разбиваем* S_0 *на* $S_0 \setminus \{\varepsilon\} \succ \{\varepsilon\}$;
6. *Пока существуют* $\boldsymbol{S}_i$ *с более чем одним членом, выполняем:*
   7. *выбираем* $\pi\aleph_l$ и $\pi\aleph_r$ *– два брата в* S_i *такие, что* $\pi \notin S_i$;
   8. *выбираем L и R – множества всех префиксов из* $\pi\aleph_l$ *и* $\pi\aleph_r$ *соответственно*;
   9. *разбиваем* S_i *на* $S_i \setminus (L \cup R) \succ R \succ L \succ \{\pi\aleph_r\} \succ \{\pi\aleph_l\}$.

Когда *алгоритм*$^{\rightarrow}$ завершается, секвенция состоит из одноэлементных множеств, при этом говорят, $\pi \succeq \pi'$, если имеет место $\ldots \{\pi\} \succ \ldots \succ \{\pi'\} \ldots$.

Следующая лемма гарантирует, что *алгоритм*$^{\rightarrow}$ удовлетворяет определению **Ф**-упорядочения.

Лемма$^{\rightarrow}$. Любое упорядочение, полученное *алгоритмом*$^{\rightarrow}$ на формуле **Ф** является **Ф**-упорядочением.

Поскольку на любом шаге *алгоритма*$^{\rightarrow}$ мы получаем **Ф**-упорядочение, имеет место очевидное *следствие:* ***Ф-упорядочение существует.***

Понятие **Ф**-упорядочения введено для того, чтобы доказать существование опровержения в форме, связанной с этим упорядочением. **Ф**-упорядочение сокращает пространство поиска вывода только таких опровержений. Для дальнейшего понадобится ряд дополнительных соглашений и определений.

Обозначение. *Пусть* π *– путь,* **Г** *– секвенция путей. Запись* $\pi \succ \mathbf{\Gamma}$ *является сокращением для утверждения, что* $\pi \succ \pi'$ *для любого* π' *из* **Г**.

Определение 5.3. ($\vee$)-вывод в $\mathbf{KT}^{\mathbf{\Phi}}_{\text{path}}$: $\dfrac{\mathbf{\Gamma}, \pi\vee_l \quad \mathbf{\Gamma}, \pi\vee_r}{\mathbf{\Gamma}, \pi}$ ($\vee$)

будем называть относящимся к **Ф**-упорядочению $\succ$, если $\pi\vee_l \succ \mathbf{\Gamma}$ и $\pi\vee_r \succ \mathbf{\Gamma}$.

Аналогично, ($\wedge$)-вывод в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$: $\dfrac{\mathbf{\Gamma}, \pi\wedge_l, \pi\wedge_r}{\mathbf{\Gamma}, \pi}$ ($\wedge$)

будем называть относящимся к **Ф**-упорядочению $\succ$, если $\pi\wedge_l \succ \mathbf{\Gamma}$ и $\pi\wedge_r \succ \mathbf{\Gamma}$.

Определение 5.4. Дерево вывода в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$ будем называть относящимся к $\succ$, если каждый ($\wedge$) и каждый ($\vee$) вывод из этого дерева относится к $\succ$.

Лемма (*о выводе, относящемся к* $\succ$). Если **Ф** – невыполнимая формула и $\succ$ – **Ф**-упорядочение, то существует опровержение в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\text{path}}$, относящееся к $\succ$.

Непосредственно не очевидно, как доказать это утверждение. Прямое применение индукции по длине пути без учета дополнительных соображений может привести к получению вывода, не относящегося к **Ф-**упорядочению $\succ$. Приведем пример подобной ситуации. Пусть $\pi_1 \succ \pi_2\vee_l$; рассмотрим секвенцию π_1, π_2. Она может быть получена в результате применения правила ($\vee$):

$$\dfrac{\pi_1, \pi_2\vee_l \qquad \pi_1, \pi_2\vee_r}{\pi_1, \pi_2} \ (\vee)$$

Фактически, этот вывод является единственным, и, тем не менее, он не относится к $\succ$. Как отмечалось выше, условия, наложенные на вывод для того, чтобы он относился к $\succ$, сформулированы в терминах посылок, но в таком индуктивном доказательстве мы можем расширить заключение только новым выводом. Для разрешения этой проблемы следует избавиться от секвенций путей, которые приводят к описанной ситуации, ограничиваясь секвенциями специального вида. Это так называемые $\succ$-компактные секвенции путей. Они определяются следующим образом.

Определение 5.5. Пусть $\pi_1,\ldots,\pi_n$ – пути в формуле **Ф** и $\succ$ – **Ф**-упорядочение. Секвенция путей $\mathbf{\Gamma}=\pi_1,\ldots,\pi_n$ называется $\succ$-компактом, если для каждого $i=1,\ldots,n$ выполняются следующие условия: (1) если π_i – $\wedge$-путь, то $\pi_i\wedge_l\succ\mathbf{\Gamma}$ и $\pi_i\wedge_r\succ\mathbf{\Gamma}$; (2) если π_i – $\vee$-путь, то $\pi_i\vee_l\succ\mathbf{\Gamma}$ и $\pi_i\vee_r\succ\mathbf{\Gamma}$.

Используя обозначения для дизъюнкции, конъюнкции и модальностей, переформулируем определение $\succ$-компактности следующим образом:

Определение 5.5*. Пусть $\pi_1,\ldots,\pi_n$ – пути в формуле **Ф** и $\succ$ – **Ф**-упорядочение. Секвенция путей $\mathbf{\Gamma}=\pi_1,\ldots,\pi_n$ *н*азывается $\succ$-компактом, если для каждого $i=1,\ldots,n$ и для каждого $\divideontimes$-пути π_i в **Г** имеем $\pi_i\divideontimes_*\succ\mathbf{\Gamma}$.

Заметим, что $\pi_i\divideontimes_*\succ\pi_i$ гарантируется **Ф**-упорядочением, следовательно, требование $\pi_i\divideontimes_*\succ\mathbf{\Gamma}$ можно заменить на $\pi_i\divideontimes_*\succ\mathbf{\Gamma}\backslash\{\pi_i\}$. Следующая лемма утверждает, что компактная секвенция **Г** не может привести к безвыходной ситуации.

Лемма (*о $\succ$-компактах*). Пусть $\succ$ – **Ф**-упорядочение, тогда (1) посылка каждого ($\Diamond$)-вывода есть $\succ$-компакт; (2) если **Г** – $\succ$-компактная секвенция, встречающая в дереве вывода ε, то каждый ($\divideontimes$)-вывод, имеющий **Г** своим заключением, относится к $\succ$; (3) если **Г** – $\succ$-компактная секвенция, встречающая в дереве вывода ε, и если **Г** содержит по крайней мере один $\vee$-путь или $\wedge$-путь, то существует ($\divideontimes$)-вывод, все посылки которого – $\succ$-компакты.

Теперь, опираясь на эту лемму, можно доказать сформулированную выше *лемму о существовании вывода, относящегося к* $\succ$.

Лемма. Если **Ф** – невыполнимая формула и $\succ$ – **Ф**-упорядочение, то существует опровержение в $\boldsymbol{KT}^{\mathbf{\Phi}}_{\mathrm{path}}$, относящееся к $\succ$.

Во *втором* параграфе приводятся результаты по дальнейшему сокращению пространства поиска вывода. Продолжается модификация инверсного исчисления путей с учетом полученных ранее критериев избыточности.

Определение 5.6. Обозначим через $\boldsymbol{KT}^{\mathbf{\Phi},\succ}_{\mathrm{IP}}$ исчисление, полученное из $\boldsymbol{KT}^{\mathbf{\Phi}}_{\mathrm{IP}}$ удалением:

- всех секвенций путей, содержащих пути различной модальной длины;
- всех секвенций путей содержащих противоречивую пару путей;

- всех выводов, не относящихся к $\succ$.

В $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}$ справедливы леммы бимоделирования и подсеквенциальности.

Лемма (*бимоделирования для* $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}$)**.** (1) Пусть D – дерево вывода в $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}$. Тогда образ D является деревом вывода $\boldsymbol{\Phi}$ в $\boldsymbol{KT}^{\boldsymbol{\Phi}}{}_{\mathrm{path}}$. (2) Пусть D' – дерево вывода секвенции $A_1,\ldots,A_n$ в $\boldsymbol{KT}^{\boldsymbol{\Phi}}{}_{\mathrm{path}}$ и $\pi_1,\ldots,\pi_{\boldsymbol{n}}$ – такие пути, что $\boldsymbol{\Phi} \mid \pi_i = A_i$, $\forall i = 1,\ldots,n$. Тогда существует дерево вывода D для $\pi_1,\ldots,\pi_n$ в $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{path}}$ такое, что D' является образом формул дерева вывода D. (3) Пункты (1) и (2) справедливы, если везде «дерево вывода» заменить на «опровержение».

Заметим, что исчисление $\boldsymbol{KT}^{\boldsymbol{\Phi}}{}_{\mathrm{IP}}$ имеет различные правила для обработки конъюнкции. Поэтому мы модифицируем определение дерева вывода, относящегося к $\boldsymbol{\Phi}$-упорядочению следующим образом.

Определение 5.7. Дерево вывода в $\boldsymbol{KT}^{\boldsymbol{\Phi}}{}_{\mathrm{IP}}$ называется относящимся к $\boldsymbol{\Phi}$-упорядочению, если на вывод наложены следующие условия*:*

a) для каждого ($\vee$)-вывода $\dfrac{\underline{\boldsymbol{\Gamma}, \pi\vee_l\ \boldsymbol{\Delta}, \pi\vee_r}}{\boldsymbol{\Gamma}, \boldsymbol{\Delta}, \pi}$ ($\vee$)

имеет место, только если $\pi\vee_l \succ \boldsymbol{\Gamma}$ и $\pi\vee_r \succ \boldsymbol{\Gamma}$;

b) для каждого ($\wedge_l$)-вывода (соответственно, ($\wedge_r$)-вывода)

$$\dfrac{\underline{\boldsymbol{\Gamma}, \pi\wedge_l}}{\boldsymbol{\Gamma},\pi}\ (\wedge_l) \qquad \dfrac{\underline{\boldsymbol{\Gamma}, \pi\wedge_r}}{\boldsymbol{\Gamma},\pi}\ (\wedge_r)$$

имеет место, только если $\pi\wedge_l \succ \boldsymbol{\Gamma}$ ($\pi\wedge_r \succ \boldsymbol{\Gamma}$).

Лемма (*подсеквенциальности для* $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}$)**.** Пусть D – опровержение ε в $\boldsymbol{KT}^{\boldsymbol{\Phi}}{}_{\mathrm{IP}}$, *о*тносящееся к $\succ$ и I – вывод в D в виде $\dfrac{\underline{\boldsymbol{\Gamma}_1} \ldots \underline{\boldsymbol{\Gamma}_n}}{\boldsymbol{\Gamma}}$,

и пусть даны секвенции путей $\boldsymbol{\Delta}_1,\ldots,\boldsymbol{\Delta}_n$ такие, что каждая $\boldsymbol{\Delta}_i$ есть подсеквенция $\boldsymbol{\Gamma}_i$, тогда существует дерево вывода D' для $\boldsymbol{\Delta}$ в $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathbf{IP}}$ из $\boldsymbol{\Delta}_1,\ldots,\boldsymbol{\Delta}_n$ такое, что $\boldsymbol{\Delta}$ – подсеквенция $\boldsymbol{\Gamma}$.

Лемма подсеквенциальности *обобщается* для произвольного дерева вывода. При этом применяется индукция по длине дерева вывода.

Лемма (*подсеквенциальности для деревьев вывода в* $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathbf{IP}}$). Пусть D – опровержение ε в $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}$, которое относится к $\succ$, D'' – поддерево вывода в D секвенции $\boldsymbol{\Gamma}$ из секвенций $\boldsymbol{\Gamma}_1,\ldots,\boldsymbol{\Gamma}_n$ и $\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_n$ – подсеквенции $\boldsymbol{\Gamma}_1,\ldots,\boldsymbol{\Gamma}_n$ соответственно. Тогда существует дерево вывода D' для секвенции $\boldsymbol{\Delta}$ в $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}$ из $\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_n$ такое, что $\boldsymbol{\Delta}$ является подсеквенцией $\boldsymbol{\Gamma}$.

Теперь может быть доказана теорема полноты для $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}$.

Теорема 5.1 (*полноты* $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}$). Формула **Φ** системы ***KT*** невыполнима тогда и только тогда, когда ε имеет опровержение в исчислении $\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}$.

В *третьем* параграфе предлагается и исследуется еще один метод повышения эффективности вывода. Программы автоматического доказательства теорем для логики предикатов первого порядка часто используют понятие предпосылки в качестве критерия избыточности таким образом, что при построении дерева вывода, из пространства поиска удаляются дизъюнкты, которые предпосылаются другими дизъюнктами.

Введем строго понятие предпосылки для прямого и обратного исчислений путей $\boldsymbol{KT}^{\boldsymbol{\Phi}}{}_{\mathrm{path}}$ и $\boldsymbol{KT}^{\boldsymbol{\Phi}}{}_{\mathrm{IP}}$.

Определение 5.8. Пусть **Π** и **Π′** – секвенции путей и секвенция **Π** является подсеквенцией **Π′**. В этом случае будем говорить, что **Π** предпосылает **Π′**.

Заметим, что сами предпосылочные дизъюнкты не являются избыточными, и их нельзя непосредственно удалять из пространства поиска вывода. Но подобные дизъюнкты становятся избыточными совместно с дизъюнктами, к которым они относятся. Реализация этого соображения позволяет сократить пространство поиска за счет удаления предпосылающих дизъюнктов, но для этого, как и ранее, вновь необходима соответствующая модификация понятия дерева вывода. Для доказательства полноты предпосылки также потребуется строгая формализация понятия пространства поиска вывода.

Правила вывода строятся в стиле *L.Bachmair* и *H.Ganzinger*[1]. Для исследуемых модальных систем вводятся необходимые понятия, доказываются вспомогательные утверждения, и теоремы полноты.

Теорема 5.2 (*полноты для системы* $\{KT^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}\}$). **П**усть **Φ** – формула *KT* и $\succ$ есть **Φ**-упорядочение, тогда (1) если какое-либо дерево вывода в системе $\{\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathbf{IP}}\}$ успешно, то формула **Φ** невыполнима и (2) если формула **Φ** невыполнима, то любое справедливое дерево вывода в $\{\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathbf{IP}}\}$ успешно.

Теорема 5.3 (*полноты для* $\{\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathrm{IP}}\}$ *с префиксной предпосылкой*). (1) Если любое дерево вывода в системе $\{\boldsymbol{KT}^{\boldsymbol{\Phi},\succ}{}_{\mathbf{IP}}\}$ с префиксной предпосылкой успешно, то формула **Φ** логики ***KT*** невыполнима. (2) Если формула **Φ** логики

---

[1] ***Bachmair L., Gansinger H.*** *Resolution theorem proving // Handbook of Automated reasoning. – Elsevier science and MIT Press. – 2000.*

KT невыполнима, то любое справедливое дерево вывода ***S*** в $\{\boldsymbol{KT}^{\Phi,\succ}_{\mathbf{IP}}\}$ с префиксной предпосылкой успешно.

Интересным свойством префиксной предпосылки является следующее. Предположим, что во время поиска опровержения мы получили секвенцию пути $\Delta = \Pi,\pi,\pi'$ такую, что $\pi \sqsubseteq \pi'$. Рассмотрим секвенцию путей $\Gamma = \Pi,\pi',\pi$. Эта секвенция префиксно-предпосылается секвенцией Δ, следовательно ее можно добавить в пространство поиска без нарушения полноты. Однако, если мы добавляем Γ к пространству поиска, то через сокращение мы можем вывести Π,π', которое предпосылает Δ. Таким образом, Π,π,π' можно заменить на более простую секвенцию Π,π', как только устанавливается, что π является префиксом π'.

Полученные результаты суммируются в *заключительном* параграфе. Необходимость сокращения пространств вывода достаточно очевидна. В общем случае, *даже для подклассов разрешимых предложений*, процедуры установления выводимости (особенно, в первопорядковых системах) являются ***NP*** (***P-space***) полными. Применение результатов этой и предшествующей глав *полезно* в практическом использовании, поскольку экономит вычислительные ресурсы. *Доступность* результатов для практического использования определяется тем, что все доказательства выполняются построением, то есть являются конструктивными. Вообще говоря, любое из приведенных доказательств выступает весьма детализированной спецификацией соответствующей программы.

3. ЗАКЛЮЧЕНИЕ

Представленные формализмы и стратегии естественно выступают основой организации машин логического вывода в системах управления знаниями, где существенны требования к используемым ресурсам, важно время реакции системы, а также задействованы различного рода модальности и личностные оценки.

Теория РДС-моделей обладает свойствами эффективной распознаваемости и эффективной разрешимости. Опираясь на механизм *схем* этой теории, можно ввести понятие «реализуемых предикатов» и на этой основе строить когнитивные системы, обладающие свойствами исчислений высших порядков, но свободные от их недостатков.

Сложностные характеристики теорий, представленных в четвертой и пятой главах, в общем случае, не хуже, чем у классического метода резолюций, в то время как описательные возможности модальных систем существенно расширяют язык классической логики. Важным также является

то, что предложенный подход исключает использование внелогических механизмов (подобных оператору усечения в Прологе).

Весьма успешным является комплексное использование результатов всех содержательных глав диссертации. Общая методология, опирающейся на нецелеориентированные методы поиска вывода, позволяет применять высокоэффективные подходы второй и третьей глав с сокращающими стратегиями и модальными языками четвертого и пятого разделов.

Диссертационное исследование определяет достаточное количество «точек роста» – новых направлений работы, которые могут оказаться полезными в областях распределенного управления, нейроинформатике, семантическом лингвистическом анализе, CASE-технологиях и так далее.

СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Новосельцев В.Б. Синтез рекурсивных программ в системе СПОРА Препринт – 43. Ленинград: ИТА АН СССР «Алгоритмы небесной механики», 1985. – 45 с.
2. Новосельцев В.Б. Структурные вычислительные модели // Синтез программ. – 1985. – С. 17-23.
3. Новосельцев В.Б., Калайда В.Т. Концепции построения программного окружения для решения задач оптики атмосферы // Журнал оптики атмосферы. – 1986. – №6. – С. 43-51.
4. Новосельцев В.Б. Синтез рекурсивных программ без развертки модели // Программное обеспечение ЭВМ. – 1987. – С. 54-59.
5. Zurenapel F., Novoseltsev V. Toward the automatic deductive proof of the recursive formula in special intuitional theories // Communications on Discrete Applied Systems. – 1990. – №3. – pp. 356-364.
6. Новосельцев В.Б. Некоторая формализация недетерминированных процессов // Параллельные вычисления. – 1991. – С. 37-41.
7. Новосельцев В.Б, Калайда В.Т., Шагисултанов В.Г. Разработка программного обеспечения для представления управляемых объектов в системах связи. Научно-технический отчет // ИОА СО РАН. – № К5–14/91–Гб. – Томск, 1991. – 84 с.
8. Novoseltsev V. Automatic inference engine for structured model theory // Applicationes de Logique. – 1991. – № 2. – 25 с.
9. Новосельцев В.Б., Бойцов С.Н. Иерархическая модель сложной динамической системы // Рукопись деп. ВИНИТИ. – 1996. – 28 с. – №1206-В1996.
10. Novoseltsev V. Computational models and recursive program synthesis // 5th International Symposium on Applied Logic. – Tallinn, 1996. – pp. 76-82.

11. Новосельцев В.Б., Нам Ю.Б. Формализм для представления и анализа сложных информационных комплексов // III Сибирский Конгресс по прикладной и индустриальной математике. – Томск, 1998. – С. 25-30.
12. Новосельцев В.Б., Бурлуцкий В.В. Логический вывод в дедуктивных системах // Вторая сибирская школа молодого ученого: т. 3, – 2001. – С. 27-41.
13. Новосельцев В.Б., Бурлуцкий В.В. Подход к реализации вывода в различных льюисовских системах // Логика и приложения. – 2001. – С. 61-65.
14. Новосельцев В.Б., Бурлуцкий В.В. Реализация обратного метода для модальной логики ***KT*** // Исследования по анализу и алгебре. – 2001. – С. 54-60.
15. Новосельцев В.Б., Тютерев В.В. Метод динамического наращивания узлов как способ построения нейронных сетей эффективного размера // Нейрокомпьютеры: разработка и применение. – 2001. – №2. – С. 82-88.
16. Новосельцев В.Б., Тютерев В.В. Автоматическое построение нейронных сетей методом эволюционного наращивания // Рукопись деп. ВИНИТИ, ТГУ. – 2001, №1944-В2001. – 26 с.
17. Новосельцев В.Б., Ларионов Д.С. Оболочка экспертной системы на основе нечетких рассуждений с прямым и обратным выводом // Математическое моделирование, ИММ РАН: т.14.– 2002. – №9 – С. 48-52.
18. Новосельцев В.Б., Романчук Е.А. Нечеткие множества в когнитивных системах // Труды ИСП РАН: т. 7. – 2004. – С. 55-60.
19. Новосельцев В.Б., Соколова В.В. Расширение алгебры Кодда операциями с рекурсивными объектами // Вестник ТГУ. – 2004. – №284. – С. 54-59.
20. Новосельцев В.Б., Колтун П.Н., Романчук Е.А. Возможность использования дескрипторных логик в системе управления знаниями // Вестник ТГУ. – 2004. – №284 – С. 17-22.
21. Novoseltsev V., Sokolova V. Expansion of Codd algebra by recursive definitions // Korea-Russia International Symposium on Science & Technology “KORUS 2004”. – Tomsk, 2004. – pp. 54-57.
22. Новосельцев В.Б., Малахов А.В. К вопросу о синтезе параллельных алгоритмов // Вестник ТГУ.– 2004. – №284. – С. 35-39.
23. Новосельцев В.Б., Романчук Е.А. О манипулировании знаниями с использованием нечетких множеств // Оптика атмосферы и океана. – 2005. – №4. – С. 867-871.
24. Novoseltsev V. Theorem proving in normal modal system of ***K***-family // 8th International Conference on Mathematics. – Spain, 2005. – pp. 82-88.
25. Новосельцев В.Б., Аксенов С.В. Организация и использование нейронных сетей. – Томск: НТЛ, 2006. – 160 с.
26. Новосельцев В.Б. Теория структурных функциональных моделей // Сибирский математический журнал. – 2006. – №5. Т. 47. – С. 1014-1030.

27. Новосельцев В.Б., Аксенов С.В. Повышение качества распознавания сцен нейронной сетью неокогнитрон // Известия Томского политехнического университета. – 2006. – №6. Т. 309. – С.150-160.
28. Новосельцев В.Б., Коваленко Д.А. Стратегия установления выводимости формул в структурных функциональных моделях // Известия Томского политехнического университета. – 2006. – №6. Т. 309. – С. 112-118.
29. Новосельцев В.Б., Копаница Г.Д. Нецелеориентированная стратегия вывода формул в модальных исчислениях // Известия Томского политехнического университета. – 2006. – №6. Т. 309. – С. 11-20.
30. Новосельцев В.Б., Соколова В.В. Обработка рекурсивных данных конечными автоматами // Известия Томского политехнического университета. – 2006. – №6. Т. 309. – С. 54-58.