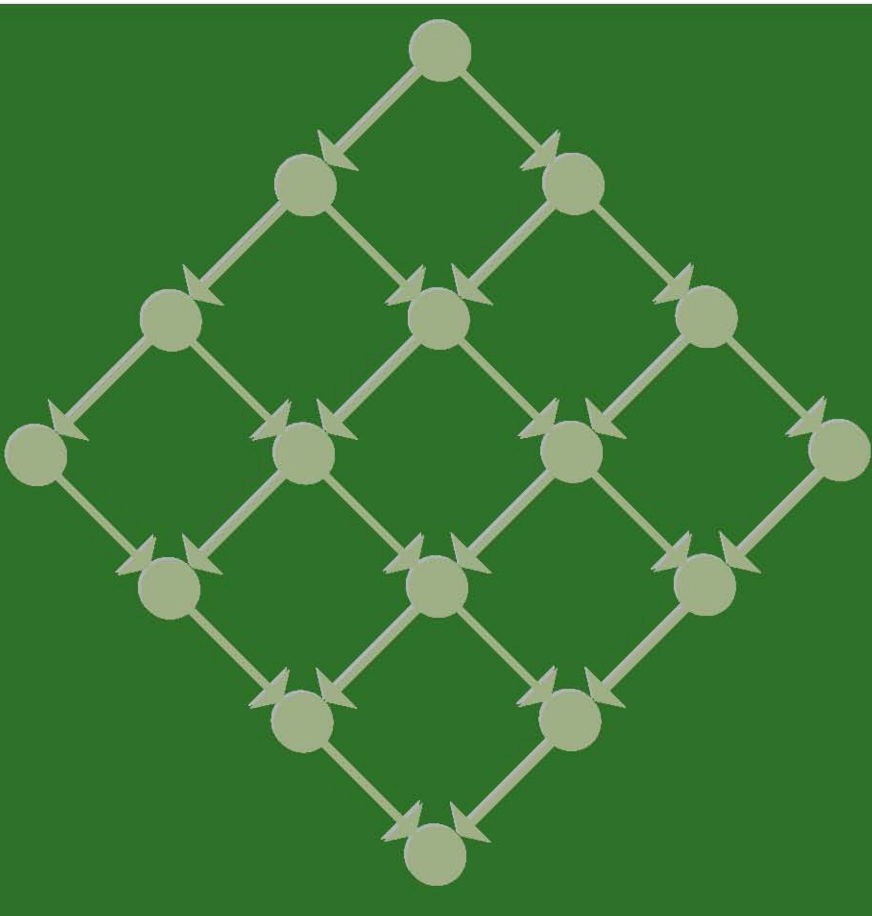
С. Д. Махортов

Математические основы искусственного интеллекта

теория LP-структур для построения и исследования моделей знаний продукционного типа



С. Д. Махортов

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА:

ТЕОРИЯ LP-СТРУКТУР ДЛЯ ПОСТРОЕНИЯ И ИССЛЕДОВАНИЯ МОДЕЛЕЙ ЗНАНИЙ ПРОДУКЦИОННОГО ТИПА

Под редакцией
В. А. Васенина

Москва
Издательство МЦНМО
2009

УДК 512.536.6:004.8
ББК 22.12+32.813
М36

Махортов С. Д.

М36 Математические основы искусственного интеллекта: теория LP-структур для построения и исследования моделей знаний продукционного типа / Под ред. В. А. Васенина. — М.: Издательство МЦНМО, 2009. — 304 с.

ISBN 987-5-94057-575-7

Излагается основанная на решетках алгебраическая теория, которая предназначена для моделирования и управления знаниями в интеллектуальных системах продукционного типа. Многие модели в информатике имеют продукционный характер, а структуры представления информации, как правило, являются иерархическими. Предложенная теория адекватно отражает вторичные продукционные связи в иерархических системах широкого спектра применения, а также обосновывает формальные исследования таких систем на предмет их эквивалентности, эквивалентных преобразований, верификации и оптимизации.

Описаны возможности применения теории LP-структур на примерах из различных областей информатики. Представлена интегрированная среда разработки продукционных экспертных систем, а также реализация в ее составе LP-структуры для верификации и оптимизации баз знаний. Приводятся результаты экспериментов, подтверждающие практическую значимость изложенной теории.

Книга предназначена для студентов, аспирантов и научных работников, занимающихся исследованиями в области алгебраических основ информатики и интеллектуальных систем.

УДК 512.536.6:004.8
ББК 22.12+32.813

ISBN 987-5-94057-575-7

ОТ РЕДАКТОРА

Информационные технологии на современном этапе развития общества проникают во все сферы деятельности человека, включая материальную и интеллектуальную, социально-культурную и политическую. Данный фактор порождает постоянно возрастающую зависимость общества от уровня надежности и эффективности программных решений, поддерживающих процессы его информатизации. К таковым в полной мере относятся сложно организованные и наукоемкие системы формального представления знаний, в первую очередь – широко распространенные на практике системы продукционного типа. Как следствие, оказывается весьма востребованным создание строгой математической базы, которая теоретически обосновывала бы корректность и надежность таких систем, а также возможности их оптимизации в автоматизированном режиме.

Настоящая книга посвящена решению указанной научной проблемы. В работе представлена созданная автором на основе выполненных им исследований теория LP-структур, описывающая новые эффективные методы для построения широкого спектра моделей знаний продукционного типа. Теория LP-структур предлагает общую методологию анализа продукционных систем в плане их эквивалентности, эквивалентных преобразований, верификации и оптимизации.

На основе рассмотрения особенностей продукционных систем в книге предложена обобщающая их новая алгебраическая модель. Данная модель базируется на иерархическом множестве (решетке) и содержит дополнительное бинарное отношение с набором специальных (продукционно-логических) свойств. Порождающий решетку частичный порядок отражает универсальные тавтологии и является фиксированным. Продукционное отношение определяется логическими связями конкретной предметной области и может подвергаться преобразованиям с целью её оптимизации по тем или иным критериям. Изучение такой модели показало, что аналогичной семантикой обладают и другие системы в информатике, которые ранее никогда не относились непосредственно к продукционным. В результате построена новая математическая теория, имеющая широкую область приложений в информатике.

В целях практического применения теория LP-структур развита до эффективных способов компьютерных представлений бинарных отношений на решетках. На основе этой теории разработаны новые методы обратного вывода. Представленная в книге программная реализация интегрированной среды разработки экспертных систем, созданные с ее помощью базы знаний, результаты их верификации и оптимизации демонстрируют работоспособность теории LP-структур. Построена эффективная система логического программирования, обладающая новыми автоматизированными возможностями.

С учетом изложенного есть все основания полагать, что представленная в настоящей книге теория LP-структур вносит весомый вклад в решение крупной, национального уровня научной проблемы создания математической базы для верификации, оценки надежности и оптимизации программного обеспечения автоматизированных систем управления знаниями. Технологические решения на ее основе обладают несомненными инновационными перспективами и практической значимостью для экономики страны. Теория LP-структур имеет возможности дальнейшего развития, а ее применения могут охватить более широкие области информатики.

Доктор физико-математических наук,
профессор МГУ имени М. В. Ломоносова
В. А. Васенин

ПРЕДИСЛОВИЕ

Современное развитие информационных технологий напоминает рекурсивную систему – технологии, технологии для технологий и так далее. Соответственно и процесс программирования давно поддерживается собственными технологическими средствами, которые, в свою очередь, реализуются в виде специальных программных систем. Объемы и сложность решаемых на данном направлении задач быстро растут, и естественной в этой связи выглядит потребность в автоматизации и переносе как можно большей части процесса производства программ на ЭВМ. Однако, чем глубже «рекурсия», чем сложнее процесс, тем выше ответственность разработчиков, риски ошибок и следующих за ними неблагоприятных ситуаций и различного рода потерь. Особое значение отмеченные факторы имеют для систем информатизации объектов критически важных инфраструктур, где допущенные «промахи» могут привести к чрезвычайным ситуациям, экологическим катастрофам, материальным и другим потерям государственного масштаба. В результате национально значимой становится проблема создания математической базы, с помощью которой можно было бы теоретически обосновывать корректность и надежность программного обеспечения, а также поддерживать процессы его автоматической оптимизации.

Книга посвящена одному из направлений решения обозначенной проблемы. Излагается алгебраическая теория LP-структур, предназначенная для формального построения и исследования широко распространенных на практике логических систем продукционного типа. Многие модели в информатике имеют продукционный характер, а структуры представления информации, как правило, являются иерархическими. Предложенная теория адекватно отражает вторичные продукционные связи в иерархических системах широкого спектра применения, а также обосновывает формальные исследования таких систем на предмет их эквивалентности, эквивалентных преобразований, верификации и оптимизации.

Описаны возможности применения теории LP-структур на примерах из различных областей информатики. Это относится к стандартным и расширенным продукционным экспертным системам, системам компьютерной алгебры, автоматического доказательства теорем, автоматизированного рефакторинга, автоматизации программирования и так далее. Описана интегрированная среда разработки продукционных экспертных систем, а также реализация и применение в ее составе LP-структуры для верификации и оптимизации баз знаний. Приводятся результаты экспериментов, подтверждающие практическую значимость изложенной теории.

Представленные в работе исследования носят в основном теоретический характер. Ее назначение состоит в определении подхода и построении класса алгебраических структур, позволяющего формулировать и успешно решать задачи исследования и оптимизации продукционно-логических систем. Формулируя специальные свойства бинарных отношений в LP-структурах, можно аналогичными методами получать адекватные алгебраические модели для применения в различных областях информатики, в том числе и таких, которые остались за рамками исследований настоящей работы.

В то же время изложенные в монографии теоретические результаты имеют хорошие перспективы практического применения. Каждая из описанных возможностей применения LP-структур может быть доведена до программной реализации – решения задач автоматизации эквивалентных преобразований, верификации и минимизации продукционно-логических систем.

Автор выражает глубокую признательность своим учителям – доценту В. Е. Калечицу, благодаря школе которого с конца 70-х годов считает себя вправе называться программистом, а также профессору В. П. Глушко, в 80-е годы существенно повлиявшему на формирование математического мировоззрения автора.

Автор благодарен руководителям и участникам ряда научных семинаров г. Москвы, чьи замечания и обсуждения в последние годы оказались весьма полезными для написания данной работы. Это, в частности, член-корреспондент РАН Л. Н. Королев, профессор Р. И. Подловченко, профессор С. А. Абрамов, профессор О. П. Кузнецов, доктор физико-математических наук И. Б. Бурдонов, доцент В. А. Захаров.

Особую благодарность хотелось бы выразить научному редактору книги профессору В. А. Васенину, сделавшему многочисленные конструктивные замечания по ее тексту.

ВВЕДЕНИЕ

Эффективным средством формального построения и исследования компьютерных программ, основанных на самых различных парадигмах и технологиях, являются алгебраические системы ([112, 179, 185] и другие). Это положение в полной мере относится и к логическому программированию, особенно в части представления теорий и знаний. Алгебраическим методам представления знаний посвящены работы [66, 78], а также монографии [92, 194].

Математическую основу для создания и исследования моделей знаний предоставляет алгебраическая логика. Ее начала были заложены в работах А. Линденбаума, А. Тарского, систематическое изложение дано в монографиях [38, 187]. Теория Линденбаума–Тарского рассматривает логику нулевого порядка как универсальную алгебру, операции которой соответствуют логическим связкам пропозиционального языка. Примеры алгебраизации исчисления предикатов первого порядка представляют полиадические алгебры Халмоша [37] и цилиндрические алгебры Хенкина–Тарского [41]. Обзор методов алгебраизации кванторов содержится в монографии [63].

Однако общая алгебраическая логика, расширяя возможности исследования самих логических теорий, существенно не облегчает их практического применения. В силу своей универсальности она не решает ряда важных частных проблем, связанных с широко распространенными на практике логическими системами продукционного типа. На этот факт указывается в книгах [180, 197]. К проблемам такого рода могут быть отнесены вопросы эквивалентности, эквивалентных преобразований, верификации продукционных и подобных им систем, рассматривавшиеся частными методами в работах [1, 27, 48, 52, 64, 186] и других исследованиях. Обзоры имеющихся подходов к верификации знаний содержатся в [34, 191].

Перечисленные вопросы играют существенную роль при создании и исследовании формальных методов работы со знаниями, а также определяют принципы построения программных средств автоматизации управления знаниями.

Особое место в ряду названных проблем занимает минимизация, поскольку в любой парадигме программирования действует золотое правило – избегать неоправданного дублирования кода или данных. В общей математической логике минимальная система аксиом называется базисом. Проблемы существования базисов допустимых правил для различных логик рассматривались В.В.Рыбаковым и его последователями [188-190]. Однако продукционные системы имеют особенности, дающие дополнительные возможности в плане минимизации.

Эквивалентные преобразования и минимизация множества *унифицированных* правил продукционных экспертных систем в некоторых работах изучались на основе пропозициональных хорновых функций (например, [7, 39, 40]). В статье [27] для исключения избыточности знаний используется логика первого порядка. В работе [30] рассмотрено несколько частных случаев упрощения множества правил на основе теории графов. Еще один путь минимизации продукционных систем может дать использование представления продукций сетями Петри [1] и решение задачи редукции сетей Петри [6, 47, 85].

В перечисленных работах нет строго обоснованного алгебраического подхода, универсального в рамках широкого спектра систем продукционного типа, который мог бы быть применен для решения задач эквивалентных преобразований, верификации и минимизации. Имеющиеся алгебраические исследования посвящены частным случаям продукционных систем либо другим их аспектам. В качестве примера можно привести связанную с теорией очевидности [76] алгебраическую теорию «демпстероидов» [35]. Она предназначена для вычислений результатов вывода в продукционных системах с функциями доверия [32].

В работе [74] расширенная продукционная система сводится к системе переписывания термов (СПТ). В семантике последней предлагается процедура обнаружения избыточных правил. Как это принято в системах переписывания термов, требуется «завершаемость» СПТ и, соответственно, исходного множества правил, иначе нет гарантии завершаемости процедуры. Этот интересный метод не охватывает продукционные системы, множества правил которых содержат циклы.

Возможности алгебраического исследования продукционно-логических систем содержит теория ультраоператоров А. В. Чечкина

[204]. В приложении к математической логике она предлагает рассматривать импликации в виде отдельного соответствия. Однако в открытой печати не представлены более подробные исследования ультраоператоров в данном направлении.

Интересная алгебраическая модель, позволяющая формализовать широкий круг логических задач, предложена Б. А. Куликом [128]. Введенная им алгебра кортежей представляет собой еще одну алгебраическую интерпретацию математической логики. В монографии [129] описываются также возможности применения изложенной теории к моделированию экспертных систем. Не вдаваясь в подробности, отметим, что опубликованные для алгебры кортежей результаты не связаны с решением перечисленных выше задач для продукционных систем.

Выше наряду с продукционными системами недаром были упомянуты также «подобные им» системы. В первую очередь, в контексте настоящей работы, конечно, имеются в виду собственно продукционные экспертные системы. Они остаются актуальными, что подтверждается значительным числом публикаций последних лет, посвященных как их теоретическим исследованиям, так и решению прикладных задач [5, 12–13, 44–45, 50–51, 55, 69, 80, 109]. База знаний одного из распространенных классов продукционных систем представляет собой совокупность правил вида «если ..., то ...», где в предпосылке и заключении фигурируют множества так называемых фактов. Эти множества независимо от правил также образуют иерархию как элементы булеана – множества подмножеств. Правила расширенной продукционной системы в предпосылке и заключении могут содержать пропозициональные формулы [15]. Такие формулы кроме правил связаны еще и иерархией в рамках соответствующей алгебры Линденбаума–Тарского. В подобном же виде могут храниться и математические знания – большинство теорем записывается в виде «дано ..., требуется доказать ...», где предпосылка и заключение являются формулами исчисления предикатов.

Широко применяемые в теории программирования условные системы переписывания термов [17, 46, 99] также основываются на правилах продукционного вида, связывающих пары элементов, которые принадлежат некоторой иерархии термов. Существуют и другие области информатики, на первый взгляд далекие от продукций, но интерпретируемые ими. В частности, элементы иерархии типов объектно-ориентированной программной системы [94] можно рассматривать как множество, на котором задано продукционное отношение агрегирования [199]. Даже в императивных программах просматриваются продукционные связи между состояниями данных до и после выполнения каждой операции.

С учетом изложенного выше можно констатировать, что актуальной является проблема создания алгебраической теории, которая бы адекватно отражала вторичные продукционные связи в различных иерархических системах широкого спектра применения, а также обосновывала автоматизированные формальные исследования таких систем в плане их эквивалентности, эквивалентных преобразований, верификации и оптимизации.

Основной целью настоящей книги является изложение такой теории, а также описание и демонстрация возможностей ее применения на примерах из различных прикладных областей.

Научная оригинальность монографии заключается в следующих положениях.

• Для автоматизированной разработки и исследования систем продукционного типа предложен новый подход, выраженный в создании основанной на решетках алгебраической теории LP-структур (Lattice-Production Structure). Предполагается, что информация о некоторой предметной области может быть формально представлена в виде решетки.Описание методов использования решеток для представления знаний можно найти в [66, 78, 194]. Основная идея теории LP-структур состоит в моделировании продукционных связей (совокупности правил) дополнительным бинарным отношением с заданными свойствами (рефлексивность, транзитивность и некоторые другие свойства, зависящие от конкретной модели). При этом определяющее решетку исходное отношение частичного порядка отражает универсальные тавтологии и является фиксированным. Второе отношение порождается логическими связями конкретной предметной области и может подвергаться эквивалентным преобразованиям.

• Введено и обосновано понятие эквивалентности продукционно-логических систем на основе их логического замыкания. Доказаны возможности автоматических локально-эквивалентных преобразований LP-структур и соответственно моделируемых ими продукционных систем.

• Введено новое понятие продукционно-логического уравнения и обоснован способ его решения, что в применении соответствует *полному* обратному выводу. Концепция уравнений может быть также применена для верификации знаний. Ранее интересные классы логических уравнений рассматривались в монографиях [111, 133], однако представленные в них уравнения имеют отличную от систем продукций природу. На нечетких бинарных отношениях основаны реляционные уравнения, рассматривавшиеся в [16] и ряде других работ. Основные трудности исследования здесь порождаются нечеткостью отношений, и поэтому процесс решения соответствует всего лишь единственному шагу обратного вывода.

• В монографии также доказано существование и получен эффективный способ построения логической редукции LP-структур. В приложениях он означает минимизацию соответствующих баз знаний, то есть построение эквивалентной продукционной системы с минимальным набором правил.

• Новым является распространение единого алгебраического подхода на достаточно широкий спектр различных систем: стандартные и расширенные продукционные экспертные системы; формальные системы математических знаний; условные системы переписывания термов; иерархии типов в объектно-ориентированном программировании. В частности, новым является введенный и использованный в работе тезис о продукционной семантике иерархии типов с отношением агрегирования. В результате на базе LP-структур обоснованы автоматизированные решения некоторых важных задач верификации типов и рефакторинга. Показана возможность применения продукционно-логических структур к новым исследованиям свойств императивных алгоритмов.

• В качестве примера для иллюстрации приложения LP-структур первого порядка в работе изложены элементы теории неклассических псевдодифференциальных операторов. Они содержат новые результаты в соответствующей области.

• Сформулирована новая концепция трехмерной структуры расширяемой программной системы, которая в дополнение к актуальной ранее двумерной модели [104, 200] содержит набор взаимосвязанных уровней программирования, от системного до пользовательского, завершающийся на верхнем уровне продукционной экспертной системой.

• Разработаны и реализованы оригинальные эффективные методы компьютерного представления основанных на решетках алгебраических структур.

• Предложены и реализованы новые методы обратного вывода и верификации для систем продукционного типа, базирующиеся на решении логических уравнений. Концепция LP-вывода направлена на минимизацию количества запросов к внешнему источнику. Запросы по возможности отправляются только для тех фактов, которые необходимы при выводе. Отрицательный ответ на единственный запрос исключает все последующие запросы об элементах целого множества. Кроме того, при LP-выводе предпочтение отдается тестированию множества фактов минимальной мощности.

Данные идеи не пересекаются с известными методами быстрого вывода, а дополняют их. Во-первых, алгоритмы RETE [25] и TREAT [58], базирующиеся на специальных сетевых представлениях множеств правил, изначально разрабатывались для стратегии *прямого* вывода и использовались в продукционных системах с прямым выводом (например, OPS5 [24], CLIPS [42]). Алгоритм LEAPS [59] осуществляет компиляцию в язык C множества правил той же продукционной системы OPS5. В последующих исследованиях наиболее популярный алгоритм RETE адаптировался для смешанного (двунаправленного) вывода [42, 49]. Изложенная в настоящей книге концепция уравнений предназначена для исключительно *обратного* вывода и не требует для своей реализации громоздких динамических структур данных, свойственных указанным выше алгоритмам, в случаях, когда нет никакой потребности в прямом (и соответственно смешанном) выводе. Во-вторых, ничто не мешает адаптировать имеющиеся быстрые алгоритмы обратного вывода для нахождения рассмотренных в работе решений продукционно-логических уравнений. Этот подход может оказаться интересным направлением развития теории LP-структур.

• Реализована интегрированная среда разработки и выполнения продукционно-логических систем, обладающая новыми возможностями исследования и оптимизации баз знаний.

Новая теория LP-структур занимает собственную нишу, однако при этом она соприкасается с некоторыми другими исследованиями.

Отметим в частности, что бирешетки [28] также предполагают наличие двух отношений на общем множестве, однако в прочих аспектах теория LP-структур имеет с ними мало общего, как с точки зрения формального определения, так и в плане возможных применений. В ряде работ (см. [14] и библиографию в ней) рассматриваются общетеоретические вопросы о свойствах бинарных отношений на частично упорядоченных множествах (в том числе и решетках), такие как монотонность, неподвижные (рефлексивные) точки, рекурсии. Однако эти исследования не учитывают специфики моделей продукционных систем.

Задача логического вывода на LP-структурах перекликается с проблемой нахождения функциональных зависимостей в реляционных базах данных (см., например, [100]). При выводе функциональных зависимостей в базе данных используются дедуктивные правила Армстронга (они впервые введены в работе [4]), применяемые к элементам булеана. Однако в этой теории из «решеточных» операций используется лишь операция объединения, а набор правил Армстронга существенно беднее множества правил вывода в LP-структурах. Вполне возможно, что исследование реляционных баз данных может стать одной из новых областей применения теории LP-структур. Имеются также определенные перспективы использования подобных LP-структурам алгебраических систем для моделирования некоторых классов полуструктурированных данных (см., например, [88, 96]) с целью исследования и оптимизации запросов.

Близким к теории LP-структур может считаться FCA – формальный концептуальный анализ [26, 127], имеющий широкую область применения в исследованиях двумерных данных с семантикой «объекты-признаки». Он также основан на решетках и рассматривает бинарные отношения между множествами. Однако LP-структуры и FCA имеют существенные отличия. Результаты анализа публикаций свидетельствуют, что общих приложений у этих двух теорий практически нет, несмотря на иногда похожие формулировки решаемых в их рамках задач. Например, минимальный базис импликаций в FCA перекликается с логической редукцией LP-структуры. С помощью этих подходов ставятся и решаются задачи, соответствующие различным моделям в информатике. В частности, данный факт подтверждают представленные далее возможности применения LP-структур в объектно-ориентированном программировании.

ОБЗОР СОДЕРЖАНИЯ

Книга состоит из введения, шести глав, двух приложений и библиографии. Работа разбита на разделы и подразделы (пункты, сокращенно – п.). В пределах каждого пункта формулы, определения и так далее нумеруются автономно натуральными числами, начиная с 1. При ссылках на текущий раздел указываются только эти номера. При ссылках на другие разделы в качестве префикса добавляется полный номер раздела. Например, лемма 5 – пятая лемма текущего пункта; теорема 3.1.2 – вторая теорема первого пункта третьей главы.

Глава 1 по отношению к основному содержанию книги носит вводный характер. В этой главе перечисляются задачи из различных областей информатики, описание которых может быть сведено к системам продукционного типа, моделируемым LP-структурами. Одна из основных целей данной главы – сделать более ясными как происхождение теории LP-структур, так и общие возможности ее применения. Материал главы 1 соответствует работам [115–119, 146, 149, 158, 162, 177].

Начальный раздел главы посвящен сформулированной автором новой концепции трехмерной структуры расширяемой программной системы. Эта концепция развивает двумерную модель, описанную в работе [200] и впоследствии в [104]. Согласно двумерной модели, каждое расширение функциональности программы можно рассматривать как добавление определенного компонента (вертикального слоя), относящегося к некоторым ранее сформированным образованиям (горизонтальным слоям). Изъятие вертикального слоя не приводит к «катастрофическим» последствиям для программы, а лишь обедняет ее функциональность. Каждый же горизонтальный слой программы является ее неотъемлемой, системообразующей частью.

Анализ особенностей современных программных систем приводит к идее о целесообразности ввода в модель описания структуры программ ее третьего измерения, которое содержало бы параллельные уровни ее разработки. Согласно данной концепции, программная система разрабатывается сразу на нескольких взаимосвязанных уровнях, от нижнего системного уровня до верхнего пользовательского. При этом самый верхний уровень разработки содержит интеллектуальную часть для поддержки пользовательского программирования («настройка» программы), которая может быть представлена в виде специальной продукционной системы. Изложенная концепция изначально стала одной из отправных посылок для введения изучаемых в работе LP-структур.

Далее в главе 1 вводится основная терминология, связанная с бинарными отношениями, решетками, LP-структурами и продукционными системами. Под LP-структурой будем подразумевать алгебраическую систему, представляющую собой решетку с заданным на ней бинарным отношением, обладающим некоторыми продукционно-логическими свойствами.

В последующих разделах главы 1 в качестве возможных приложений LP-структур рассматриваются несколько видов продукционных систем. Они различаются используемой формой правила, точнее, выражений для его предпосылки и заключения. В каждом случае продукционную систему можно формально представить LP-структурой на основе решетки, элементами которой являются предпосылки и заключения правил.

Глава 2 содержит основные положения общей теории LP-структур. Теория названа общей, поскольку результаты последующих глав работы основываются на тех же положениях, развивая их в том или ином специальном направлении. Изложение соответствует работам [147–152].

Первоначально формулируется частная конечная теоретико-множественная модель продукционной системы, без использования аппарата решеток. Вводится ряд математических понятий, базовым среди которых является минимальное порождающее множество. Затем показывается, как эта теория может быть применена для исследования свойств продукционных систем и разработки связанных с ними алгоритмов. Использование предлагаемого подхода позволяет

- математически обосновывать возможности эквивалентных преобразований баз знаний;
- оптимизировать и исследовать алгоритмы обратного вывода;
- строго формулировать критерии корректности баз знаний, а также разрабатывать методы их верификации.

Далее во второй главе работы определяется понятие LP-структуры, которое обобщает рассмотренную ранее теоретико-множественную модель продукционной системы и выводит ее на более абстрактный уровень. Это достигается использованием решетки конечных множеств в качестве основы алгебраической системы. На решетке вводятся бинарные отношения, содержащие семантику продукционно-логического вывода. Доказывается существование логического замыкания произвольного бинарного отношения на решетке, что позволяет определить понятие эквивалентного отношения.

На базе этого понятия рассматриваются вопросы эквивалентных преобразований LP-структур. Доказано, что локальные эквивалентные преобразования приводят к эквивалентной общей структуре.

Исследуются также вопросы, связанные с минимизацией LP-структур. Минимальность понимается в смысле частично упорядоченных множеств, а именно – когда исключение из минимального отношения хотя бы одной пары приведет к неэквивалентному отношению.

С указанной целью предварительно рассматривается вопрос о том, не является ли *логическое* замыкание данного отношения *транзитивным* замыканием некоторого другого отношения. Полученный положительный ответ на этот вопрос оказывается полезным для разработки эффективных алгоритмов построения логического замыкания и редукции. Он позволяет свести исследование вопросов о нахождении логического замыкания и редукции отношений к рассмотрению соответствующих свойств транзитивных отношений.

В заключительном разделе главы 2 вводится и исследуется связанный с LP-структурами специальный класс продукционно-логических уравнений. Рассматриваются вопросы о разрешимости и количестве решений этих уравнений, а также методы их решения. Предложенный метод решения основан на понятии структурного расслоения бинарного отношения и допускает широкое применение параллелизма при компьютерной реализации.

Нахождение решения продукционно-логического уравнения эквивалентно обратному логическому выводу на решетке. Использование метода решения таких уравнений позволяет при выводе минимизировать обращения к базе данных либо к интерактивному пользователю. Суть такой минимизации: если получено решение уравнения и становится известно об отсутствии в базе данных хотя бы одной его точки, нет смысла проверять остальные точки.

В главе 3 определяются LP-структуры на основе решеток, обладающих свойством полноты. Изложение соответствует работам [165–169, 176]. Вводятся бинарные отношения, содержащие семантику продукционно-логического вывода со свойством транзитивной завершаемости. Приводится обоснование необходимости введения в модель ограничений, соответствующих этому свойству. В ряде алгебраических систем подобное свойство (отсутствие неограниченных транзитивных последовательностей) называется нетеровостью (см. [86] и библиографию в ней). Как известно, проблема нетеровости в общем случае алгоритмически неразрешима. Имеются методы доказательства нетеровости транзитивных отношений в системах переписывания термов, основанные на упорядочении. Ссылки на соответствующие работы содержатся в [17, 72, 131]. Условие нетеровости LP-структуры слабее аналогичного свойства систем переписывания термов, так как в отличие от них допускает циклы.

Проведенные в главе 3 исследования обобщают основную часть материала главы 2. Они предназначены для моделирования продукционных систем, правила которых содержат бесконечные множества фактов. Такие правила, в частности, могут порождаться продукционными системами, допускающими использование в правилах переменных [20].

Некоторые определения по форме выглядят подобными соответствующим определениям главы 2, однако теперь их содержание связано с *полными* решетками, и это обстоятельство усложняет как смысл данных понятий, так и доказательство их свойств. Получены результаты, полностью соответствующие изложенным в основных разделах главы 2, но относящиеся к нетеровым продукционно-логическим отношениям на полных решетках.

Заметим, что в исследованиях, которые представлены в главах 2–3, не применяются рекурсивные подходы. Это обстоятельство усложняет математические выкладки, однако предоставляет больше возможностей для использования параллельных вычислений. Параллельным методам в продукционных системах посвящены, например, работы [33, 62, 73]. Эти методы могут быть применены на более абстрактном уровне при компьютерной реализации LP структур.

Глава 4 настоящей монографии посвящена развитию представленной ранее теории LP-структур применительно к некоторым расширенным моделям продукционных систем. В связи с усложнением механизмов продукционно-логического вывода, при определении соответствующих понятий и при доказательстве их свойств в данной главе используется рекурсивный подход, аналогично принятому в классическом пропозициональном исчислении [121, 203]. Изложение соответствует работам [54, 156, 159, 160–161, 163–164].

Вначале определяется и исследуется LP-структура, логика которой расширена до полного набора логических связок пропозиционального языка – импликации, конъюнкции, дизъюнкции и отрицания. Отсюда происходит название такой алгебраической системы – «LP-структура нулевого порядка». В качестве ее основы используется булева решетка [93, 187]. Изучены следующие основные вопросы: существование LP-замыкания; его архитектура; эквивалентные преобразования. Доказана теорема о существовании логической редукции произвольного отношения и описан процесс ее построения, один из этапов которого связан с нахождением транзитивной редукции некоторого «большего» отношения.

Далее в главе 4 определяются расширенные LP-структуры на полных булевых решетках. Наряду с отображением набора логических связок пропозиционального исчисления, в данных алгебраических системах определены также бесконечно-местные операции пересечения и объединения. Согласно подходу [187], такие операции для модели продукционной логики реализуют кванторы общности и существования. Как и в главе 3, переход к полным решеткам привел к необходимости введения дополнительных ограничений модели, связанных с аналогом свойства нетеровости (завершаемости вывода).

Рассматриваются вопросы, связанные с существованием логического замыкания, его архитектурой, эквивалентными преобразованиями. Доказаны новые теоремы об архитектуре логического замыкания и о существовании логической редукции рассматриваемой LP-структуры. Полученные результаты соответствуют предыдущему разделу, однако относятся к нетеровым продукционно-логическим отношениям на полных булевых решетках. Они могут быть применены для исследования и автоматической оптимизации расширенных баз знаний продукционного типа. К таковым, в частности, относятся базы знаний систем компьютерной алгебры.

Далее рассматривается новая модель – модель эквациональной LP-структуры. Такая структура возникает в качестве алгебраической интерпретации условной эквациональной теории и соответствующей условной системы переписывания термов. Эти системы служат математической основой многих исследований в различных областях теории программирования. С их помощью, в частности, решаются задачи символьного упрощения алгебраических выражений [9], автоматического доказательства теорем [43], верификации программ [99] и другие.

Определяется основанная на решетках алгебраическая модель условной эквациональной теории, учитывающая возможные связи между термами, которые обусловлены применениями функций и подстановок. Вводятся алгебраические аналоги аксиом и правил условной эквациональной дедукции, соответствующие известным работам [46].

Кроме построенной модели, материалы данного раздела содержат исследование стандартного для настоящей монграфии круга вопросов: существование логического замыкания эквациональных LP-структур; возможности локально-эквивалентных преобразований; исследование архитектуры логического замыкания; существование и построение логической редукции бинарного отношения.

В заключительной части раздела на небольшом примере иллюстрируется процесс минимизации множества правил условной эквациональной теории. Подводятся итоги и указываются перспективы дальнейших исследований эквациональных LP-структур.

Применение рассмотренных выше моделей LP-структур ограничено системами с *монотонным* выводом, то есть таких систем продукционного типа, логический вывод в которых означает монотонное накопление знаний. Однако существуют востребованные на практике системы искусственного интеллекта, которые предполагают не только накопление, но и модификацию получаемых знаний. Различные вопросы, связанные с немонотонной логикой, исследовались в ряде работ (например, [90, 95, 194]). В главе 5 рассматриваются две новые задачи, формализация и исследование которых могут быть осуществлены на основе LP-структур с немонотонным выводом. Изложение материала соответствует работам [153–155, 168, 171, 173–175].

Отношения обобщения и агрегации в объектно-ориентированных программных системах [94, 199] обладают семантикой продукционного вывода. Это обстоятельство навело на мысль о возможности распространения теории LP-структур на подобные системы. В первой части главы 5 вводится LP-структура, содержащая семантику продукционно-логического вывода на иерархии типов в объектно-ориентированной системе с дополнительным отношением и набором ограничений монотонности. Исследуются стандартные свойства таких структур – замкнутость, возможность эквивалентных преобразований, существование логической редукции. Показано, как изложенная теория может быть применена для верификации и автоматической оптимизации иерархий типов, в частности, при рефакторинге – модернизации устаревшего кода [198]. Обзор современных методов рефакторинга содержится в [56]. Одним из важных направлений в этом плане является устранение дублирования кода путем «подъема» общих атрибутов по иерархии типов. Такая задача решается автоматически при построении логической редукции LP-структуры.

Решение родственных задач алгебраическими методами (формального концептуального анализа – [26]) предлагается в [31]. В этой работе элементам некоторого множества классов требуется в определенном смысле оптимальным образом назначить наборы атрибутов – элементов другого независимого множества. В соответствии с выбранными назначениями формируется иерархия классов. В постановке, которая рассматривается в настоящей работе, в отличие от [31], атрибуты сами относятся к исследуемой иерархии классов (типов). Это обстоятельство усложняет решение задачи и вряд ли оставляет возможность непосредственного применения формального концептуального анализа.

Алгебраическую интерпретацию типов данных в виде решеток рассматривал Dana S. Scott [75]. Мощный аппарат исследования иерархий типов представляют описанные в [21–22] многоуровневые упорядоченно-сортные алгебры. Однако, как справедливо замечено в [95], разработка этих алгебр еще не завершена. В открытой печати пока не представлены исследования, посвященные редукции многоуровневых упорядоченно-сортных алгебр, которая предусматривала бы указанное выше автоматическое устранение дублирования кода.

Еще одна немонотонная LP-структура исследуется во второй части главы 5. Она связана с анализом логических свойств и с преобразованиями императивных алгоритмов. Вводится и исследуется LP-структура, семантика которой моделирует постулат о том, что во время работы императивной программы информация не только накапливается, но и часто замещается. Используемые в программе переменные с их значениями можно считать фактами некоторой базы знаний. Тогда инициализация любой переменной может рассматриваться как добавление нового факта к базе знаний. Присвоение же инициализированной переменной нового значения можно интерпретировать как модификацию имеющегося факта. Рассматривается пример записи императивной программы в виде набора декларативных продукций.

Классическое понятие решетки оказывается недостаточно выразительным для моделирования немонотонного продукционно-логического вывода с описанной выше семантикой. В связи с этим обстоятельством вводятся некоммутативные решетки как обобщение классических решеток. В новых решетках операция объединения выполняется с частичным замещением одного из операндов. Для построения соответствующих LP-структур рассматриваются логические бинарные отношения на этих решетках.Основным результатом здесь является теорема об ассоциативности операции некоммутативного объединения. Доказана также теорема о существовании логического замыкания отношений на некоммутативных решетках.

Материалы данного раздела закладывают методологическую основу для еще одного подхода к логической формализации императивных программ. В частности, рассмотрен пример приложения теории некоммутативных LP-структур для исследования логических свойств императивных алгоритмов. На основе LP-структур сопоставляются возможности императивного и логического программирования.

Обзор других, менее абстрактных подходов, сочетающих различные парадигмы программирования, можно найти в работе [114]. В частности, вопросам интеграции логического и императивного программирования посвящены работы [10–11, 53].

При разработке математических теорий иногда недостаточно внимания уделяется их прикладным аспектам. В результате существует опасность того, что теория останется лишь на бумаге, а ее применение, по крайней мере, на текущем этапе, будет затруднено отсутствием в должном объеме интеллектуальных или технических ресурсов. Основная цель главы 6 – продемонстрировать работоспособность теории LP-структур. С этой целью описывается интегрированная среда разработки продукционных экспертных систем, а также реализация и применение в ее составе LP-структуры для автоматизированной верификации и оптимизации баз знаний. В заключительной части главы описывается применение теории LP-структур для алгебраической формализации математических знаний. Некоторая часть материала главы 6 содержится в работах [170, 172].

Предварительно приводятся некоторые общие принципы компьютерной реализации положений п.п. 2.3–2.6, не столь значимые для теоретического исследования, однако имеющие существенное значение на практике. Рассматриваются вопросы кодирования LP-структур, опирающиеся на методы представления решеток битовыми векторами. Оригинальными и эффективными являются идеи о настраиваемой «клиентом» структуре битового вектора, смежном хранении пар ссылок, представлении канонического отношения на решетке в виде вектора множеств. Затем описываются особенности и интерфейс объектно-ориентированного класса LPStructure, разработанного автором и реализованного на языке C++ с использованием библиотеки STL. Класс LPStructure инкапсулирует наиболее важные свойства и методы описанных в главе 2 стандартных LP-структур, включая нахождение логической редукции и решение продукционно-логических уравнений.

В последующем разделе рассматриваются особенности архитектуры и основные возможности пакета программ LPExpert (реализованного на Delphi), предназначенного для разработки и эксплуатации продукционных экспертных систем. Данный пакет первоначально создавался на основе идей книги B. Sowyer – D. Foster [79], с существенными их усовершенствованиями в синтаксисе правил и программной реализации. Впоследствии к нему была добавлена динамическая библиотека – интерфейс класса LPStructure, что значительно расширило возможности пакета в плане эффективности создания баз знаний. К основным преимуществам данной системы логического программирования можно отнести следующие возможности:

• выявление избыточных правил в базах знаний,

• проверка ее непротиворечивости,

• новые методы обратного логического вывода (релевантный и кластерно-релевантный LP-выводы), основанные на решении продукционно-логических уравнений,

• исследование образов и прообразов подмножеств фактов.

В п. 6.4.6 указанные возможности экспериментально демонстрируются на ряде примеров. Один из таковых (база знаний «Здоровье») взят из книги [79]. Выбор данного примера был сделан в силу его популярности в некоторых других исследованиях по экспертным системам (например, [29]). Для него, в частности, в данной работе получены следующие новые практические результаты:

• выявлено 3 избыточных правила (в работе [29] отмечено лишь одно таковое);

• показано, что LP-вывод по сравнению с обычным обратным выводом дает аналогичные результаты, однако пользователю при этом задается меньше вопросов (в тестовой экспертизе – на 20 %);

• доказана противоречивость данной базы знаний, что свидетельствует о нецелесообразности ее практического применения.

Кроме подробного исследования базы знаний «Здоровье», в п. 6.4.6 проведены также тесты пакета LPExpert на примерах больших баз знаний промышленного назначения, а также обучающих систем. В результате получены дополнительные свидетельства преимуществ LP-вывода, а именно – снижение количества внешних запросов на 15–20 % и увеличение числа выдаваемых результатов до 60 %.

В заключительном разделе главы 6 демонстрируется еще одно подтверждение применимости теории LP-структур – описывается алгебраическая формализация математических знаний LP-структурой первого порядка. В качестве примера таких знаний выбрана теория весовых псевдодифференциальных операторов и некоторые полученные с ее помощью новые результаты. Поскольку данная теория относится к области дифференциальных уравнений, необходимая для этого раздела ее часть в полном объеме приводится лишь в приложении B. В основном тексте раздела используются формулировки теорем со ссылками на приложение B. Построена LP-структура первого порядка – элементы соответствующей полной решетки Линденбаума–Тарского и пары элементов логического отношения, которые в виде записанных продукционных правил формализуют указанные математические знания.

Приложение A содержит фрагменты тестовых баз знаний, использовавшихся в работе для демонстрации применений теории LP-структур и пакета LPExpert. В п. A.1 приведена часть указанной выше базы знаний «Здоровье», записанная в расширенном синтаксисе пакета LPExpert. Раздел A.2 содержит фрагмент текста базы знаний промышленного уровня «Электрики», разработанной в рамках исследований по интеллектуализации систем управления охраной объектов [91, 113]. В разделе A.3 представлена обучающая база знаний «Закон распределения», созданная в 2008 г. студентами кафедры математического обеспечения ЭВМ Воронежского государственного университета по заданию автора настоящей монографии. Полные тексты баз знаний и реализованные автором компьютерные программы можно скачать с его личной web-страницы http://expert.vrn.ru/SD/.

В Приложении B приведены элементы разработанной автором теории весовых псевдодифференциальных операторов (ПДО). Более полное ее изложение можно найти в работах [102, 137–145]. Материал Приложения B соответствует работе [157]. Кроме результатов, относящихся к области неклассических уравнений с частными производными, данное исследование служит иллюстрацией возможностей применения LP-структур к моделированию математических знаний.

Рассматриваются специальные классы ПДО, не относящиеся к главному типу [108]. Эти операторы имеют множество вырождения ненулевой меры, обусловленное наличием в их символах весового множителя α. Предлагаемый метод исследования вырождающихся ПДО основан на их сравнении с другими ПДО, построенными по введенному В. П. Глушко [101] весовому преобразованию Фурье F_α. Преобразование «поглощает» множитель α, в результате получаются операторы с символами из хорошо известного класса. Для построения этой теории требуются также весовые пространства типа Соболева–Слободецкого.

Глава 1
ЗАДАЧИ, ПРИВОДЯЩИЕ К LP-СТРУКТУРАМ

Как отмечалось во введении, настоящая работа посвящена алгебраическим системам (LP-структурам), формализующим продукционно-логический вывод на основе теории решеток и бинарных отношений. Результаты исследований показали применимость таких теоретических положений к созданию программного обеспечения процессов представления и использования знаний на базе моделей продукционного типа. Цель настоящей главы – описать возможности таких приложений. Перечисляются задачи в различных областях информатики, описание которых может быть сведено к продукционным системам, моделируемым LP-структурами. К таковым относятся: верификация и оптимизация баз знаний экспертных продукционных систем; представление математических знаний; упрощение множеств правил условных систем переписывания термов; автоматизация некоторых методов рефакторинга; исследование свойств императивных алгоритмов. Соответственно решение перечисленных задач может осуществляться на основе LP-структур или их модификаций

Раздел 1.1 посвящен обсуждению сформулированной автором концепции многоуровневой разработки программных систем. Согласно этой концепции, верхний уровень разработки программ содержит интеллектуальную надстройку для пользовательского программирования на основе продукционных систем. Это положение стало отправной точкой для введения рассматриваемых в данной работе LP-структур. В последующих разделах главы анализируются возможные применения LP-структур для описания различных систем продукционного типа.

1.1. О многоуровневом программировании

В данном разделе формулируются концептуальные положения трехмерной структуры расширяемой программной системы. В качестве содержания третьего измерения предлагается классификация параллельных уровней разработки программы. Приводится пример реализации программных средств поддержки указанной технологии.

Общеизвестно, что разработка любого более или менее серьезного программного комплекса связана с применением какой-либо одной или нескольких технологий программирования. Это обусловлено постоянно возрастающими сложностью и объемами решаемых в ходе такой разработки задач. При этом важнейшее место в процессе использования информационных технологий занимают характеризующие и формулирующие их концептуальные средства. Они определяют стиль, а также методы проектирования и разработки программного обеспечения (ПО). Важной особенностью настоящего времени является появление таких областей применения программных систем, для которых сопровождение ПО по длительности и ресурсозатратам сопоставимо с его разработкой. Этот факт означает, что никогда не наступает момент, когда «программа окончательно готова». К такого рода областям относится, например, автоматизация финансово-хозяйственной деятельности предприятий, где непрерывные и существенные изменения в законодательстве постоянно держат эксплуатируемое ПО в стадии частичной разработки. Для подобных программных систем особенное значение приобретает не только определение последовательности создания отдельных его частей, но и вопросы о том, разработчики какого уровня и на каких этапах занимаются их созданием. Аналогичные вопросы связаны также с разработкой ПО сложно организованных автоматизированных систем управления критически важными объектами [122–123].

Несмотря на эволюцию парадигм проектирования и написания кода от модульного к объектно-ориентированному программированию [94], по мнению автора, по-прежнему актуальным является представление о двумерности структуры развивающейся программы [104, 200]. Согласно такому представлению, каждое расширение функциональности программы можно рассматривать как добавление определенного компонента (вертикального слоя), относящегося к некоторым уже сформированным образованиям (горизонтальным слоям). Изъятие вертикального слоя при этом не приводит к «катастрофическим» последствиям для программы, а лишь обедняет ее функциональность. Каждый же горизонтальный слой программы является ее неотъемлемой, системообразующей частью. Даже, если допустить, что программу удалось спроектировать в виде единственного объекта некоторого класса, то и в этом случае его публичные свойства и методы, непосредственно экспортирующие функциональность объекта, можно считать вертикальными слоями, а скрытые свойства и методы, опосредованно реализующие функциональность, – составляющими горизонтальные слои. Различие точек зрения на такое представление состоит лишь в том, какие слои являются первичными – горизонтальные или вертикальные.

Однако математик, привыкший к абстрактному обобщению, задался бы вопросом: «почему только два измерения?». В настоящем разделе делается попытка ввести третье измерение в модель структуры расширяемой программной системы.

В программировании давно существует понятие уровня разработки. Так множество существующих языков программирования всегда разделялось по их уровню. Представим на самом нижнем уровне такой иерархии машинный язык, а на самом верхнем – разговорный человеческий. Тогда какой-либо язык программирования относится к низкому или высокому уровню в соответствии с его относительным расположением в указанной иерархии (Ассемблер – язык низкого уровня, Паскаль – высокого). Заметим, что повышение уровня при этом, как правило, облегчает разработку программы, однако сужает возможности. Немного обобщим это понятие. Будем считать, что вся программная система разрабатывается сразу на нескольких уровнях, каждый из которых занимает свое место в представленной выше иерархии. Можно, например, выделить три таких уровня (в порядке повышения): системный, прикладной, пользовательский.

Системный уровень разработки программного комплекса (не путать с операционной системой, расположенной еще ниже, однако в иерархии это самый близкий к ОС уровень) – это базовый уровень, который содержит средства поддержки технологии разработки и создается наиболее квалифицированными программистами, возможно, незнакомыми с конечной областью применения всего программного комплекса. Программные объекты этого уровня могут использоваться для создания не только одной, но и целого класса систем, основанных на данной технологии.

Прикладной уровень занимает позицию выше и содержит в основном реализацию решения конкретных задач на уровне программиста. Специалисты, создающие этот уровень, могут быть менее квалифицированными в программировании, однако должны разбираться в предметной области, для которой создается программная система.

Пользовательский уровень – самый высокий в смысле приближения к уровню мышления обычного пользователя. Разработку на этом уровне часто называют настройкой, однако ее сложность и объемы порой не уступают разработке, особенно это касается упомянутых выше систем автоматизации управления сложно организованными объектами. Специалисты этого уровня могут быть экспертами в предметной области, но иметь лишь общее представление о низкоуровневом программировании.

Приведенная классификация является наиболее общей, идеализированной и может уточняться по мере развития и реализации технологии. В частности, пользовательский уровень может состоять из нескольких подуровней, характеризующих степень «образованности» пользователя в вопросах программирования. В дальнейшем также оказывается, что уровни могут пересекаться. Кроме того, рассматриваемая классификация предполагает, что исходный язык программирования является компилируемым, по крайней мере, в близкий к машинному код. Конечно, режим интерпретации программ имеет свои существенные преимущества в плане верификации программ, возможностей отладки и выполнения других подобных функций. Это, в частности, показывает ряд разработок, в которых автор принимал непосредственное участие [116–119]. Однако в контексте обсуждаемых вопросов эффективность по крайней мере базовой части и наиболее «тонких» в смысле сложности используемых алгоритмов мест является определяющей.

Описываемая технология, как и почти любая другая технология программирования, нуждается в программных средствах поддержки. Рассмотрим такие средства в реализованных автором программных системах на основе операционной системы Windows с использованием среды разработки Delphi и сервера баз данных Interbase. Следует иметь в виду, что выбранные средства не являются единственно возможными в этом плане. Аналогичные идеи можно реализовать в C#/.Net или еще в какой-либо среде, поддерживающей компонентную модель программирования. Выбор возможных серверов баз данных (БД) еще более широк. В дальнейшем под термином «компонент» подразумевается компонент в смысле Delphi.

Итак, первоначально на упомянутом выше системном уровне разработки создается основа – базовое приложение, содержащее минимальный интерфейс пользователя и средства его формирования, а также механизм добавления/исключения компонентов. Этот механизм умеет загружать пакеты компонентов, регистрировать компоненты, настраивать их свойства, вызывать методы. Таким образом, уже на данном этапе программа без дальнейшей корректировки ее базового кода становится неограниченно расширяемой. Причина в том, что подключить к ней можно любой пакет компонентов Delphi, разработкой которых занимаются сотни фирм и тысячи программистов всего мира. На следующем этапе разрабатывается и добавляется к базовому приложению визуальный дизайнер форм, позволяющий создавать новые формы из загруженных компонентов. При таком подходе одним из возможных путей, по которому пошел и автор, является моделирование в базовом приложении исходной среды разработки (IDE Delphi), в которой компоненты легко переводятся в режим дизайна. Основной метод здесь – изучение недокументированных интерфейсов среды и их реализация в специальном классе. Полезными источниками информации в этом плане явились книги [135, 181]. При загрузке базовым приложением пакетов Delphi кроме регистрации самих компонентов регистрируются также связанные с ними редакторы, что позволяет в режиме дизайна форм добиться полной визуальности, во многом не уступающей Delphi. Разрабатываемые пользователем формы можно сохранять как на клиентском компьютере («локальная настройка»), так и в общей базе данных («глобальная настройка»). В базовое приложение (на системном уровне) включаются также несколько классов наиболее часто используемых в данной области применения стандартных форм (форма-дерево, форма-таблица, форма-дерево/таблица), которые будут служить основой для других уровней разработки. На начальном этапе реализуется также как базовый стандартный набор операций с БД (соединение, сохранение/восстановление, выполнение SQL-команд). Вообще в базовое приложение рекомендуется включить возможности, которые могут потребоваться в дальнейшем в любой разрабатываемой системе проектируемого класса.

Следующим шагом является создание и распространение с базовым приложением *прикладных компонентов*. Среди них может содержаться, например, компонент «экспортер бухгалтерской проводки» или «расчет налога». Теперь, когда компоненты могут подключаться без программирования, пользователь сам сможет собрать нужную конфигурацию программы (например, для конкретного рабочего места бухгалтерии) и в дизайнере форм разработать, например, новую форму первичного бухгалтерского документа не с нуля (хотя и это возможно в нашей системе). Тираж распространения компонента соизмерим с тиражом разрабатываемой программы, причем компонент будет использоваться для разработки на пользовательском уровне. Создание же самих прикладных компонентов согласно рассматриваемой классификации осуществляется программистами на прикладном уровне. Прикладные компоненты можно создавать более адаптированными к пользователю, чем обычные компоненты (назовем их системными). Разработанный дизайнер форм способен распознавать прикладные компоненты и показывать названия их свойств и методов в пользовательской интерпретации (в том числе на русском языке). Заметим, что системы пользовательского программирования путем сборки из заготовок разрабатывались В.Ф. Жоголевым и его учениками. В данных исследованиях они называются *системами обосновательного гиперпрограммирования* [110].

Далее выясняется, что кроме установки свойств и возможности вызова методов компонентов, необходимо обрабатывать связанные с ними события. Чтобы это было возможным без изменения исходных текстов базового приложения, разрабатывается (другой возможный вариант – используется готовый) скриптовый язык (например, подмножество Pascal), на котором можно создавать обработчики событий и другие программные части, сохраняемые в разрабатываемых формах. Конструкции языка в целях адаптации к потребностям пользователя должны допускать возможность переименовывания (в том числе по-русски). Реализуется двусторонняя связь опубликованных свойств и методов компонентов с переменными скриптового языка. Из соображений эффективности важно, чтобы этот язык допускал расширения в виде пользовательских функций (UDF), написанных на обычном языке программирования (например, Delphi) и подключаемых в виде DLL. Реализуется также редактор скриптовых программ с возможностью подсветки синтаксиса (с выбором языка – Pascal или SQL).

На данном этапе пользователь уже имеет неограниченные возможности для наращивания функциональных возможностей программ без изменения базового кода. Однако этому препятствует немаловажный фактор – пользователь должен быть в определенной мере программистом. Ему необходимо иметь представление о свойствах и методах используемых компонентов, знать основы несложного скриптового языка. И этого мало, поскольку нужны также определенные навыки программирования, которые невозможно получить без практики, даже если с точки зрения профессионального программиста решаемые на данном уровне задачи примитивны. Идея применения искусственного интеллекта для программирования не нова (см. [195] с библиографией), однако рассматриваемая ситуация имеет существенную особенность – в ней достаточно элементарного программирования при хорошем владении предметной областью. Эта особенность дает основания для оптимизма относительно разрешимости поставленной задачи. Если в [195] в качестве основы представления знаний о решаемой задаче декларируются семантические сети и фреймы [178], то в случае несложного пользовательского программирования достаточно систем продукций. Соответственно для дальнейшего повышения пользовательского уровня разработки программной системы можно предложить создание и использование проблемно-ориентированной транслирующей экспертной продукционной системы. Такая система может интегрировать знания *среднего программиста*, необходимые для связи *квалифицированного пользователя* с «миром программирования». Одной из главных задач настоящей работы является создание математической базы для адекватного описания прикладных систем продукционного типа, которая позволяла бы контролировать в плане эквивалентности преобразований, верифицировать и оптимизировать процессы разработки и модификации ПО.

Важным аспектом многоуровневой технологии является предоставление, начиная с некоторого момента, возможности параллельной разработки системы на всех уровнях, возможности работы на более низком уровне для повышения эффективности разрабатываемой системы.

Почему же изложенную схему разделения на уровни можно считать третьим измерением в модели структуры разработки программ? Во-первых, потому, что на каждом из рассмотренных уровней может применяться упомянутое выше представление о слоях. Во-вторых, при должной технологической поддержке являются взаимозависимыми вдоль уровней (по крайней мере, в сторону повышения), соответствующие горизонтальные и вертикальные слои. Однако более глубокое исследование этого вопроса не является предметом данного раздела. Отметим, что предложенное разделение процесса разработки на иерархические уровни основано на личном опыте автора и его субъективном видении. Как следствие, возможны другие классификации, что, однако, не противоречит самому факту существования третьего измерения.

1.2. Основная терминология решаемых задач

Введем основные обозначения и определения, необходимые для дальнейшего изложения.

Пусть R – бинарное отношение на некотором множестве F. Для каждой упорядоченной пары $(a,b) \in R$ элемент a будем называть ее левой частью, а элемент b – правой частью.

Бинарное отношение R на произвольном множестве F называется:

- рефлексивным, если для любого $a \in F$ справедливо $(a,a) \in R$;

• транзитивным, если для любых $a,b,c \in F$ из $(a,b),(b,c) \in R$ следует $(a,c) \in R$.

Известно, что существует замыкание R^* произвольного отношения R относительно свойств рефлексивности и транзитивности (рефлексивно-транзитивное замыкание).

Пара элементов $a,b \in F$ называется транзитивной в R, если $(a,b) \in R_1^*$, где R_1^* – транзитивное замыкание отношения $R_1 = R \setminus \{(a,b)\}$.

Обратная задача – нахождение транзитивной редукции: по данному R ищется минимальное отношение R' такое, что его транзитивное замыкание совпадает с транзитивным замыканием R. Напомним, что для частично упорядоченных множеств различаются понятия минимального элемента (для него нет меньшего элемента) и наименьшего элемента (он меньше всех) [93]. В [2] приведен алгоритм построения транзитивной редукции ориентированных графов. Показано, что эта задача вычислительно эквивалентна построению транзитивного замыкания, а также доказана единственность транзитивной редукции ациклического графа. Как показано в [71], построение наименьшей транзитивной редукции произвольного графа является NP-полной задачей.

Решеткой $\mathbb{F}$ [93] называется множество с частичным порядком $\leq$ («не больше», «содержится»), на котором для любой пары элементов определены операции $\wedge$ («пересечение») и $\vee$ («объединение»). Элемент $c = a \wedge b$ – это точная нижняя грань элементов a,b, то есть наибольший элемент решетки, удовлетворяющий неравенствам $c \leq a$ и $c \leq b$. Соответственно $d = a \vee b$ – точная верхняя грань a,b, то есть наименьший элемент решетки, для которого выполнено $a \leq d$ и $b \leq d$. Таким образом, при любых $a,b \in \mathbb{F}$ справедливо:

- $a \wedge b \leq a$, $a \wedge b \leq b$;
- если $c \in \mathbb{F}$ и $c \leq a, c \leq b$, то $c \leq a \wedge b$;
- $a \leq a \vee b$, $b \leq a \vee b$;
- если $c \in \mathbb{F}$ и $a \leq c, b \leq c$, то $a \vee b \leq c$.

Пример решетки – множество $\lambda(F)$ всех конечных подмножеств некоторого универсума F.

Решетка $\mathbb{F}$ называется:

• ограниченной, если она содержит общие нижнюю и верхнюю грани, а именно – такие два элемента O, I, что $O \leq a \leq I$ для любого $a \in \mathbb{F}$;

• полной, если каждое ее подмножество имеет в $\mathbb{F}$ точные нижнюю и верхнюю грани.

Пример полной ограниченной решетки – булеан 2^F на некотором универсуме F.

В полной решетке для любого подмножества элементов (конечного или бесконечного) можно определить (многоместные) пересечение и объединение. Для таких операций используются обозначения $\bigwedge$ и $\bigvee$ (с соответствующими индексами, если они нужны). Эти обозначения подобны знакам многоместных операций вычисления произведения $\prod$ и суммы $\sum$ в обычной алгебре.

Точкой (или атомом) ограниченной (снизу) решетки $\mathbb{F}$ называется минимальный элемент ее подмножества $\mathbb{F} \setminus O$. Решетка называется точечной или атомно порожденной, если каждый ее элемент является объединением точек (атомов). Например, в булеане точками являются все подмножества, состоящие ровно из одного элемента универсума. Таким образом, булеан является точечной решеткой. Для точки a элемента A будем иногда использовать обозначение $a \in A$ (наряду с $a \leq A$). В точечной решетке, по определению не являющейся полной, полагаем каждый элемент состоящим из конечного числа точек, поскольку в такой решетке бесконечноместные операции не определены.

Решетка $\mathbb{F}$ называется дистрибутивной, если в ней при любых a, b, c выполняются следующие равенства:

- $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$;
- $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$.

Можно показать, что каждое из этих тождеств следует из другого.

Под дополнением элемента a в ограниченной решетке $\mathbb{F}$ подразумевают элемент $a' \in \mathbb{F}$ такой, что $a \wedge a' = O$ и $a \vee a' = I$. Ограниченная решетка, в которой любой элемент имеет дополнение, называется решеткой с дополнениями. Решетка, в которой каждый замкнутый интервал является решеткой с дополнениями, называется решеткой с относительными дополнениями.

Дистрибутивная решетка с дополнениями называется булевой. В булевой решетке каждый элемент имеет единственное дополнение, причем справедливы тождества (законы де Моргана):

- $(a \wedge b)' = a' \vee b'$;
- $(a \vee b)' = a' \wedge b'$.

В полной булевой решетке справедливы их «бесконечные» аналоги [192]:

- $(\bigwedge\limits_{t \in T} a_t)' = \bigvee\limits_{t \in T} a'_t$;
- $(\bigvee\limits_{t \in T} a_t)' = \bigwedge\limits_{t \in T} a'_t$.

Далее понадобятся следующие свойства ассоциативности и дистрибутивности, связывающие конечные операции с «бесконечными» [192]:

- $b \wedge (\bigwedge_{t \in T} a_t) = \bigwedge_{t \in T}(b \wedge a_t)$; $b \vee (\bigvee_{t \in T} a_t) = \bigvee_{t \in T}(b \vee a_t)$;
- $b \vee (\bigwedge_{t \in T} a_t) = \bigwedge_{t \in T}(b \vee a_t)$; $b \wedge (\bigvee_{t \in T} a_t) = \bigvee_{t \in T}(b \wedge a_t)$.

Под LP-структурой будем подразумевать алгебраическую систему, представляющую решетку, на которой задано бинарное отношение, обладающее некоторыми продукционно-логическими свойствами. Решетка в контексте данного определения рассматривается в широком смысле, и ее тип может уточняться в конкретных моделях.

Отношение называется продукционно-логическим, если оно обладает рефлексивностью, транзитивностью и другими свойствами, которые также определяются конкретной моделью. Одно из таких свойств – дистрибутивность. Неформально оно означает возможность логического вывода по частям и объединения его результатов на основе решеточных операций $\wedge$ и $\vee$. Заданное на абстрактной решетке отношение R называется:

- $\wedge$-дистрибутивным, если из $(a,b_1),(a,b_2) \in R$ следует $(a,b_1 \wedge b_2) \in R$;
- $\vee$-дистрибутивным, если из $(a_1,b),(a_2,b) \in R$ следует $(a_1 \vee a_2,b) \in R$.

Отношение называется дистрибутивным при наличии обоих указанных свойств. На полной решетке (полная) дистрибутивность отношения определяется аналогично, однако с использованием бесконечноместных операций пересечения и объединения.

В некоторых разделах настоящей работы в качестве основы LP-структур рассматриваются решетки с семантикой подмножеств – $\lambda(F)$ (множество конечных подмножеств F) или 2^F (множество всех подмножеств F). Это делается с целью подчеркнуть теоретико-множественное происхождение соответствующих моделей. Соответственно вместо символов $\le$, $\ge$, $\wedge$ и $\vee$ используются знаки теоретико-множественных операций $\subseteq$, $\supseteq$, $\bigcap$ и $\bigcup$, а элементы решетки обозначаются большими буквами. В этих случаях будем иметь в виду лишь второе из двух указанных выше свойств. В такой нотации $\bigcup$-дистрибутивность трактуется в следующем смысле: из $(A,B_1),(A,B_2) \in R$ следует $(A,B_1 \bigcup B_2) \in R$. Это свойство будем называть дистрибутивностью.

В ряде моделей используется еще одно свойство отношений. Бинарное отношение R на решетке с дополнениями называется контрапозиционным, если из $(a,b) \in R$ следует $(b',a') \in R$.

Подобно транзитивным замыканию и редукции отношения на обычном множестве, в теории LP-структур решаются более сложные задачи нахождения логического замыкания и логической редукции отношений на иерархических множествах – различных видах решеток.

Логическим замыканием заданного на решетке произвольного отношения R называется наименьшее продукционно-логическое отношение, содержащее R. Два отношения R_1, R_2 называются эквивалентными, если их логические замыкания совпадают. Для таких отношений используется обозначение $R_1 \sim R_2$. Эквивалентным преобразованием данного отношения называется некоторая замена подмножества его пар, приводящая к эквивалентному отношению. Логической редукцией отношения R называется минимальное отношение, эквивалентное R.

Последующие разделы главы посвящены возможным применениям LP-структур к системам продукционного типа. С этой целью введем терминологию, связанную с экспертными продукционными системами. Указанные системы манипулируют множествами фактов и правил (продукций). *Факт* представляет собой единицу декларативной информации – некоторое суждение о внешнем мире или состоянии системы. Стандартным представлением факта является триплет вида «объект.атрибут = значение» [20] (например, «термометр.температура = высокая»). В книге [79], описывающей практическую реализацию продукционных систем, триплет редуцируется к паре «параметр = значение». Это означает, что объект и атрибут интегрируются в единый параметр. В данной связи «термометр.температура» и «термометр.изготовитель» считаются разными параметрами. В некоторых разделах данной работы пойдем дальше, интегрируя в факт параметр и его значение, считая факт независимым элементом общего множества фактов. Упрощение делается там, где необходимо больше сосредоточиться на основных идеях применения LP-структур. В дальнейшем можно реализовать и более сложную конструкцию фактов, скорректировав соответствующим образом описание LP-структуры.

Продукционная система содержит так называемую *рабочую память*. Это некоторое подмножество фактов, которые на текущий момент считаются выполненными. Иногда будем называть такое множество базой данных продукционной системы.

Правило (*продукция*) состоит из предпосылки и заключения. *Предпосылка* обычно представляет собой выражение над фактами (например, их конъюнкцию или дизъюнкцию). Предпосылка может быть выполненной (истинной) или невыполненной (ложной) при текущем состоянии рабочей памяти. Если предпосылка верна, то правило может быть применено. *Заключение* – это действие, которое можно осуществить, если верна предпосылка (например, добавить к рабочей памяти некоторый новый факт). *Применение правила* состоит в выполнении действия заключения. В контексте настоящей работы рассматриваются приложения, в которых заключение, как и предпосылка, является выражением над фактами, и соответствующее заключению действие интерпретируется как «считать истинным». Таким образом, в данном случае применение правила означает некоторую модификацию рабочей памяти, обычно – запись в нее тех фактов, справедливость которых вытекает из истинности выражения в заключении правила. Совокупность правил называется *базой знаний.*

Прямым выводом в продукционной системе называется процесс циклического применения правил к содержимому рабочей памяти (его исходное состояние задано в начале работы), и соответственно получение в результате новых фактов, которые считаются справедливыми. Прямой вывод может производиться до тех пор, когда получение новых фактов станет невозможным (при текущем содержимом рабочей памяти не окажется ни одной истинной предпосылки правила, заключение которого способно изменить рабочую память). *Обратный вывод* – это противоположный процесс. В нем по некоторому набору результирующих фактов – *гипотезе*, путем анализа правил в направлении от заключения к предпосылке, подтверждается или опровергается справедливость гипотезы при заданном исходном содержимом рабочей памяти. В процессе обратного вывода содержимое рабочей памяти также меняется.

Далее в качестве возможных приложений теории LP-структур рассматриваются несколько видов продукционных систем, различающихся используемой структурой правила, точнее — выражения для его предпосылки и заключения.

1.3. Стандартная продукционная система

Простейшую продукционную систему можно представить как множество элементарных фактов F и множество R правил $a \to b$, где $a, b \in F$. Элементарный факт – это не содержащее логических операций (элементарное) суждение. Правило $a \to b$ имеет продукционную семантику «**если** верно a, **то** верно b» (например, «**если** идет дождик, **то** взять зонтик»; «**если** температура высокая, **то** заболел»).

Алгебраический метод исследования такой системы сводится к изучению бинарного отношения R на множестве F. Очевидно, в силу особенностей данной продукционной системы R является рефлексивным и транзитивным. Действительно, для любого $a \in F$ можно констатировать, что «если верно a, то верно a». Также при наличии в R двух правил $a \to b$ и $b \to c$ при выводе фактически действует и правило $a \to c$. Например, по правилам «**если** температура высокая, **то** заболел» и «**если** заболел, **то** на работу не ходить» можно сформировать правило «**если** температура высокая, **то** на работу не ходить». Соответствующую данной модели алгебраическую систему можно считать вырожденным случаем LP-структуры с пустым базовым отношением частичного порядка.

Вряд ли можно назвать исчерпывающим знанием правило «**если** заболел, **то** на работу не ходить». Более близкими к истине являлись бы следующие высказывания: «**если** заболел, сегодня рабочий день, время утреннее, **то** на работу не ходить, лечиться», «**если** заболел, сегодня рабочий день, время вечернее, **то** лечиться», «**если** заболел, сегодня выходной день, **то** лечиться».

Принимая во внимание изложенные выше соображения, рассмотрим более сложную, называемую стандартной, модель базы знаний, правила которой в качестве предпосылки и заключения могут содержать наборы элементарных фактов ($\{a\}$, $\{a,b\}$, $\{a,b,c\}$, ...). Общий вид продукции в данном случае таков: $\{a_1,\dots,a_n\} \to \{b_1,\dots,b_m\}$, где a_i и b_j – факты. Смысл такого правила состоит в следующем: если верны все $a_i, i=1,\dots,n$, то верны и все $b_j, j=1,\dots,m$.

В этой модели объектами бинарного отношения $\to$ являются не элементарные факты, а конечные подмножества их исходного множества F. Состоящая из таких объектов математическая структура является частным случаем решеток $\mathbb{F} = \lambda(F)$.

В данном разделе с учетом происхождения рассматриваемой модели используются обозначения, принятые в теории множеств, а не в теории решеток. Чтобы не путать с объектами простейшей логики, элементы $\mathbb{F}$ будем обозначать большими латинскими буквами. Для любых A,B в $\mathbb{F}$ определены (теоретико-множественные) операции пересечения ($A \cap B$) и объединения ($A \cup B$). Кроме того, в $\mathbb{F}$ задан частичный порядок – отношение включения $\subseteq$.

Таким образом, на множестве $\mathbb{F}$ имеем базовое отношение $\subseteq$, задающее структуру решетки, и дополнительное отношение $\to$, алгебраически отражающее логические связи знаний конкретной предметной области. Ввиду усложнения модели, логические отношения в данной LP-структуре, кроме имевшихся в предыдущем разделе свойств рефлексивности и транзитивности, должны также обладать дополнительными свойствами. Свойство рефлексивности является теперь частным случаем свойства тавтологии включения: при $A \supseteq B$ имеет место $A \to B$. Действительно, если справедливо некоторое множество фактов A, то верно и любое его подмножество B. Кроме того, любую совокупность фактов можно выводить по частям: если $A \to B$ и $A \to C$, то $A \to B \bigcup C$. Это свойство логических отношений называем дистрибутивностью. Из него формально следует монотонность такой логики: поскольку $A \to A$, то из $A \to B$ по свойству дистрибутивности получаем $A \to A \bigcup B$. Это означает монотонное накопление знаний при применении правил.

Теоретически возможны также продукционные системы с бесконечными множествами правил. Они могут быть описаны, например, правилами с использованием в них переменных [20], формально имеющих бесконечные области определения. Эквивалентные преобразования таких систем приводят к правилам с бесконечными предпосылками и заключениями.

Стандартная продукционная система с бесконечными правилами моделируется на основе булеана $\mathbb{F} = 2^F$ – полной ограниченной решетки. В такой модели в качестве дополнительного свойства соответствующего продукционно-логического отношения требуется сформулировать свойство полной дистрибутивности: если $A \to B_t,\ t \in T$, то $A \to \bigcup_t B_t$.

В приведенных стандартных моделях продукционных систем более сложными (по сравнению с простейшим случаем) являются не только сами объекты продукций (предпосылка и заключение), но и выводимые логические связи. Если в простейшей системе «лишние» правила можно легко определить только на основе свойства транзитивности отношения, то в случае стандартной системы для этого необходимо решить задачу нахождения логической редукции LP-структуры.

1.4. Расширенная продукционная система

В рамках предыдущего раздела невозможно было бы использовать довольно естественное правило вида «**если** на улице дождик, **то** взять зонтик **или** отложить поход на завтра». Кроме того, несмотря на возможность присутствия в той модели правила «**если** дождика нет **и** прогноз благоприятный, **то** зонтик не брать», в процессе исследования системы возникает естественное желание учитывать, что некоторые фигурирующие в этом правиле факты представляют отрицания фактов из предыдущего правила. Например, нельзя одновременно взять зонтик и пойти без него.

Таким образом, использования в предпосылках и заключениях правил конъюнкций элементарных фактов явно недостаточно. В литературе описываются продукционные системы [15], в правилах которых могут присутствовать сложные логические выражения, содержащие конъюнкции, дизъюнкции и отрицания. Конечно, теоретически в пропозициональных формулах (если относить к таковым выражения предпосылки и заключения) возможны и импликации. Однако каждую из них всегда можно исключить с использованием дизъюнкции и отрицания. Таким образом, в настоящем разделе рассматриваются базы знаний, продукции которых в качестве предпосылки и заключения содержат логические выражения над фактами, построенные с помощью конъюнкций, дизъюнкций и отрицаний. Данную логическую систему можно назвать продукционной системой нулевого порядка, или расширенной продукционной системой.

Для такого вида продукционных систем также может быть построена алгебраическая модель в виде LP-структуры. Ее решетка должна поддерживать три отмеченные ранее операции пропозиционального языка. Такова булева решетка Линденбаума–Тарского [187]. Ее элементами являются формулы пропозиционального исчисления. Логическим операциям конъюнкции, дизъюнкции и отрицания соответствуют определенные в п. 1.2 алгебраические операции $\wedge$, $\vee$ и $'$. Для любых двух формул $a, b \in \mathbb{F}$ справедливо $a \leq b$, если из истинности формулы a следует истинность формулы b. Эквивалентные логические формулы отождествляются.

Как и в предыдущем разделе, объектами продукционно-логического отношения $\to$ будут элементы решетки. Однако в данном случае решетка имеет более сложную архитектуру. Чтобы адекватно отражать расширенную продукционную логику, логическое отношение должно удовлетворять большему количеству требований по сравнению с предыдущим разделом.

Таким образом, методы LP-структур могут быть применены к исследованию, преобразованию и оптимизации баз знаний продукционных систем с расширенной логикой.

1.5. Продукционная система первого порядка

Существуют виды интеллектуальных систем, которые наряду с логическими выражениями в правилах вывода могут потребовать использования также кванторов общности и существования. Такие модели, в частности, возникают при разработке систем символьной математики. Большинство математических теорем имеют форму продукции вида «дано» → «требуется доказать», где предпосылка и заключение являются формулами логики предикатов первого порядка.

Рассмотрим продукционную систему, правила которой в качестве предпосылки и заключения могут содержать формулы исчисления предикатов первого порядка. Как и в предыдущем разделе, будем считать, что эти формулы не содержат импликаций (при необходимости их можно исключить). Будем называть ее продукционной системой первого порядка. Соответственно возникает идея о распространении методов теории LP-структур на такую логику. Это позволило бы использовать ее аппарат для формальных преобразований математических знаний, а также их верификации и оптимизации. Для реализации этой идеи потребуется некоторое алгебраическое описание логических кванторов, которое дополнило бы использованное в предыдущем разделе понятие булевой решетки.

Обзор имеющихся подходов к данному вопросу содержится в [63]. Воспользуемся алгебраизацией кванторов, предложенной в [187], поскольку она естественным образом расширяет рассматриваемую продукционную логику на решетке и содержит достаточные возможности для решения поставленных задач. Согласно этой теории, кванторы общности и существования соответственно моделируются в общем случае бесконечноместными операциями пересечения и объединения. Пусть $P(x)$ – предикат со свободной переменно x, интерпретируемый в некоторой предметной области X. Его можно представить совокупностью элементов решетки Линденбаума–Тарского, соответствующих формуле $P(x)$ при различных значениях $x \in X$. Тогда, в силу свойств логических операций, выражение $\forall x P(x)$ алгебраически представляется операцией пересечения всех элементов решетки, соответствующих $P(x)$. Аналогично формула $\exists x P(x)$ может быть представлена объединением всех элементов решетки, порожденных предикатом $P(x)$.

Как уже отмечалось выше, не каждая решетка допускает выполнение бесконечноместных операций. Чтобы они были возможны и корректны, можно использовать полные булевы решетки. В полной решетке для любого подмножества элементов (конечного или бесконечного) можно определить (многоместное) пересечение и объединение. Формулам $\forall x P(x)$ и $\exists x P(x)$ в полной решетке будут соответствовать элементы $\bigwedge_{x \in X} P(x)$ и $\bigvee_{x \in X} P(x)$. Изложенные соображения открывают возможности формальных исследований, преобразований и автоматической оптимизации продукционных баз знаний с логикой первого порядка.

В качестве иллюстрации рассмотрим пример из приложения В данной работы, относящийся к области неклассических уравнений с частными производными. В этом приложении в коэффициентах символов псевдодифференциальных операторов используется «весовая» вещественная функция α, существенно влияющая на разрешимость уравнений с этими операторами. В основе результатов, представленных в приложении В, лежат следующие два условия на данную функцию.

1. $\alpha(t) = 0,\, t \le 0;\ \ \alpha(t) > 0,\, t > 0;\ \ \alpha(t) = \mathrm{const},\, t \ge d > 0$.
2. $\alpha|_{R^+} \in C^\infty\left(\bar{R}^+\right);\ \exists M \ge 0 : \alpha \in C^{M+1}(R)$.

Для построения соответствующей LP-структуры введем обозначения логических формул. Пусть $P_{11}(t)$, $P_{12}(t)$ и $P_{13}(t,c)$ – предикаты, соответствующие утверждениям $\alpha(t) = 0$, $\alpha(t) > 0$ и $\alpha(t) = c$. Тогда при каждом значении переменной t имеем элементы решетки $P_{11}(t)$, $P_{12}(t)$ и каждой паре значений t, c сопоставляем элемент $P_{13}(t,c)$. Условию 1 соответствует элемент $\bigwedge\limits_{t\le 0} P_{11}(t) \wedge \bigwedge\limits_{t>0} P_{12}(t) \wedge \bigvee\limits_{c>0} \bigwedge\limits_{t\ge d>0} P_{13}(t,c)$.

Для условия 2 также введем символы $P_{21}(t)$ и $P_{21}(t,M)$, соответственно означающие «функция α бесконечно дифференцируема в точке t» и «функция α M раз непрерывно дифференцируема в точке t». Алгебраическая запись условия 2 будет выглядеть как $\bigwedge\limits_{t\ge 0} P_{21}(t) \wedge \bigvee\limits_{M\ge 0} \bigwedge\limits_{-\infty<t<+\infty} P_{22}(t,M)$. Таким образом, большую часть утверждений приложения В алгебраически можно представить продукциями, предпосылки которых содержат выражение вида $\bigwedge\limits_{t\le 0} P_{11}(t) \wedge \bigwedge\limits_{t>0} P_{12}(t) \wedge \bigvee\limits_{c>0} \bigwedge\limits_{t\ge d>0} P_{13}(t,c) \wedge \bigwedge\limits_{t\ge 0} P_{21}(t) \wedge \bigvee\limits_{M\ge 0} \bigwedge\limits_{-\infty<t<+\infty} P_{22}(t,M)$.

1.6. Условная эквациональная теория

Еще одним из возможных направлений применения идей и методов LP-структур является построение и исследование моделей эквациональных теорий и систем переписывания термов (СПТ) [18, 72].

Важными задачами, связанными с СПТ, являются эквивалентные преобразования и оптимизация их множеств правил. В то время как для обычных СПТ подобные вопросы уже рассматривались в ряде работ [23, 57, 81], для условных СПТ они, по-видимому, еще остаются открытыми. Этот факт можно объяснить более сложной структурой правил условных СПТ. Для обычных систем задача минимизации множества правил в конечном счете сводится к транзитивной редукции некоторого бинарного отношения («элиминация транзитивности»). Для условных систем, как и в предыдущих разделах, можно говорить о более сложной задаче нахождения логической редукции.

При определении системы переписывания термов отправной точкой обычно является эквациональная теория, множество правил которой состоит из равенств. Совокупность правил переписывания получается путем «ориентации» равенств и, возможно, их пополнения для достижения свойства конфлюэнтности. Аналогичный подход используется и для условных СПТ [19]. Поскольку обычно именно эквациональная теория является критерием эквивалентности систем переписывания, исследование в этом плане условных СПТ можно начать с рассмотрения эквивалентности условных эквациональных теорий.

Для дальнейшего изложения потребуется ряд базовых определений, связанных с *термами* [46]. Пусть Σ – алфавит, представляющий собой объединение следующих непересекающихся множеств: V – множество переменных; Σ_n, $n = 0,1,\ldots$ – множества n-арных функций (функциональных символов); 0-арные функции, которые называются также константами.

Множество термов $T(\Sigma)$ определяется рекурсивно:

- $V \subset T(\Sigma)$;
- $\Sigma_0 \subset T(\Sigma)$;
- если $f \in \Sigma_n$ и $t_1,\ldots,t_n \in T(\Sigma)$, то $f(t_1,\ldots,t_n) \in T(\Sigma)$.

Отображение $\sigma : V \to T(\Sigma)$ называется подстановкой. Это понятие распространяется на все $t \in T(\Sigma)$ следующим образом:

- если $t = x \in V$, то $\sigma(t) = \sigma(x)$;
- если $t = f \in \Sigma_0$, то $\sigma(t) = f$;
- если $f \in \Sigma_n$, $t_1,\ldots,t_n \in T(\Sigma)$ и $t = f(t_1,\ldots,t_n)$, то $\sigma(t) = f(\sigma(t_1),\ldots,\sigma(t_n))$.

Эквациональная теория определяется парой (Σ, E), где Σ – алфавит, состоящий из счетного множества переменных и непустого множества функциональных символов (сигнатура), а $E \subseteq T(\Sigma) \times T(\Sigma)$ – множество равенств вида $s = t$ $(s,t \in T(\Sigma))$. В рамках этой теории определяется понятие выводимости равенства $s = t$ из E ($(\Sigma, E) \vdash s = t$):

- если $s = t \in E$, то $(\Sigma, E) \vdash s = t$;
- если $(\Sigma, E) \vdash s_1 = t_1,\ldots,s_n = t_n$, то $(\Sigma, E) \vdash f(s_1,\ldots,s_n) = f(t_1,\ldots,t_n)$ для любого $f \in \Sigma_n$;
- если $(\Sigma, E) \vdash s = t$, то $(\Sigma, E) \vdash \sigma(s) = \sigma(t)$ для любой подстановки σ;
- $(\Sigma, E) \vdash t = t$;

- если $(\Sigma, E) \vdash t_1 = t_2, t_2 = t_3$, то $(\Sigma, E) \vdash t_1 = t_3$;
- если $(\Sigma, E) \vdash s = t$, то $(\Sigma, E) \vdash t = s$.

Предположим, что эквациональная теория (аналогично [19]) содержит набор позитивно-условных правил вида $s_1 = t_1, \ldots, s_n = t_n : s = t$. Смысл такого правила в следующем: если имеют место все равенства термов $s_i = t_i, i = 1, \ldots, n$, то выполнено и $s = t$. Если вместо термов рассматривать независимые элементы, то задача упрощения множества правил может быть сведена к уже описанным выше LP-структурам. Этот же метод применим и в случае равенств между термами, однако оптимизация при этом может оказаться лишь частичной.

Рассмотрим пример, полученный модификацией примера из [19]:

1) $x + y = z : s(x) + y = s(z)$;
2) $x + y = z : g(x) + y = g(z)$;
3) $s(x) + y = s(z), g(x) + y = g(z) : f(x) = f(z)$;
4) $x + y = z : f(x) = f(z)$.

Последнее равенство здесь является «лишним», так как выводится из первых трех. Действительно, пусть $x + y = z$. Тогда из 1)-2) имеем $s(x) + y = s(z)$, $g(x) + y = g(z)$, откуда с помощью 3) получим $f(x) = f(z)$. Таким образом, при $x + y = z$ справедливо $f(x) = f(z)$, то есть условное равенство 4) неявно содержится в 1)–3).

Этот факт может быть обнаружен методами стандартных LP-структур, поскольку для этого не требуется учета особенностей термов. Заменим теперь второе правило следующим:

2) $h(x + y) = h(z) : g(x) + y = g(z)$.

Теперь формальное применение продукционного вывода не поможет исключить «лишнее» правило 4), если не учесть связь $x + y = z : h(x + y) = h(z)$, которая обусловлена свойствами термов.

Таким образом, можно ввести новую LP-структуру – алгебраическую модель условной эквациональной теории. Эта модель должна учитывать не только обычные логические связи, но и взаимозависимости между термами, обусловленные применениями функций и подстановками. Множество правил можно задать как бинарное отношение на решетке $\mathbb{F} = \lambda(F)$, где F – множество равенств вида $\{s_i = t_i\}$. На этой решетке необходимо ввести дополнительные операции, отражающие использование функциональных символов и подстановок в термах. Такая модель закладывает теоретические основы для автоматической минимизации множества правил условной эквациональной теории.

1.7. Модель иерархии типов

Перечисленные выше возможности применения LP-структур ограничены системами с *монотонным* выводом. Однако, существуют практические задачи, модели которых подобным свойством не обладают. Одна из таковых основана на использовании решетки типов. Как известно [194], структура типов объектно-ориентированной системы образует математическую решетку. Для двух типов объединением считается их ближайший общий тип-предок, пересечением – ближайший общий тип-потомок. Семантика частичного порядка и основных операций на такой решетке существенно отличается от случая решетки множеств. Доказано [77], что решетки типов не изоморфны решеткам множеств. По этой причине LP-структуры на решетках типов и возможности их применения требуют дополнительного анализа.

Рассмотрим некоторую иерархию типов $\mathbb{F}$ в объектно-ориентированном программировании. Между парами типов могут существовать как минимум два вида связей – наследование (тип наследует атрибуты типа-предка) и агрегация (тип содержит в качестве атрибута представителя другого типа) [199]. На множестве $\mathbb{F}$ введем отношение частичного порядка: если тип b является наследником a (соответственно a – предком b), то $b \le a$. Для любых $a, b \in \mathbb{F}$ определены две операции: пересечение $a \wedge b$ – наибольший общий потомок и объединение $a \vee b$ – наименьший общий предок a, b (первая операция актуальна в системах с множественным наследованием). Для ограниченности полученной решетки добавим к $\mathbb{F}$ два специальных элемента: I – универсальный тип (общий предок, имеется во многих современных системах) и O – фиктивный потомок всех типов.

На решетке $\mathbb{F}$ рассмотрим второе (соответствующее агрегации) отношение R: если объект типа a в качестве атрибута содержится в типе b , то $(b, a) \in R$. Оба отношения ($\le$ и R) имеют общую семантику: в каждом случае, $b \le a$ или $(b, a) \in R$, тип b получает возможности типа a в виде доступа к его атрибутам. Семантически ясно, что это общее отношение «обладания набором возможностей», которое мы обозначаем $\xleftarrow{R}$, обязано быть рефлексивным и транзитивным.

Обсудим еще одно свойство введенных отношений. Пусть для элементов $a, b_1, b_2 \in \mathbb{F}$ справедливо $b_1 \le a$, $b_2 \le a$. Тогда в соответствии с определением решетки имеем $b_1 \vee b_2 \le a$. Это естественное для отношения $\le$ свойство будем называть $\vee$-дистрибутивностью. Уточним, что будет означать обладание этим же свойством для отношения $\xleftarrow{R}$. Пусть $b_1 \xleftarrow{R} a$ и $b_2 \xleftarrow{R} a$, то есть каждый тип b_1 и b_2 обладает возможностями типа a. Отсюда, в силу предполагаемой $\vee$-дистрибутивности, имеем $b_1 \vee b_2 \xleftarrow{R} a$. Этот факт означает, что тип $b_1 \vee b_2$ также обладает возможностями типа a. С точки зрения проектирования типов это не обязательно. Однако в случае, если более одного наследника (b_1 и b_2) содержат одинаковые атрибуты, по принципам рефакторинга [56] нужно «поднять» общие атрибуты. Это, в свою очередь, означает необходимость поместить один такой атрибут в общий тип-предок $b_1 \vee b_2$, после чего каждый b_1 и b_2 получит возможности a в порядке наследования. В данном случае $\vee$-дистрибутивность отношения $\xleftarrow{R}$ означает решение важной задачи – устранение дублирования кода.

Подчеркнем, что здесь рассматривается обобщенная постановка в плане «распределения возможностей» между типами. Вариант, когда b_1 и b_2 по каким-либо практическим соображениям содержат в виде атрибутов представителей типа a под различными идентификаторами, в расчет не принимается. На практике тип может содержать много атрибутов, однако не все они одинаково существенны при построении иерархии.

Заметим далее, что отношение $\xleftarrow{R}$, обладая свойством транзитивности вне зависимости от контекста, не может аналогично во всех ситуациях удовлетворять $\vee$-дистрибутивности, иначе это приведет к некорректным результатам. Для иллюстрации рассмотрим пример $b \xleftarrow{R} a$ и $a \xleftarrow{R} a$. При дистрибутивности $\xleftarrow{R}$ выполнялось бы $b \vee a \xleftarrow{R} a$. Согласно принципам объектно-ориентированного программирования, тип $b \vee a$ не имеет права что-либо знать о своих наследниках. В этой связи в данной ситуации $b \vee a$, являясь общим предком типов a и b, может обладать возможностями типа a лишь в случае, если он совпадает с a ($b \leq a$). В остальных вариантах соотношение $b \vee a \xleftarrow{R} a$ будет некорректным.

Существуют ситуации, когда выполнение $\vee$-дистрибутивности отношения $\xleftarrow{R}$ теоретически возможно, однако нецелесообразно с точки зрения качества кода. Пусть при $b_1 \xleftarrow{R} a, b_2 \xleftarrow{R} a$ элементы $b_1 \vee b_2$ и a имеют непустое пересечение: $(b_1 \vee b_2) \wedge a = d \neq O$, причем $d < b_1 \vee b_2$ и $d < a$. Если в данном случае допустить $(b_1 \vee b_2, a) \in R$, то окажется, что тип d обладает возможностями типа a одновременно по двум линиям, а именно – как его наследник и как наследник типа $b_1 \vee b_2$, также имеющего возможности a. Некачественность такого кода состоит в его избыточности – разрыв связи $d < a$ (при $a \neq I$) не приведет к потере функциональности системы типов.

Другая подобная ситуация – наличие конфликта. Пусть имеет место $(b_1,a),(b_2,a),(b_3,a) \in R$, причем элементы b_1, b_2, b_3, $b_1 \vee b_2$, $b_2 \vee b_3$ попарно различны и $(b_1 \vee b_2) \wedge (b_2 \vee b_3) = b_2$. Тогда пары $(b_1,a),(b_2,a)$ «конфликтуют» с парами $(b_2,a),(b_3,a)$. Это означает, если в обоих случаях «поднять» атрибуты, то тип b_2 унаследует атрибут типа a одновременно от двух различных предков – $b_1 \vee b_2$ и $b_2 \vee b_3$, что также ухудшит код.

Одна из возможных в данном случае стратегий предполагает отказ от «поднятия» общих атрибутов при наличии подобных ситуаций (невыполнение $\vee$-дистрибутивности). Возможны и другие подходы, более тонко учитывающие особенности конкретной системы.

На основе изложенных выше соображений можно дать строгое определение логического отношения на ограниченной решетке. Оно будет отражать свойство «обладания набором возможностей» в иерархии типов. В силу приведенных примеров, свойство дистрибутивности такого отношения зависит от контекста. Этот факт, в частности, порождает немонотонность логического вывода на таких LP-структурах.

Логическое замыкание произвольного отношения R на решетке $\mathbb{F}$ предоставляет все такие пары (b,a), что в типе b доступны возможности типа a. Решив задачу построения логического замыкания, можно автоматизировать верификацию системы типов. К таким возможностям, в частности, относится исследование LP-структуры на наличие циклов. Аналогичным образом, исходя из предметной области, можно формулировать правила, которым должна удовлетворять система типов, и контролировать их выполнение. Например, возможна проверка системы на наличие нужных или отсутствие запрещенных продукционных связей («автомобиль обладает возможностями руля», «руль не имеет двигателя» и других, подобных им). Логическая редукция дает иерархию с минимальным эквивалентным набором связей и, соответственно, в определенном смысле минимальным дублированием кода. Обнаружение и устранение дублирования кода также можно считать разновидностью верификации типов.

Таким образом, рассматриваемый формализм позволяет проводить автоматизированные исследования иерархий типов, включая эквивалентные преобразования, верификацию и оптимизацию. Он может служить основой для практической реализации (или модернизации) типов.

1.8. Продукционная модель императивных алгоритмов

Существуют практические задачи искусственного интеллекта, предполагающие не только накопление, но и модификацию получаемых знаний. Эти задачи рассматриваются в ряде работ (например, [90, 95, 194]). В данном разделе добавим к ним еще одну предметную область – анализ свойств и преобразования императивных алгоритмов. Во время их работы информация не только накапливается, но и часто замещается.

Приведем фрагмент программы на Паскале, вычисляющий максимальный элемент целочисленного массива.

```
var
   A: array[1..N] of integer;
   Max, i: integer;
begin
   Max := A[1]; i := 2;
   while i <= N do
   begin
      if A[i] > Max
      then Max := A[i]; i := i + 1
   end
end;
```

Используемые в программе переменные с их значениями можно считать фактами некоторой базы знаний. Тогда инициализация любой переменной может рассматриваться как добавление нового факта к базе знаний. Присвоение же инициализированной переменной нового значения можно интерпретировать как модификацию имеющегося факта.

Попробуем для приведенной выше императивной программы записать эквивалентную логическую программу – исходный набор фактов и совокупность правил продукционного вывода. Для этой цели аналогично подходу, изложенному в п.1.3, будем использовать простейший синтаксис вида

```
если <Факт 1>, <Факт 2>, …, <Факт K>
то <Действие 1> , <Действие 2>, <Действие M>;
```

Таким образом, предположим, что исходная база знаний состоит из следующих фактов: значения всех переменных не определены (*null*), кроме одной вспомогательной переменной *начало*, имеющей значение *да.* Соответствующая логическая программа может иметь вид

```
если начало = да то Max = A[1], i = 2, цикл = да, начало = нет;
если цикл = да, i <= N, A[i] > Max то Max = A[i], i = i + 1;
если цикл = да, i <= N, A[i] <= Max то i = i + 1;
если цикл = да, i > N то цикл = нет;
```

Возможно, используемый синтаксис правил требует уточнений, а приведенная логическая программа не оптимальна. В данном случае существенно то, что она имеет декларативный характер и дает эквивалентный результат, независимо от порядка записи правил.

Классическое понятие решетки оказывается недостаточно выразительным для моделирования немонотонного продукционно-логического вывода. В связи с этим обстоятельством далее будут определены некоммутативные решетки как обобщение классических решеток. В новых решетках операция объединения выполняется с частичным замещением одного из операндов. Для построения соответствующих LP-структур будут введены логические бинарные отношения на этих решетках, а также описана возможность применения полученной модели к формальному исследованию логических свойств императивных алгоритмов.

Итак, в данной главе был рассмотрен ряд задач из нескольких разделов информатики и теории программирования, описание которых может быть сведено к продукционным системам, моделируемым LP-структурами. Это обстоятельство демонстрирует не только теоретическую значимость, но и прикладную ценность идей, которые лежат в основе настоящей работы. Представленные соображения позволяют естественным образом перейти ко второй главе – изложению общей теории LP-структур и далее, в следующих главах, к развитию этой теории с целью решения практических задач.

Глава 2
ОБЩАЯ ТЕОРИЯ LP-СТРУКТУР

В настоящей главе излагаются основные положения теории LP-структур. Теория названа общей, поскольку результаты последующих глав книги основываются на тех же положениях, развивая их в том или ином специальном направлении. Первоначально в разделе 2.1 формулируется более частная конечная теоретико-множественная модель продукционной системы. Обобщение этой модели и связанных с ней результатов приводит в дальнейшем к теории LP-структур, которая рассматривается в разделах 2.2–2.6.

2.1. Порождающие множества в продукционных системах

В данном разделе предлагается формальный теоретико-множественный подход к описанию структуры продукционных систем. Вводится ряд математических понятий, базовым среди которых является минимальное порождающее множество. Процесс обратного логического вывода сводится к формальному построению минимальных порождающих множеств. На основе построенной модели определяется ряд важных свойств продукционных систем (противоречивость, избыточность, полнота и другие), на основании которых реализуется формальный подход к выполнению автоматических преобразований и верификации баз знаний. Использование предлагаемого подхода позволяет:

- математически обосновывать возможности эквивалентных преобразований баз знаний продукционных систем;
- оптимизировать и исследовать алгоритмы обратного вывода;
- строго формулировать критерии корректности баз знаний, а также разрабатывать методы их верификации.

Необходимые основные понятия, связанные с продукционными системами, были приведены ранее в п. 1.2. Добавим к ним еще некоторые термины.

В настоящем разделе для продукционной системы принята модель фактов, используемая в [79] (см. также п.1.2), то есть в форме «параметр = значение».

В множестве параметров можно выделить следующие подмножества: *целевые параметры* – те, которые могут служить конечной целью вывода; *начальные параметры*, которые не встречаются в заключениях правил, соответственно, значения которых изначально хранятся в базе данных (БД) либо могут быть получены от пользователя и записаны в БД; *промежуточные параметры* – не вошедшие в первые два подмножества и носящие вспомогательный характер; *многозначные параметры*, которые могут принимать одновременно несколько значений. Для параметров, не являющихся многозначными, возможные значения являются взаимоисключающими. Факт, порожденный начальным параметром, будем называть *начальным фактом*.

В ряде работ по тематике продукционных систем (например, [1]) предполагается, что все правила находятся в *унифицированной форме*, в которой предпосылка состоит только из конъюнкций фактов, а заключение делает вывод значения только одного параметра. Такие базы знаний называются также «хорошо структурированными» [67]. В логической интерпретации продукционных систем базам знаний с таким множеством правил соответствуют функции Хорна [39]. Указанное допущение обычно не ограничивает общности результатов, так как они могут распространяться на более общую структуру правил. В большей части настоящего раздела будем исходить из того, что для формулы предпосылки каждого правила можно указать конъюнкцию фактов, при которой предпосылка выполнена. Сделаем также допущение о том, что из выражения заключения каждого правила можно вывести некоторую конъюнкцию других фактов. В данной модели рассмотрим вопросы эквивалентных преобразований базы знаний, позволяющих привести ее к унифицированному виду.

2.1.1. Основные определения и обозначения

Для получения формализованной модели продукционной системы и изучения ее свойств введем ряд обозначений и определений.

Итак, имеется $P = \{p_1,\dots,p_n\}$ – конечное множество параметров; $V_i = \{v_1^i,\dots,v_{m_i}^i\}$ – конечное множество значений параметра p_i, $i = 1,\dots,n$.

Множество всех теоретически возможных фактов обозначим $F = \{f_1,\dots,f_q\}$, где $q = \sum_{i=1}^{n} m_i$. Имеется также множество выполненных фактов $F_d \subseteq F$, образующее базу данных (рабочее множество) экспертной системы. Там, где это удобно, элементы множества фактов F мы будем также обозначать f_{ij} – факт, утверждающий равенство i-го параметра своему j-му значению («$p_i = v_j^i$»).

Множество всех теоретически возможных начальных фактов для данной продукционной системы обозначим F_0 ($F_0 \subseteq F$). Любое подмножество F_0 будем называть *начальным множеством* фактов. Заданное множество выполненных начальных фактов будем обозначать F_{0d} ($F_{0d} \subseteq F_0$).

Будем рассматривать также множество $F^* = 2^F$, состоящее из всех подмножеств F (булеан). Аналогично определяются $F_0^* = 2^{F_0}$, $F_d^* = 2^{F_d}$, $F_{0d}^* = 2^{F_{0d}}$.

Обозначим $R = \{r_1, ..., r_p\}$ – имеющееся конечное множество правил. Предпосылку и заключение правила будем разделять символом $\to$.

В этих обозначениях набор правил может выглядеть следующим образом:

$$f_{21} \to f_{41}$$
$$f_{22} \wedge f_{11} \to f_{42}$$
$$f_{42} \wedge f_{31} \to f_{51}$$
$$f_{42} \wedge f_{32} \to f_{52} \wedge f_{53} \wedge f_{54}$$

При прямом выводе экспертная система, исходя из некоторого множества $A \in F^*$, путем применения правил формирует множество $B \in F^*$, которое содержит интересующие пользователя факты.

Определение 1. Если по исходному множеству фактов $A \in F^*$ путем применения правил R возможно получение множества $B \in F^*$, то будем считать, что B выводимо из A в R ($A \xrightarrow{R} B$). Такое A называется порождающим множеством (ПМ) для B.

Непосредственно из определения 1 вытекают свойства:

- если $B \subseteq A$, то $A \xrightarrow{R} B$ (в частности, $A \xrightarrow{R} A$);
- если $A \xrightarrow{R} B$ и $C \subset B$, то $A \xrightarrow{R} C$;
- если $A \xrightarrow{R} B$ и $A \subset D$, то $D \xrightarrow{R} B$.

Упорядоченный набор принадлежащих R правил $\vec{r} = \{r_{i_1}, r_{i_2}, ..., r_{i_k}\}$, последовательное применение которых приводит к получению множества B, называется цепочкой вывода, соответствующей выводу $A \xrightarrow{R} B$, или цепочкой правил, осуществляющей вывод $A \xrightarrow{R} B$.

Правило r_0 называется лишним в цепочке $\vec{r}$, осуществляющей вывод $A \xrightarrow{R} B$, если после удаления r_0 из $\vec{r}$ оставшаяся цепочка тем не менее осуществляет вывод $A \xrightarrow{R} B$.

Замечание 1. Нетрудно заметить, что правило r_0 является лишним в цепочке $\vec{r}$ тогда и только тогда, когда каждый выводимый его заключением факт либо содержится в A, либо выводится некоторым предшествующим в цепочке $\vec{r}$ правилом, либо не используется в предпосылках правил, следующих за r_0, и не содержится в множестве $B \setminus A$.

Замечание 2. По определению каждое повторяющееся правило в цепочке вывода является лишним.

Цепочка правил $\vec{r}$, осуществляющая вывод $A \xrightarrow{R} B$, называется минимальной, если она не содержит лишних правил.

Очевидно, множество правил R определяет на F^* бинарное отношение $\xrightarrow{R}$ («выводимость»), обладающее свойствами рефлексивности, транзитивности и дистрибутивности. Как следствие, построив логическое замыкание этого отношения, теоретически можно получить «идеальную» базу знаний, любой вывод в которой осуществляется применением единственного правила. Однако добиться этого на практике трудно в силу ограниченности вычислительных ресурсов систем, которые могут быть для этого задействованы.

При обратном выводе в продукционной системе по данному множеству $B \in F^*$, называемому гипотезой, осуществляется поиск $A \in F^*_{0d}$, такого, что $A \xrightarrow{R} B$. При успешном поиске делается заключение о справедливости всех фактов, содержащихся в B. Таким образом, можно утверждать, что при прямом выводе $A \xrightarrow{R} B$ соответствующая цепочка вывода строится слева направо, при обратном – справа налево.

Прямой вывод в продукционной системе может оказаться неэкономичным. В некоторых случаях во время его выполнения экспоненциально растет число применяемых правил при линейном увеличении их количества в системе (происходит так называемый комбинаторный взрыв (например, [106]). По этой причине существуют классы экспертных систем, в которых за основу принят обратный вывод (например, MYCIN [8]). С другой стороны, и обратный вывод требует минимизации. Такая потребность, в частности, обусловлена тем обстоятельством, что исходная база данных может оказаться объемной, однако неполной. В результате значения ряда начальных параметров не хранятся в БД, но могут интерактивно запрашиваться у пользователя. Необходимо по возможности задавать пользователю лишь значимые вопросы, ответы на которые будут использованы для вывода. Кроме того, количество обращений к БД целесообразно свести к минимуму.

С этой целью введем понятие минимального порождающего множества.

Определение 2. Множество $A \in F^*$ называется минимальным порождающим множеством (МПМ) для данного $B \in F^*$ в R (относительно некоторой цепочки правил $\vec{r}$), если $A \xrightarrow{R} B$, причем существует минимальная цепочка вывода $\vec{r} = \{r_{i_1}, r_{i_2}, \ldots, r_{i_k}\}, r_{i_l} \in R$, выполняющая вывод $A \xrightarrow{R} B$ и не реализующая вывода $A_1 \xrightarrow{R} B$ ни для какого собственного подмножества $A_1 \subset A$.

Лемма 1. Если $A \xrightarrow{R} B$, то A содержит некоторое C , являющееся минимальным порождающим множеством для B в R .

Доказательство. Пусть $\vec{r}$ – соответствующая выводу $A \xrightarrow{R} B$ цепочка правил. Будем считать ее минимальной, удалив при необходимости из нее лишние правила. Пометим факты множества A , которые содержатся также и в B . Далее проведем последовательно указанный вывод, помечая факты, используемые предпосылками каждого применяемого правила цепочки $\vec{r}$. После вывода удалим из A факты, которые изначально были в A , но остались непомеченными. Получим множество $C \subset A$, которое и является для B МПМ относительно цепочки $\vec{r}$. □

Определение 3. Множество $A_0 \in F^*$ называется абсолютным МПМ для данного $B \in F^*$ в R (АМПМ), если $A_0 \xrightarrow{R} B$ в R и B не выводимо в R ни из какого собственного подмножества A_0 .

Замечание 3. Из определений 2-3 следует, что каждое абсолютное МПМ является и относительным (относительно любой минимальной цепочки вывода).

Замечание 4. Абсолютное МПМ не содержит в качестве собственного подмножества ни одного МПМ (относительного или абсолютного), так как существование «меньшего» МПМ противоречит определению 6.

Замечание 5. В общем случае для данного $B \in F^*$ могут существовать более одного МПМ. Этот вывод касается как относительных, так и абсолютных МПМ.

Следствие 1. Если $A \xrightarrow{R} B$, то A содержит некоторое C , являющееся абсолютным минимальным порождающим множеством для B в R .

Доказательство. Справедливость утверждения очевидна ввиду конечности рассматриваемого множества A . □

2.1.2 Эквивалентные преобразования баз знаний

Пусть R_1 и R_2 – два множества правил на одном и том же множестве фактов F . Будем считать, что множество R_1 поглощается множеством R_2 ($R_1 \prec R_2$), если при любых $A, B \in F^*$ из $A \xrightarrow{R_1} B$ следует $A \xrightarrow{R_2} B$.

Множества правил R_1 и R_2 называются эквивалентными ($R_1 \sim R_2$), если выполнены $R_1 \prec R_2$ и $R_2 \prec R_1$.

Теорема 1. Множества правил R_1 и R_2 эквивалентны тогда и только тогда, когда для любого факта $f \in F$ его совокупности АМПМ в R_1 и R_2 совпадают.

Доказательство. Пусть R_1 и R_2 эквивалентны. Рассмотрим произвольный факт $f \in F$. Если $A \in F^*$ – некоторое АМПМ для f в R_1, то в силу эквивалентности множеств правил справедливо и $A \xrightarrow{R_2} \{f\}$. Предположим противное, а именно – что A не является минимальным в R_2. Тогда существует собственное подмножество $A_1 \subset A$ такое, что $A_1 \xrightarrow{R_2} \{f\}$. В силу эквивалентности $R_1 \sim R_2$ имеем и $A_1 \xrightarrow{R_1} \{f\}$, что противоречит минимальности A в R_1.

Предположим теперь, что выполнена вторая часть утверждения теоремы. Покажем, что в этом случае R_1 и R_2 эквивалентны. Пусть для некоторых $A, B \in F^*$ выполнено $A \xrightarrow{R_1} B$. Требуется доказать, что имеет место $A \xrightarrow{R_2} B$. Для этого рассмотрим произвольный факт $f \in B$. Очевидно, $A \xrightarrow{R_1} \{f\}$, причем по следствию 2.1.1.1 множество A содержит некоторое АМПМ A_1 для f в R_1. В силу предположения теоремы A_1 является АМПМ для f и в R_2. Как следствие, $A_1 \xrightarrow{R_2} \{f\}$. Поскольку $A_1 \subseteq A$, то $A \xrightarrow{R_2} \{f\}$. Таким образом, каждый элемент f множества B выводится из A в R_2. Следовательно, $A \xrightarrow{R_2} B$. □

Теорема 2. Пусть R, R_1 и R_2 – множества правил на множестве фактов F. Если при этом $R_1 \sim R_2$, то $R \cup R_1 \sim R \cup R_2$.

Доказательство. Покажем, что если для некоторых $A, B \in F^*$ справедливо $A \xrightarrow{R \cup R_1} B$, то имеет место также и $A \xrightarrow{R \cup R_2} B$. Пусть $\vec{r} = \{r_{i_1}, r_{i_2}, \ldots, r_{i_k}\}$ – цепочка правил, соответствующая исходному выводу. Она может содержать как правила из R, так и из R_1. Рассмотрим ее произвольную подцепочку $\vec{r}_1$, целиком принадлежащую R_1. Если бы последней не существовало, то указанный вывод целиком осуществлялся бы в $R \subset R \cup R_2$, и утверждение было бы уже доказано. Пусть $\vec{r} = \vec{r}_0 \vec{r}_1 \vec{r}_3$. Множество фактов, полученных из A путем применения предшествующей $\vec{r}_1$ подцепочки $\vec{r}_0$, обозначим A_0. Множество фактов, полученных из A_0 путем применения подцепочки $\vec{r}_1$, обозначим A_1. Таким образом, $A_0 \xrightarrow{R_1} A_1$. В силу эквивалентности R_1 и R_2, существует цепочка правил $\vec{r}_2$ из R_2, осуществляющая вывод $A_0 \xrightarrow{R_2} A_1$. Заменим в $\vec{r}$ подцепочку $\vec{r}_1$ цепочкой $\vec{r}_2$. Аналогично поступим с каждой подцепочкой $\vec{r}$, принадлежащей R_1. В результате таких преобразований из $\vec{r}$ получится цепочка правил, осуществляющая вывод $A \xrightarrow{R \cup R_2} B$. □

Для данного множества фактов F преобразование множества правил R_1 в эквивалентное ему множество R_2 будем называть эквивалентным преобразованием базы знаний. Теоремы 1–2 позволяют обосновывать такие преобразования. Рассмотрим правило $r_0 \in R$ вида $f_1 \vee f_2 \to f_3$. Заменим в R это правило двумя правилами: $r_1 : f_1 \to f_3$ и $r_2 : f_2 \to f_3$. Легко заметить, что каждое из двух множеств правил $R_1 = \{r_0\}$ и $R_2 = \{r_1, r_2\}$ порождает для факта f_3 одну и ту же совокупность АМПМ: $\{\{f_1\},\{f_2\}\}$. Отсюда по теореме 1 множества правил R_1 и R_2 эквивалентны. Тогда по теореме 2 множество $R = R \setminus \{r_0\} \bigcup R_1$ эквивалентно множеству $R = R \setminus \{r_0\} \bigcup R_2$. Таким образом, замена правила r_0 в R двумя правилами r_1, r_2 означает эквивалентное преобразование базы знаний. Аналогично обосновывается эквивалентность другого элементарного преобразования: замена правила вида $f_1 \to f_2 \wedge f_3$ двумя правилами $f_1 \to f_2$ и $f_1 \to f_3$.

Рассмотрим вопрос о возможности эквивалентного преобразования базы знаний произвольной структуры в унифицированную (см. начало п. 2.1). К сожалению, в общем случае в принятой нами модели при неизменном множестве фактов такое преобразование невозможно. Этому препятствуют так называемые правила II типа [52] вида $f_1 \to \neg f_2$ и $f_1 \to f_2 \vee f_3$. Первое из них при определенных условиях приводится к виду второго заменой отрицания дизъюнкцией фактов, дополняющих факт f_2 (содержащих все остальные значения того же параметра). Второе правило противоречит принятому в данной работе допущению о том, что каждое следствие правила должно выводить факты. Компромиссным решением является введение нового факта $f_4 = f_2 \vee f_3$ с двумя дополнительными правилами $f_2 \to f_4$ и $f_3 \to f_4$.

Таким образом, процедура преобразования базы знаний общего вида в унифицированную состоит из следующих трех этапов. Преобразуем пропозициональную формулу предпосылки каждого правила в дизъюнктивную нормальную форму, формулу заключения каждого правила – в конъюнктивную нормальную форму. Полученную совокупность правил преобразуем в эквивалентную с помощью двух указанных выше элементарных преобразований. Наконец, на третьем этапе избавимся от правил II типа.

2.1.3. Построение минимальных порождающих множеств

Для рассмотрения вопросов о построении порождающих множеств докажем несколько полезных утверждений.

Лемма 1. Пусть $A \in F^*$ – МПМ для $B \in F^*$; $C \in F^*$ – МПМ для $D \in F^*$. Тогда $A \bigcup C$ содержит МПМ для $B \bigcup D$.

Доказательство. Из условия леммы непосредственно следует $A \bigcup C \xrightarrow{R} B$ и $A \bigcup C \xrightarrow{R} D$. Отсюда $A \bigcup C \xrightarrow{R} B \bigcup D$. Таким образом, по лемме 2.1.1.1 $A \bigcup C$ содержит МПМ для $B \bigcup D$. □

Лемма 2. Пусть $A \in F^*$ – МПМ для $B \in F^*$; B – МПМ для $C \in F^*$. Тогда A содержит МПМ для C.

Доказательство. В силу транзитивности отношения выводимости имеем $A \xrightarrow{R} C$, откуда по лемме 2.1.1.1 следует требуемое утверждение. □

Из лемм 1–2 вытекает

Теорема 1. Пусть множества $A, B, C, D \in F^*$ таковы, что A является МПМ для B, $B \bigcup C$ – МПМ для D. Тогда множество $A \bigcup C$ содержит МПМ для D.

Доказательство. Поскольку по условию A – МПМ для B, в силу леммы 1 множество $A \bigcup C$ содержит МПМ для $B \bigcup C$. Так как по условию $B \bigcup C$ – МПМ для D, то отсюда по лемме 2 получаем, что $A \bigcup C$ содержит МПМ для D. □

Суть теоремы 1 состоит в том, что в минимальном порождающем множестве ($B \bigcup C$) некоторое подмножество (B) можно заменить на МПМ для этого подмножества (A), получив новое МПМ (содержащееся в $A \bigcup C$).

Далее в этом разделе мы будем предполагать, что все правила БЗ имеют указанный в начале п.2.1 унифицированный вид.

Введем понятие структурного расслоения исходного множества правил R на виртуальные слои $R_1, R_2, ..., R_N$, позволяющее упростить построение ряда связанных с продукционной системой алгоритмов, а также изучение их свойств.

Разобьем R на непересекающиеся подмножества, каждое из которых образовано правилами с одним и тем же заключением. Такое разбиение возможно благодаря унифицированным правилам, так как в этом виде каждое правило выводит единственный факт. Обозначим эти подмножества R^{ij} соответственно выводимому в них факту $f_{ij}, i = 1, ..., n; j = 1, ..., m_i$.

Определение 1. Слоем R_k в множестве правил R называется подмножество R, образованное правилами, взятыми по одному из каждого непустого R^{ij}. Два слоя, отличающиеся хотя бы одним правилом, считаются различными.

Будем считать, что цепочка правил $\vec{r}$ принадлежит в R некоторому слою R_k, если в этом слое содержатся все правила цепочки R.

Замечание 1. Слой содержит максимально возможный набор правил из R, выводящих попарно различные факты. Добавление к слою еще одного правила из R нарушило бы это условие.

Замечание 2. Любая цепочка правил, выводящих попарно различные факты, принадлежит некоторому слою.

Замечание 3. Слои могут иметь непустые пересечения. Объединение всех слоев равно R.

Нетрудно заметить, что общее количество слоев N определяется равенством $N = \prod_{k=1}^{k=q} N_k$, где q – количество различных фактов в правых частях множества правил; N_k – количество правил, выводящих факт f_k.

Лемма 3. Если $\vec{r}$ – минимальная цепочка некоторого вывода $A \xrightarrow{R} B$, то существует слой в R, содержащий цепочку $\vec{r}$.

Доказательство. Правила цепочки $\vec{r}$ вследствие ее минимальности выводят попарно различные факты. По этой причине в силу замечания 2 существует слой, набор правил которого содержит эту цепочку. □

Лемма 4. Пусть $A_1 \in F_0^*$ и $A_2 \in F_0^*$ – два различных начальных МПМ для $B \in F^*$. Тогда любые две соответствующие множествам A_1 и A_2 минимальные цепочки вывода $\vec{r}_1$ и $\vec{r}_2$ не могут принадлежать в R одному и тому же слою.

Доказательство. По лемме 3 цепочки $\vec{r}_1$ и $\vec{r}_2$ целиком принадлежат некоторым слоям R_1 и R_2. Покажем, что эти слои всегда разные. В силу условия леммы существует начальный факт g, принадлежащий одному из множеств A_1, A_2, но не принадлежащий другому. Пусть для определенности $g \in A_1$. При этом g не может принадлежать B, так как в этом случае он бы принадлежал и A_2 либо выводился правилами $\vec{r}_2$. В силу того, что g – начальный факт, это невозможно. Тогда, так как A_1 – начальное МПМ для B, то g присутствует в предпосылке хотя бы одного правила $r_g \in \vec{r}_1$. Поскольку при этом $g \notin A_2$, то $r_g \notin \vec{r}_2$ (можно сказать, что g вообще не встречается в цепочке правил $\vec{r}_2$).

Таким образом, некоторое правило r_g содержится в цепочке $\vec{r}_1$ и не содержится в $\vec{r}_2$. Пусть тогда f – факт, выводимый заключением правила r_g. Так как правило r_g не может являться лишним в $\vec{r}_1$, то факт f либо содержится в $B \setminus A_1$, либо используется в предпосылках правил, следующих за r_g в $\vec{r}_1$ (см. замечание 2.1.1.1).

В первом из этих двух вариантов, поскольку f – не начальный, то $f \in B \setminus A_2$, и в цепочке $\vec{r}_2$ обязано быть другое правило, также выводящее факт f. Последнее обстоятельство означает принадлежность рассматриваемых цепочек разным слоям, ведь каждый слой содержит для любого факта лишь одно правило. Во втором варианте факт $f \notin B$ и тогда он либо используется как промежуточный в цепочке $\vec{r}_2$, либо нет. Если используется, то рассуждения аналогичны первому варианту.

Осталось рассмотреть случай, когда f является промежуточным в $\vec{r}_1$ и вообще не встречается в цепочке $\vec{r}_2$. Тогда факт f можно проанализировать с тех же позиций, что и ранее факт g. Итак, существует правило r_f, расположенное в $\vec{r}_1$ правее правила r_g, использующее в предпосылке факт f и не встречающееся в цепочке $\vec{r}_2$. Если рассмотреть эту ситуацию аналогично рассмотренной выше с правилом r_g, то как и ранее завершим доказательство либо продвинемся еще правее в цепочке $\vec{r}_1$.

Продвигаясь таким образом далее, дойдем в $\vec{r}_1$ в силу ее конечности до того правила, которое выводит факт множества $B \setminus A_1$, а не промежуточный (последнее правило в минимальной цепочке всегда таково), и на этом доказательство завершится. □

Теорема 2. В унифицированной базе знаний для любого $B \in F^*$ каждое (относительное) начальное МПМ $A \in F_0^*$ может быть построено с помощью правил некоторого слоя. Кроме того, каждый слой может быть использован для построения лишь единственного начального МПМ.

Утверждение теоремы вытекает непосредственно из лемм 3–4. □

Теорема 2 позволяет свести построение совокупности всех начальных МПМ для данного $B \in F^*$ к построению одного МПМ в отдельном слое. Построив МПМ в каждом слое и удалив из полученных множеств повторяющиеся, получим всю совокупность МПМ. К сожалению, этот алгоритм не является оптимальным, так как несколько различных слоев могут давать одно и то же МПМ. Однако он может явиться основой для построения более эффективных алгоритмов, а также для доказательства их свойств. По этой причине остановимся на вопросе построения МПМ в отдельном слое.

В силу своей структуры отдельный слой правил допускает простую сетевую интерпретацию. Как известно [106, 180], продукционная система может быть представлена так называемым И/ИЛИ-графом, который применяется, например, в механизме управления правилами в известной продукционной системе MYCIN [8]. Не вдаваясь в детали, отметим, что в рассматриваемом случае отдельного слоя И/ИЛИ-граф вырождается в И-граф. Такой граф является классическим ориентированным графом, так как его ИЛИ-составляющая отсутствует в силу единственности правила для каждого факта. С учетом изложенного, каждому факту продукционной системы сопоставим вершину графа. Каждому правилу рассматриваемого слоя будет соответствовать совокупность дуг, ведущих из вершин-фактов предпосылки правила в вершину-следствие правила. Алгоритм построения для некоторого множества $B \in F^*$ МПМ в отдельном слое основывается на теореме 1, и для каждого факта $f \in B$ может быть интерпретирован как обход вершин графа, из которых достижима данная вершина.

Итак, пусть дано $B \in F^*$, для которого требуется построить МПМ в некотором слое. Элементарным МПМ (относительно цепочки правил нулевой длины) является само B. Далее, по теореме 1 каждый не являющийся начальным факт f в текущем МПМ можно заменить на МПМ для f, которое состоит из всех фактов в предпосылке правила для f. После каждой такой замены получается множество, содержащее новое МПМ для B. Можно показать, что в рассматриваемой ситуации, поскольку в слое каждый факт выводится единственным способом либо не выводится вообще, имеет место точное совпадение указанного множества с МПМ. Действительно, выбирая единственным образом для каждого факта предпосылку, соответственно строим минимальную цепочку правил. При этом все факты предпосылок участвуют в выводе. Порядок выбора фактов для замены может диктоваться общей стратегией (например, «в ширину» или «в глубину», если проводить аналогию с графом). Если соответствующий граф не содержит циклов, то через конечное число замен получим множество, являющееся (в силу теоремы 2) единственным в данном слое начальным МПМ для B. В графе ему будет соответствовать множество всех начальных вершин, из которых достижимы вершины, соответствующие фактам множества B. Если при выполнении описанного процесса обнаружится цикл, то такой слой не содержит искомого начального МПМ. Описанный процесс содержит также доказательство усиленного варианта теоремы 1 для отдельного слоя унифицированной базы знаний.

Теорема 3. Пусть при унифицированной базе знаний множества фактов $A, B, C, D \in F^*$ таковы, что A является МПМ для B, $B \cup C$ – МПМ для D, в одном и том же слое правил. Тогда множество $A \cup C$ представляет собой МПМ для D.

Следующая теорема показывает, что совокупность всех абсолютных МПМ может быть получена из построенной в слоях совокупности относительных МПМ.

Теорема 4. Пусть в унифицированной базе знаний для множества фактов $B \in F^*$ построена совокупность всех относительных МПМ $\tilde{A} = \{A_1, ..., A_n\}$ ($A_i \in F^*$, $i = 1, ..., n$). Тогда совокупность всех абсолютных МПМ получается из $\tilde{A}$ исключением таких множеств $A_l \in \tilde{A}$, которые содержат в качестве собственных подмножеств другие $A_{l_k} \in \tilde{A}$.

Доказательство. Проведем указанную в теореме процедуру исключения.

Докажем теперь, что каждое из оставшихся МПМ A_i является абсолютным. Пусть для некоторого A_i это не так, то есть существует $C_1 \subset A_i$ такое, что $C_1 \xrightarrow{R} B$. По лемме 2.1.1.1 множество C_1 содержит некоторое C – МПМ для B . По этой причине C самостоятельно должно присутствовать в совокупности множеств $\tilde{A}$, а в силу $C \subset A_i$ множество A_i в начале доказательства должно было быть исключено из $\tilde{A}$, что и приводит к противоречию.

Докажем теперь, что среди оставшихся после исключения в $\tilde{A}$ множеств A_i содержатся *все* абсолютные МПМ для B . Предположим противное, что некоторое $C \in F^*$ является абсолютным МПМ для B и после процедуры исключения не содержится среди множеств совокупности $\tilde{A}$. Поскольку (см. замечание 1.1.3) оно является и относительным, то должно было содержаться в $\tilde{A}$ до выполнения процедуры исключения. В силу замечания 2.1.1.4 оно не могло быть исключено, что противоречит предположению о его отсутствии.

Теорема 4 доказана. □

Нетрудно заметить, что утверждение теоремы 4 останется справедливым, если речь в его формулировке будет идти о начальных МПМ.

Для проведения обратного вывода, соответствующего некоторой гипотезе $B \in F^*$, достаточно для B построить совокупность начальных абсолютных МПМ. Затем, если какое-либо из них состоит только из выполненных фактов (элементов множества F_{0d}), присвоить этой гипотезе статус выполненной. Экономичность данного метода обусловлена двумя факторами. Во-первых, обращения к БД (или пользователю) ограничиваются фактами из абсолютных МПМ (т. е. лишь минимальным набором фактов, потенциально участвующих в выводах). Во вторых, если хотя бы один факт из некоторого МПМ не выполнен, то нет необходимости проверять остальные факты этого МПМ – оно отвергается сразу. В связи с последним обстоятельством имеет смысл начинать с проверки фактов, принадлежащих пересечениям МПМ. Для ускорения алгоритма проверку истинности фактов, содержащихся в МПМ, можно производить сразу в процессе их построения.

2.1.4. Корректность и верификация баз знаний

Важными свойствами корректных баз знаний продукционных систем являются непротиворечивость, отсутствие циклов, неизбыточность, полнота, достижимость, живость. Опубликованные на этом направлении работы (например, [1, 52]) содержат различные алгоритмы выявления аномалий БЗ. Однако в них не всегда строго формулируются сами понятия этих аномалий. Кратко изложим формализацию понятий, которая вытекает из представленной в данной работе теоретико-множественной модели.

Множество фактов $A \in F^*$ называется противоречивым, если оно содержит более одного факта для некоторого параметра, не являющегося многозначным. В противном случае множество A называется непротиворечивым.

Множество правил R является противоречивым на множестве фактов F, если для некоторых $A, B \in F^*$ таких, что $A \xrightarrow{R} B$ и A непротиворечиво, следует, что B противоречиво.

Методом обнаружения противоречий в БЗ может служить выявление непротиворечивых МПМ для противоречивых множеств фактов.

Множество правил R содержит цикл, если некоторый не начальный факт $f \in F$ содержится в одном из своих относительных МПМ, построенных по R.

База знаний называется избыточной, если для некоторого правила $r \in R$, осуществляющего вывод $A \xrightarrow{R} B (B \setminus A \neq \varnothing)$, этот же вывод возможен в множестве правил $R \setminus \{r\}$.

Чтобы выяснить, является ли правило $r \in R$ избыточным для данной БЗ, достаточно для выводимого его следствием множества фактов B построить в $R \setminus \{r\}$ совокупность абсолютных МПМ и проверить, не содержит ли множество фактов его предпосылки A в качестве подмножества хотя бы одно МПМ.

Множество фактов $A \in F^*$ называется достижимым в R, если оно имеет непустое начальное МПМ.

База знаний называется полной, если каждый факт, относящийся к целевому параметру, является достижимым.

Множество фактов $A \in F^*$ называется живым в R, если оно содержится в МПМ для факта, относящегося к некоторому целевому параметру.

2.2. Дополнительные сведения о бинарных отношениях и решетках

Связанные с решетками и отношениями основные понятия были приведены ранее в п.1.2. Докажем некоторые полезные утверждения, относящиеся к общей теории решеток и отношений.

Как уже отмечалось ранее, $\mathbb{F} = \lambda(F)$ (множество конечных подмножеств некоторого универсума F) является решеткой. В данной главе рассматриваются решетки именно с такой семантикой. Чтобы это подчеркнуть, вместо символов $\le$, $\ge$, $\wedge$ и $\vee$ будем использовать знаки теоретико-множественных операций $\subseteq$, $\supseteq$, $\bigcap$ и $\bigcup$, а элементы решетки будем обозначать большими буквами. Однако использование термина «решетка» сохраняется, поскольку полученные результаты в дальнейшем будут распространены и на другие виды решеток.

Пусть дана решетка $\mathbb{F}$. Пусть c – некоторое свойство (или совокупность свойств), сформулированное для элементов $\mathbb{F}$. Произвольный элемент $A \in \mathbb{F}$ может обладать или не обладать этим свойством.

Замыканием элемента $A \in \mathbb{F}$ относительно свойства c называется элемент $c(A) \in \mathbb{F}$, являющийся наименьшим в подмножестве элементов решетки, которые содержат A и обладают свойством c.

Замыкание произвольного элемента A в приведенной формулировке существует не для любого свойства c. В качестве отрицательного примера можно привести свойство на булеане «множество содержит не менее n элементов» при мощности A, меньшей n. Если такое замыкание существует, то в силу определения оно единственно.

Лемма 1. Пусть $A_1, A_2 \in \mathbb{F}$ и существуют их замыкания $c(A_1), c(A_2)$ относительно некоторого свойства c. Пусть также выполнены условия:

$$c(A_2) \supseteq A_1\,; \tag{1}$$

$$c(A_1) \supseteq A_2\,. \tag{2}$$

Тогда $c(A_1) = c(A_2)$.

Доказательство. Из (1) и определения замыкания $c(A_1)$ следует, что $c(A_2) \supseteq c(A_1) \supseteq A_1$. Соответственно из (2) имеем $c(A_1) \supseteq c(A_2) \supseteq A_2$. Следовательно, $c(A_1) = c(A_2)$. □

Пусть задано множество F, на котором определено бинарное отношение R – совокупность упорядоченных пар элементов этого множества. Поскольку различные отношения на одном и том же F являются множествами с общим универсумом $F \times F$, они образуют ограниченную решетку. Для дальнейших исследований понадобятся некоторые свойства рефлексивно-транзитивного замыкания отношения.

Пусть даны элементы $a, b \in F$. Если существует упорядоченный набор элементов $\vec{r} = (b_1, \ldots, b_m)$ ($b_1, \ldots, b_m \in F$, $0 \le m < \infty$) такой, что $(a, b_1), (b_1, b_2), \ldots, (b_m, b) \in R$ (в случае $(a, b) \in R$ предполагается $m = 0$), то указанный набор $\vec{r} \in R$ будем называть транзитивной цепочкой (длины m), соединяющей a и b.

Лемма 2. Пусть $a, b \in F$ и R_1 – транзитивное отношение на F. Тогда, если существует транзитивная цепочка $\vec{r}_{ab} \in R_1$, соединяющая a и b, то $(a, b) \in R_1$.

Утверждение леммы очевидным образом вытекает из определения 2 и транзитивности R_1. □

Справедлива также следующая лемма.

Лемма 3. Для данного отношения R на F рефлексивно-транзитивное замыкание представляет собой объединение R^* множества всех рефлексивных пар (a, a) ($a \in F$) с множеством всех упорядоченных пар $a, b \in F$, для которых существует соединяющая их транзитивная цепочка в R.

Доказательство. Очевидно, что определенное в условии леммы отношение R^* является рефлексивным. Покажем, что оно транзитивно. Пусть $(a, b), (b, c) \in R^*$. Тогда по условию леммы существуют транзитивные цепочки $\vec{r}_{ab}, \vec{r}_{bc} \in R$, соединяющие соответственно (a, b) и (b, c). Рассмотрим цепочку $\vec{r}$, составленную последовательно из $\vec{r}_{ab}, b, \vec{r}_{bc}$. Очевидно, $\vec{r} \in R$ соединяет a и c. Отсюда следует, что $(a, c) \in R^*$, то есть определенное в лемме отношение транзитивно.

Покажем, наконец, что оно наименьшее. Пусть R_1 – другое транзитивное отношение, содержащее R. Пусть также $(a, b) \in R^*$. Тогда, поскольку существует соединяющая (a, b) транзитивная цепочка $\vec{r}_{ab} \in R \subseteq R_1$, из леммы 2 следует, что $(a, b) \in R_1$. Следовательно, $R^* \subseteq R_1$. □

Замечание 1. Из леммы 3 непосредственно следует свойство: если $R_1 \subseteq R_2$, то $R_1^* \subseteq R_2^*$.

Справедливы также следующие утверждения.

Лемма 4. Для любых бинарных отношений R_1 и R_2, заданных на некотором общем множестве F, выполнено $R_1^* \cup R_2^* \subseteq (R_1 \cup R_2)^*$.

Доказательство. Поскольку $R_1 \subseteq R_1 \bigcup R_2$ и $R_2 \subseteq R_1 \bigcup R_2$, то в силу замечания 1 имеем $R_1^* \subseteq (R_1 \bigcup R_2)^*$ и $R_2^* \subseteq (R_1 \bigcup R_2)^*$, откуда сразу следует утверждение леммы. □

Замечание 2. Доказанное в лемме включение, вообще говоря, не является равенством. Например, при $R_1 = \{(a,b)\}$, $R_2 = \{(b,c)\}$ имеем $R_1^* \bigcup R_2^* = \{(a,b),(b,c)\}$, но $(R_1 \bigcup R_2)^* = \{(a,b),(b,c),(a,c)\}$.

Замечание 3. Из приведенного примера также видно, что отношение $R_1^* \bigcup R_2^*$ в общем случае не транзитивно.

Лемма 5. Для произвольных R_1 и R_2 справедливо $(R_1 \bigcap R_2)^* \subseteq R_1^* \bigcap R_2^*$.

Доказательство. Пусть $(a,b) \in (R_1 \bigcap R_2)^*$. Тогда по лемме 3 существует транзитивная цепочка $\vec{r} = (b_1,...,b_m)$ такая, что $(a,b_1),(b_1,b_2),...,(b_m,b) \in R_1 \bigcap R_2$. Это означает также $(a,b_1),(b_1,b_2),...,$ $(b_m,b) \in R_1$ и $(a,b_1),(b_1,b_2),...,(b_m,b) \in R_2$. Отсюда снова по лемме 3 имеем $(a,b) \in R_1^*$ и $(a,b) \in R_2^*$, что и означает справедливость леммы 5. □

Замечание 4. В лемме 5 включение в общем случае нельзя заменить равенством. При $R_1 = \{(a,b),(b,c)\}$, $R_2 = \{(a,d),(d,c)\}$ получим $(R_1 \bigcap R_2)^* = \varnothing$, в то время как $R_1^* \bigcap R_2^* = \{(a,c)\}$.

Замечание 5. В отличие от случая объединения, отношение $R_1^* \bigcap R_2^*$ при любых R_1 и R_2 транзитивно. Чтобы это установить, предположим противное. Пусть для некоторых $a,b,c \in F$ справедливо $(a,c) \notin R_1^* \bigcap R_2^*$, несмотря на $(a,b) \in R_1^* \bigcap R_2^*, (b,c) \in R_1^* \bigcap R_2^*$. В этом случае имеем $(a,b) \in R_1^*, (b,c) \in R_1^*$ и, следовательно, $(a,c) \in R_1^*$. Аналогично $(a,b) \in R_2^*, (b,c) \in R_2^*$ и поэтому $(a,c) \in R_2^*$. Таким образом, имеем $(a,c) \in R_1^* \bigcap R_2^*$, что противоречит сделанному предположению. □

Следствие 1. Таким образом, из лемм 4 и 5 вытекают соотношения:

$$(R_1 \bigcap R_2)^* \subseteq R_1^* \bigcap R_2^* \subseteq R_1^* \subseteq R_1^* \bigcup R_2^* \subseteq (R_1 \bigcup R_2)^*;$$

$$(R_1 \bigcap R_2)^* \subseteq R_1^* \bigcap R_2^* \subseteq R_2^* \subseteq R_1^* \bigcup R_2^* \subseteq (R_1 \bigcup R_2)^*.$$

2.3. Понятие LP-структуры. Логические отношения

В настоящем разделе вводится понятие LP-структуры, которое обобщает теоретико-множественную модель продукционной системы, представленную в предыдущем пункте, и выводит ее на более абстрактный уровень описания. Такое обобщение достигается использованием математических решеток в качестве основы алгебраической системы. На решетке задаются бинарные отношения, содержащие семантику продукционно-логического вывода. Доказывается существование логического замыкания произвольного бинарного отношения на решетке, что позволяет определить понятие эквивалентного отношения.

Напомним, что бинарное отношение R на решетке называется $\bigcup$-дистрибутивным (в данной главе – просто дистрибутивным), если из $(A,B_1),(A,B_2)\in R$ следует $(A,B_1\bigcup B_2)\in R$. В частности, определяющее решетку отношение включения $\supseteq$ является дистрибутивным.

Отношение на решетке называется продукционно-логическим (в рамках данной главы – просто логическим), если оно содержит $\supseteq$, дистрибутивно и транзитивно. Под LP-структурой подразумевается решетка с заданным на ней логическим отношением (см. также п. 1.2). Логическим замыканием произвольного бинарного отношения R называется наименьшее логическое отношение, содержащее R. Отсюда следует, что отношение включения $\supseteq$ является наименьшим логическим отношением. Рассматриваемый тип отношений относится к так называемым *монотонным* отношениям. В следующей лемме показано, что при определении логического отношения условие дистрибутивности можно заменить условием монотонности.

Лемма 1. Бинарное отношение R на решетке $\mathbb{F}$ является логическим тогда и только тогда, когда оно содержит $\supseteq$, транзитивно и удовлетворяет следующему условию монотонности:

$$\text{если } (A,B)\in R,\ \text{то } (A,A\bigcup B)\in R \quad (\forall A,B\in\mathbb{F}). \tag{1}$$

Доказательство. Предположим вначале, что отношение R – логическое. Тогда при $(A,B)\in R$ с использованием соотношения $(A,A)\in R$ и свойства дистрибутивности R находим, что верно и $(A,A\bigcup B)\in R$.

Докажем теперь противоположную часть утверждения леммы. Пусть R содержит $\supseteq$, транзитивно и таково, что выполнено (1). Покажем, что в этом случае R дистрибутивно. Для этого рассмотрим пары $(A,B_1),(A,B_2)\in R$. По условию (1) из $(A,B_1)\in R$ следует $(A,A\bigcup B_1)\in R$. Пользуясь транзитивностью отношения R, для $A\bigcup B_1\supseteq A$ и $(A,B_2)\in R$ имеем $(A\bigcup B_1,B_2)\in R$. Применяя к этому соотношению условие вида (1), получим $(A\bigcup B_1,A\bigcup B_1\bigcup B_2)\in R$. Отсюда, так как $(A,A\bigcup B_1)\in R$, в силу транзитивности R имеем $(A,A\bigcup B_1\bigcup B_2)\in R$. Наконец, учитывая тот факт, что $(A\bigcup B_1\bigcup B_2,B_1\bigcup B_2)\in R$ и применяя еще раз транзитивность отношения R, приходим к соотношению $(A,B_1\bigcup B_2)\in R$, которое означает дистрибутивность R. □

Доказанная лемма позволяет дать иное, эквивалентное определение продукционно-логического отношения. Новое определение ориентировано на последовательный логический вывод, в то время как исходное предоставляет больше возможностей для моделирования логического вывода с максимальным параллелизмом.

Прежде чем решить вопрос о существовании логического замыкания произвольного отношения, выясним предварительно структуру логических отношений (подобно выяснению структуры транзитивного отношения – см. п. 2.2).

Пусть задано некоторое отношение R на решетке $\mathbb{F}$ и выбраны два элемента $A,B \in \mathbb{F}$. Пусть существует конечное множество пар $\{(A_i,B_i) \mid i=1,...,p\}$, где для любого i справедливо $A_i = B_i$, $A_i \in \mathbb{F}$ либо $(A_i,B_i) \in R$. Если при этом $A \supseteq A_i, 1 \le i \le p$ и $\bigcup_i B_i \supseteq B$, то будем считать, что упорядоченная пара элементов (A,B) дистрибутивно связана отношением R.

Замечание 1. Очевидно, что если $A \supseteq B$ либо $(A,B) \in R$, то (A,B) дистрибутивно связана отношением R.

Замечание 2. Нетрудно также показать, что если пары (A,B) и (C,D) дистрибутивно связаны R, то и пара $(A \cup C, B \cup D)$ аналогичным образом связана отношением R.

Лемма 2. Если R – логическое отношение на $\mathbb{F}$ и пара (A,B) дистрибутивно связана R, то $(A,B) \in R$.

Доказательство. Поскольку логическое отношение R содержит $\supseteq$, то из определения 3 следует $(A,A_i) \in R$ при $1 \le i \le p$. Учитывая, что R рефлексивно, из того же определения имеем $(A_i,B_i) \in R$. Пользуясь далее транзитивностью R, получим $(A,B_i) \in R$. Наконец, в силу дистрибутивности отношения R находим, что справедливо $(A,\bigcup_i B_i) \in R$ и, соответственно, $(A,B) \in R$. □

Определение 1. Пусть R – произвольное отношение на $\mathbb{F}$ и выбраны два элемента $A,B \in \mathbb{F}$. Пусть также существует упорядоченный набор элементов $\vec{r}_{AB} = (B_1,...,B_m)$ ($B_1,...,B_m \in \mathbb{F}$, $0 < m < \infty$), такой, что в последовательности $(B_0,B_1),(B_1,B_2),...,(B_m,B_{m+1})$, где $B_0 = A$, $B_{m+1} = B$, каждая пара дистрибутивно связана R (если дистрибутивно связана сама пара (A,B), полагаем $m = 0$). Тогда указанный набор $\vec{r}_{AB}$ будем называть логической цепочкой (длины m), соединяющей A и B. Пару (A,B) при этом будем называть логически связанной отношением R и обозначать этот факт $A \xrightarrow{R} B$.

Отметим, что здесь пары $(A,B_1),(B_1,B_2),...,(B_m,B)$ не обязаны принадлежать R.

Лемма 3. Если R – логическое отношение на решетке $\mathbb{F}$ и пара (A,B) логически связана отношением R, то $(A,B)\in R$.

Доказательство. Утверждение данной леммы автоматически вытекает из определения 1, леммы 2 и транзитивности логического отношения R. □

В целях наглядности ряда последующих доказательств, совокупность пар элементов, участвующих в определении логической цепочки $\vec{r}_{AB}=(B_1,\dots,B_m)$, удобно представлять матрицей с переменным количеством элементов в столбцах (квазиматрицей – [133]):

$$M_{AB}=\begin{pmatrix} (A_{1,1},B_{1,1}) & (A_{1,2},B_{1,2}) & \dots & (A_{1,m+1},B_{1,m+1}) \\ (A_{2,1},B_{2,1}) & (A_{2,2},B_{2,2}) & \dots & (A_{2,m+1},B_{2,m+1}) \\ \dots & \dots & \dots & \dots \\ (A_{p_1,1},B_{p_1,1}) & (A_{p_2,2},B_{p_2,2}) & \dots & (A_{p_{m+1},m+1},B_{p_{m+1},m+1}) \end{pmatrix}, \qquad (2)$$

где каждый столбец с номером k образован парами $\{(A_{i,k},B_{i,k})\mid 1\le i\le p_k\}$, осуществляющими дистрибутивную связь (B_{k-1},B_k) в цепочке $\vec{r}_{AB}$ $(1\le k\le m+1)$. Согласно определению 3, справедливы следующие соотношения:

$$B_{k-1}\supseteq A_{i,k}\ (1\le i\le p_k)\ \text{и}\ \bigcup_{1\le i\le p_k} B_{i,k}\supseteq B_k\ \ (1\le k\le m+1); \qquad (3)$$

$$(A_{i,k},B_{i,k})\in R\ \text{либо}\ A_{i,k}=B_{i,k}\ \ (1\le i\le p_k; 1\le k\le m+1). \qquad (4)$$

Будем называть ее матрицей вывода (для данной пары (A,B)). Очевидно, что произвольная перестановка пар внутри любого столбца этой матрицы сохраняет свойства (3)-(4). Нетрудно также заметить, что при заданной матрице M_{AB} вместо элементов B_{k-1} логической цепочки, соединяющей (A,B), можно выбрать $\breve{B}_{k-1}=\bigcup\limits_{1\le i\le p_k} A_{i,k}$ $(1\le k\le m+1)$. Тогда при $k=1$ вместо A можно получить меньший исходный элемент $\breve{B}_0\subseteq A$. Рассмотрим некоторые свойства матриц вывода, которые понадобятся в данной главе.

Лемма 4. Пусть дана некоторая матрица вывода M_{AB}, соответствующая логической цепочке $\vec{r}_{AB}=(B_1,\dots,B_m)$, и, как следствие, удовлетворяющая при данной паре (A,B) условиям (3)–(4). Пусть при некоторых i,k справедливо $B_{i,k}\supseteq X$, $X\in\mathbb{F}$. Тогда матрица M'_{AB}, полученная из M_{AB} дописыванием к $k+1$ столбцу рефлексивного элемента (X,X), определяет логическую цепочку $\vec{r}\,'_{AB}=(B_1,\dots,B_{k-1},B_k\cup X,B_{k+1}\cup X,B_{k+2},\dots,B_m)$, также реализующую связь $A\xrightarrow{R}B$.

Доказательство. Достаточно для новой матрицы M'_{AB} проверить справедливость условий (3)–(4) при тех значениях k, при которых имеются изменения в логической цепочке и ее матрице вывода. Следуя этой логике, поскольку $\bigcup_{1\le i\le p_k} B_{i,k} \supseteq B_k$ и при некотором k выполнено $B_{i,k} \supseteq X$, получаем $\bigcup_{1\le i\le p_k} B_{i,k} \supseteq B_k \cup X$. Далее, так как $B_k \supseteq A_{i,k+1}$, то $B_k \cup X \supseteq A_{i,k+1}$ и $B_k \cup X \supseteq X$. Наконец, из $\bigcup_{1\le i\le p_{k+1}} B_{i,k+1} \supseteq B_{k+1}$ следует $\bigcup_{1\le i\le p_{k+1}} B_{i,k+1} \cup X \supseteq B_{k+1} \cup X$, а из $B_{k+1} \supseteq A_{i,k+2}$ следует, что $B_{k+1} \cup X \supseteq A_{i,k+2}$. Таким образом, условие (3) выполнено. Условие (4) также сохраняется, так как к матрице добавляется рефлексивный элемент. □

Следствие 1. Элементарной индукцией доказывается, что указанному в лемме 4 преобразованию можно подвергнуть любое количество столбцов подряд, начиная с $k+1$-го. Новая матрица при этом также будет осуществлять логическую связь $A \xrightarrow{R} B$.

Лемма 5. Пусть дана матрица вывода вида (2), удовлетворяющая при некоторой паре (A, B) условиям (3)–(4). Преобразуем ее следующим образом. Будем последовательно слева направо просматривать ее столбцы. Предположим, что в некотором столбце встретится нерефлексивный элемент $(A_{i,k}, B_{i,k})$ такой, что $B_{i,k}$ уже появлялся ранее в качестве правой части некоторой пары. Если это произойдет в том же столбце, то удалим из столбца элемент $(A_{i,k}, B_{i,k})$. Если первоначальное появление было в некотором предыдущем столбце l, то к каждому столбцу с $l+1$ по k припишем дополнительный элемент $(B_{i,k}, B_{i,k})$, после чего удалим элемент $(A_{i,k}, B_{i,k})$.

В результате получим новую матрицу вывода, также удовлетворяющую условиям (3)–(4) при той же паре (A, B).

Доказательство. Удаление элемента с повторной правой частью в том же столбце не нарушает условия (3). Действительно, $B_{k-1} \supseteq A_{i,k}$ для всех $1 \le i \le p$ означает, что B_{k-1} содержит и любое меньшее количество элементов $A_{i,k}$. С другой стороны, после удаления $B_{i,k}$ объединение $\bigcup B_{i,k}$ не изменится, так как $B_{i,k}$ встречается в другой паре этого же столбца. При удалении элемента матрицы не может нарушиться и условие (4). Если же пара с элементом $B_{i,k}$ появилась в одном из предыдущих столбцов, то предварительное применение следствия 1 приводит к ранее рассмотренному случаю, если предыдущим появлением элемента $B_{i,k}$ считать добавленную к текущему столбцу пару $(B_{i,k}, B_{i,k})$. □

Следствие 2. Если справедливо $A \xrightarrow{R} B$, то эту связь можно реализовать подмножеством R, в котором все правые части нерефлексивных пар уникальны. Справедливость этого утверждения непосредственно вытекает из леммы 5. □

Рассмотрим свойства некоторых операций над логическими цепочками.

Пусть существуют логические цепочки $\vec{r}_{AB}$ и $\vec{r}_{BC}$, соединяющие отношением R соответственно пары (A,B) и (B,C). Цепочку $\vec{r}_{AC} = \vec{r}_{AB}\vec{r}_{BC}$, составленную последовательно из $\vec{r}_{AB}, B, \vec{r}_{BC}$, будем называть транзитивным объединением цепочек $\vec{r}_{AB}$ и $\vec{r}_{BC}$.

Замечание 3. Очевидно, $\vec{r}_{AC} = \vec{r}_{AB}\vec{r}_{BC}$ является логической цепочкой, реализующей связь $A \xrightarrow{R} C$. Таким образом, если имеют место $A \xrightarrow{R} B$ и $B \xrightarrow{R} C$, то справедливо и $A \xrightarrow{R} C$.

Пусть выбраны элементы $A, B, C, D \in \mathbb{F}$, такие, что существуют логические цепочки $\vec{r}_{AB} = (B_1, \ldots, B_m)$ и $\vec{r}_{CD} = (D_1, \ldots, D_p)$, соединяющие соответственно (A,B) и (C,D). Предположим для определенности, что $m \geq p$. Суммой цепочек $\vec{r}_{AB}$ и $\vec{r}_{CD}$ будем называть цепочку $\vec{r}_{AB} + \vec{r}_{CD} = (B_1 \cup D_1, \ldots, B_p \cup D_p, B_{p+1} \cup D_p, \ldots, B_m \cup D_p)$.

Замечание 4. Из определения 1 и замечания 2 легко следует, что $\vec{r}_{AB} + \vec{r}_{CD}$ является логической цепочкой, реализующей связь $A \cup B \xrightarrow{R} C \cup D$.

Пусть даны логические цепочки $\vec{r}_{AB} = (B_1, \ldots, B_m)$ и $\vec{r}_{AC} = (C_1, \ldots, C_p)$, соответствующие связям $A \xrightarrow{R} B$ и $A \xrightarrow{R} C$. Дистрибутивным объединением цепочек $\vec{r}_{AB}$ и $\vec{r}_{AC}$ назовем цепочку $\vec{r}_{A(B+C)} = \vec{r}_{AB} + \vec{r}_{AC}$.

Замечание 5. Поскольку дистрибутивное объединение – это частный случай суммы, то $\vec{r}_{A(B+C)}$ является логической цепочкой, реализующей логическую связь $A \xrightarrow{R} B \cup C$. Таким образом, из $A \xrightarrow{R} B$ и $A \xrightarrow{R} C$ следует $A \xrightarrow{R} B \cup C$.

Введенные над логическими цепочками операции легко распространяются на конечное число операндов с сохранением описанных свойств.

Для дальнейших исследований потребуется следующая лемма.

Лемма 6. Пусть R – произвольное отношение, заданное на решетке $\mathbb{F}$. Пусть также для некоторой пары элементов (A,B) существует конечное множество логически связанных пар $\{(A_i, B_i),\ i = 1, \ldots, n\}$, причем $A \supseteq A_i,\ 1 \leq i \leq n$ и $\bigcup\limits_i B_i \supseteq B$. Тогда имеет место связь $A \xrightarrow{R} B$.

Доказательство. Пусть $\vec{r}_i$ – логическая цепочка, связывающая пару (A_i, B_i) ($1 \leq i \leq n$). Обозначим $\tilde{A} = \bigcup\limits_i A_i$ и $\tilde{B} = \bigcup\limits_i B_i$. Тогда, согласно замечанию 4, цепочка $\vec{r} = \sum \vec{r}_i$ логически связывает пару $(\tilde{A}, \tilde{B})$ в R. Отсюда в силу вложений $A \supseteq \tilde{A}$ и $\tilde{B} \supseteq B$ следует, что пара (A,B) логически связана цепочкой, составленной последовательно из $\tilde{A},\vec{r},\tilde{B}$. □

Перейдем к доказательству существования логического замыкания.

Теорема 1. Для произвольного отношения R на решетке $\mathbb{F}$ логическое замыкание существует и представляет собой множество $\xrightarrow{R}$ всех упорядоченных пар $A,B \in \mathbb{F}$, логически связанных отношением R.

Доказательство. В силу замечания 1, отношение $\xrightarrow{R}$ содержит $\supseteq$ и R. Из замечания 5 следует его дистрибутивность, а из замечания 3 – транзитивность. Таким образом, отношение $\xrightarrow{R}$ – логическое. Покажем, что это наименьшее логическое отношение для R. Пусть R_1 – любое другое логическое отношение, содержащее R. Пусть также $A \xrightarrow{R} B$. Тогда из $R \subseteq R_1$ следует $A \xrightarrow{R_1} B$. Отсюда по лемме 3 получаем, что $(A,B) \in R_1$. Следовательно, $\xrightarrow{R} \subseteq R_1$. □

Таким образом, удалось показать, что для произвольного бинарного отношения на решетке логическое замыкание существует. Этот факт позволяет ввести понятие эквивалентных отношений, что в свою очередь, может явиться основой для реализации формальных преобразований и оптимизации продукционных баз знаний.

2.4. Эквивалентные преобразования

Продукционно-логические отношения служат математической основой решения прикладных задач, связанных с автоматизацией логического вывода и логическим программированием. В связи с этим обстоятельством возникают также вопросы автоматического преобразования отношений, при которых логическое замыкание остается без изменения. Такие преобразования могут быть использованы для приведения базы знаний к некоторому каноническому виду, который удобен для исследования и реализации. Эти вопросы в более узкой постановке освещались в п. 2.1.2.

Напомним, что два отношения R_1 и R_2, заданные на общей решетке $\mathbb{F}$, называются логически эквивалентными (в контексте данной главы просто эквивалентными), если их логические замыкания совпадают. Для таких отношений используется обозначение $R_1 \sim R_2$.

Лемма 1. Пусть R – бинарное отношение на решетке $\mathbb{F}$ и $A \xrightarrow{R} B$. Тогда отношение $R' = R \cup \{(A,B)\}$ логически эквивалентно R.

Доказательство. Заметим вначале, что по определению любое логическое отношение является собственным логическим замыканием. Такой же вывод в силу теоремы 2.3.1[1] относится и к отношению $\xrightarrow{R}$. Далее, из определения 2.3.4[2] очевидным образом следует, что для любых $R_1 \subseteq R_2$ справедливо $\xrightarrow{R_1} \subseteq \xrightarrow{R_2}$. В рассматриваемом случае по построению отношения R' выполнены включения $R \subseteq R' \subseteq \xrightarrow{R}$. Переходя к логическим замыканиям и учитывая изложенное выше, получаем, что отношения R и R' имеют общее логическое замыкание $\xrightarrow{R}$. □

Лемма 1 обосновывает такие эквивалентные преобразования отношения, которые состоят в добавлении или исключении пары, логически связанной меньшим отношением.

Теорема 1. Пусть R_1, R_2, R_3, R_4 – отношения на $\mathbb{F}$. Если при этом $R_1 \sim R_2$ и $R_3 \sim R_4$, то $R_1 \bigcup R_3 \sim R_2 \bigcup R_4$.

Доказательство. Согласно теореме 2.3.1, достаточно доказать равенство двух множеств $\xrightarrow{R_1 \cup R_3} = \xrightarrow{R_2 \cup R_4}$. Пусть вначале $A \xrightarrow{R_1 \cup R_3} B$. Это означает, что существует логическая цепочка $\vec{r}_{AB} = (B_1, \ldots, B_m)$, соединяющая (A, B) посредством отношения $R_1 \bigcup R_3$. Для произвольного k ($1 \le k \le m+1$) рассмотрим пару (B_{k-1}, B_k) (как обычно, для единообразия полагаем $B_0 = A$, $B_{m+1} = B$). По определению логической цепочки эта пара дистрибутивно связана отношением $R_1 \bigcup R_3$. Отсюда следует (определение 2.3.3), что существует конечное множество пар $\{(A_i^k, B_i^k),\ i = 1, \ldots, n\}$ таких, что $A_i^k = B_i^k$ либо $(A_i^k, B_i^k) \in R_1 \bigcup R_3$, причем $B_{k-1} \supseteq A_i^k$, $1 \le i \le n$ и $\bigcup\limits_i B_i^k \supseteq B_k$.

Если $A_i^k = B_i^k$, то пара (A_i^k, B_i^k) логически связана любым отношением R. Пусть $(A_i^k, B_i^k) \in R_1 \bigcup R_3$. Предположим для определенности, что эта пара принадлежит R_1. Отсюда по условию теоремы ($R_1 \sim R_2$) имеем $A_i^k \xrightarrow{R_2} B_i^k$. Таким образом, все пары (A_i^k, B_i^k) логически связаны отношением $R_2 \bigcup R_4$. По утверждению леммы 2.3.6 получаем, что и пара (B_{k-1}, B_k) логически связана отношением $R_2 \bigcup R_4$. Поскольку k – произвольное, в силу транзитивности отношения $\xrightarrow{R_2 \cup R_4}$ приходим к выводу, что и пара (A, B) логически связана отношением $R_2 \bigcup R_4$, то есть $A \xrightarrow{R_2 \cup R_4} B$.

Таким образом, доказано включение $\xrightarrow{R_1 \cup R_3} \subseteq \xrightarrow{R_2 \cup R_4}$. Обратное включение устанавливается аналогично. □

Следствие 1. Пусть R_1, R_2, R – отношения на $\mathbb{F}$. Если при этом $R_1 \sim R_2$, то $R_1 \bigcup R \sim R_2 \bigcup R$.

---

[1] Напомним, что в силу принятых в работе обозначений это теорема 1 из п. 2.3.

[2] Соответственно – определение 4 из п. 2.3.

Теорема 1 и следствие 1 обосновывают принцип локальности эквивалентных преобразований: заменяя некоторое подмножество данного отношения на эквивалентное множество, получаем общее эквивалентное отношение. Эти результаты могут быть использованы для приведения отношений к каноническому виду.

Лемма 2. Пусть R – отношение на решетке $\mathbb{F}$. Пусть также $(A,B)\in R$, причем $B=\bigcup_i B_i$ – конечное объединение элементов $B_i \in \mathbb{F}$ $(1\le i\le n)$. Тогда отношение R', полученное из R заменой пары (A,B) совокупностью всех пар (A,B_i), эквивалентно R.

Доказательство. Введем обозначения $R_1=\{(A,B)\}$; $R_2 = \{(A,B_1),...,(A,B_n)\}$ и $R^- = R\setminus R_1$. Таким образом, $R = R^-\bigcup R_1$; $R' = R^-\bigcup R_2$. Рассмотрим логические замыкания $\xrightarrow{R_1}$ и $\xrightarrow{R_2}$ отношений R_1 и R_2 соответственно. Поскольку по определению $\xrightarrow{R_1}$ содержит отношение $\supseteq$, то в силу своей транзитивности оно включает и R_2, то есть $R_2 \subseteq \xrightarrow{R_1}$. С другой стороны, отношение $\xrightarrow{R_2}$ из-за дистрибутивности включает R_1. Таким образом, R_1 и R_2 как элементы решетки отношений удовлетворяют лемме 2.2.1. По этой лемме имеем $\xrightarrow{R_1} = \xrightarrow{R_2}$. Последний факт означает, что отношения R_1 и R_2 эквивалентны. Применяя теперь к R_1, R_2, R^- следствие 1, получим, что $R \sim R'$. □

Теорема 2. Если в отношении R на решетке $\mathbb{F}$ каждую пару вида (A,B), где $B=\bigcup_i B_i$ – конечное объединение элементов $B_i\in\mathbb{F}$ $(1\le i\le n)$, заменить совокупностью пар $\{(A,B_1),...,(A,B_n)\}$, то полученное отношение R' будет эквивалентно исходному R.

Доказательство. Согласно лемме 2, описанная замена одной пары (соответственно и конечного их числа) приводит к эквивалентному отношению. В этой связи сомнения может вызывать лишь замена в R бесконечного подмножества пар. Для выяснения этого вопроса рассмотрим логические замыкания $\xrightarrow{R}$ и $\xrightarrow{R'}$ отношений R и R'. Пусть $A\xrightarrow{R}B$. Тогда существует логическая цепочка в R, соединяющая (A,B). По своему определению любая логическая цепочка формируется на основе некоторого конечного числа пар $(A_i,B_i)\in R,\ i=1,...,n$ – элементов матрицы вывода (2.3.2). Эту совокупность пар можно рассматривать как конечное отношение на решетке $\mathbb{F}$. Обозначим его R_1 и проведем для него все описанные в условии теоремы замены. Полученное отношение обозначим R_2. Как замечено в начале доказательства, $R_1 \sim R_2$. Как следствие, существует логическая цепочка в $R_2 \subseteq R'$, связывающая пару (A,B). Отсюда, $A\xrightarrow{R'}B$. Таким образом, $\xrightarrow{R}\subseteq\xrightarrow{R'}$.

Для доказательства обратного включения рассмотрим упорядоченную пару $A \xrightarrow{R'} B$. Существует логическая цепочка в R', образованная посредством конечного числа пар $(A_j, B_j) \in R'$, $j = 1, \ldots, m$, соединяющая (A, B). Множество пар $\{(A_j, B_j)\}$ получено применением описанной в условии теоремы процедуры к некоторому конечному $R_1 \subseteq R$ и содержится в соответствующем конечном множестве $R_2 \subseteq R'$, полученном из R_1. Дальнейшие рассуждения аналогичны первой части доказательства. □

В качестве применения полученных результатов рассмотрим некоторый частный случай. Отношение R на точечной решетке $\mathbb{F}$ называется каноническим, если оно задано множеством пар вида (A, a), где $A \in \mathbb{F}$, a – точка в $\mathbb{F}$.

Следствие 2. Согласно теореме 2, для любого отношения на точечной решетке существует эквивалентное ему каноническое отношение.

2.5. Логическая редукция

В приложениях естественным образом возникает вопрос о минимизации представления логического отношения R. Решение такой задачи актуально не только с точки зрения экономии занимаемых ресурсов, но и с позиций выполнения ключевого правила «хорошего» программирования – по возможности избегать дублирования кода и данных. Очевидным решением является такой способ, когда в памяти компьютера хранится лишь уникальная часть информации о данном отношении, а остальная информация может быть получена с использованием общих свойств логических отношений. Под уникальной частью подразумевается подобранное по определенным критериям отношение $R_0 \subset R$, из которого R получается построением логического замыкания.

Логической редукцией данного отношения R на решетке $\mathbb{F}$ называется эквивалентное ему минимальное отношение R_0. Минимальность в данном определении понимается в смысле частично упорядоченных множеств. Это означает, что исключив из R_0 хотя бы одну пару, получим отношение, не эквивалентное R.

Вначале выясним, не является ли логическое замыкание данного отношения R транзитивным замыканием некоторого другого отношения $\tilde{R} \supseteq R$. Положительный ответ на этот вопрос будет полезен для разработки эффективных алгоритмов построения логического замыкания и редукции. Он позволит свести исследование вопросов о нахождении логического замыкания и редукции отношений к рассмотрению соответствующих свойств транзитивных отношений.

Для произвольного бинарного отношения R на решетке $\mathbb{F}$ рассмотрим отношение $\tilde{R}$, построенное по R последовательным выполнением следующих действий (шагов):

• добавить к R все пары вида (A,A), где $A \in \mathbb{F}$ (рефлексивные пары), и обозначить новое отношение R_1;

• добавить к R_1 всевозможные пары (A,B) с элементами вида $A = \bigcup_i A_i,\ B = \bigcup_i B_i$, где все (A_i,B_i) ($i = 1,\dots,n$) принадлежат R_1;

• объединить полученное отношение с отношением включения $\supseteq$.

Замечание 1. Пользуясь леммой 2.4.1, нетрудно показать, что отношение $\tilde{R}$ логически эквивалентно R.

Теорема 1. Для произвольного отношения R на решетке $\mathbb{F}$ логическое замыкание $\xrightarrow{R}$ представляет собой транзитивное замыкание $\tilde{R}^*$ соответствующего отношения $\tilde{R}$.

Доказательство. По теореме 2.3.1 логическое замыкание R совпадает с отношением $\xrightarrow{R}$. Покажем вначале, что $\tilde{R} \subseteq \xrightarrow{R}$. Пусть $(A,B) \in \tilde{R}$. Если при этом $A \supseteq B$, то по определению $A \xrightarrow{R} B$. Остается случай, когда $A = \bigcup_i A_i,\ B = \bigcup_i B_i$, где каждая из пар (A_i,B_i) содержится в R либо рефлексивна и поэтому принадлежит $\xrightarrow{R}$. Поскольку $\xrightarrow{R}$ содержит отношение включения, то пары $(A,A_1),(A,A_2),\dots,(A,A_n)$ принадлежат $\xrightarrow{R}$. Отсюда, в силу транзитивности $\xrightarrow{R}$, получаем, что и пары $(A,B_1),(A,B_2),\dots,(A,B_n)$ принадлежат $\xrightarrow{R}$. Наконец, из свойства дистрибутивности $\xrightarrow{R}$ заключаем, что $A \xrightarrow{R} B$. Таким образом, $\tilde{R} \subseteq \xrightarrow{R}$. Поскольку транзитивное замыкание $\tilde{R}^*$ – наименьшее транзитивное отношение, содержащее $\tilde{R}$, и $\xrightarrow{R}$ также транзитивно, то имеем включение $\tilde{R}^* \subseteq \xrightarrow{R}$.

Докажем теперь обратное включение. Предположим, что $A \xrightarrow{R} B$. Тогда по определению существует логическая цепочка $\vec{r}_{AB} = (B_1,\dots,B_m)$ соединяющая (A,B). Для произвольного k ($1 \le k \le m+1$) рассмотрим пару (B_{k-1},B_k) (для единообразия полагаем $B_0 = A,\ B_{m+1} = B$). По определению логической цепочки эта пара дистрибутивно связана отношением R. Этот факт означает (см. определение 2.3.3), что существует конечное множество пар $\{(A_i^k,B_i^k),\ i = 1,\dots,n\}$ таких, что $A_i^k = B_i^k$ либо $(A_i^k,B_i^k) \in R$, причем $B_{k-1} \supseteq A_i^k,\ 1 \le i \le n$ и $\bigcup_i B_i^k \supseteq B_k$. Обозначим $\tilde{A}_k = \bigcup_i A_i^k$ и $\tilde{B}_k = \bigcup_i B_i^k$. Тогда по построению отношения $\tilde{R}$ имеем $(\tilde{A}_k,\tilde{B}_k) \in \tilde{R}$. Так как $\tilde{R}$ содержит отношение включения $\supseteq$, то справедливо также $(B_{k-1}, \tilde{A}_k) \in \tilde{R}$ и $(\tilde{B}_k, B_k) \in \tilde{R}$. Таким образом, пары $(B_{k-1}, \tilde{A}_k), (\tilde{A}_k, \tilde{B}_k), (\tilde{B}_k, B_k)$ принадлежат $\tilde{R}$. Этими тремя парами заменим в цепочке $\vec{r}_{AB}$ пару (B_{k-1}, B_k). Таким образом поступим для всех $1 \le k \le m+1$. В результате получим из $\vec{r}_{AB}$ транзитивную цепочку в $\tilde{R}$, соединяющую (A, B). Следовательно, по лемме 2.2.3 имеем $(A, B) \in \tilde{R}^*$. □

Следствие 1. Для любой связи $A \xrightarrow{R} B$ элементы соответствующей логической цепочки $\vec{r}_{AB} = (B_1, \ldots, B_m)$ могут быть подобраны специальным образом: каждая пара $(B_{k-1}, B_k) \in \tilde{R}$, причем справедливо $B_{k-1} \supset B_k$ либо $B_{k-1} = \bigcup A_i^k$ и $B_k = \bigcup B_i^k$, где $A_i^k = B_i^k$ или $(A_i^k, B_i^k) \in R$ (здесь $1 \le k \le m+1$, $B_0 = A$, $B_{m+1} = B$). Данный факт был попутно установлен при доказательстве второй части теоремы 1. □

Следствие 2. Если $R_1 \subseteq R_2$, то $\xrightarrow{R_1} \subseteq \xrightarrow{R_2}$.

Доказательство. Во-первых, из описания процесса построения $\tilde{R}$ легко заметить, что если $R_1 \subseteq R_2$, то $\tilde{R}_1 \subseteq \tilde{R}_2$. Далее, из свойств транзитивного замыкания (замечание 2.2.1) имеем $\tilde{R}_1^* \subseteq \tilde{R}_2^*$, что в силу теоремы 1 и означает доказываемый факт. □

Переходим непосредственно к вопросу о существовании и построении логической редукции отношений рассматриваемого класса.

Лемма 1. Пусть R – бинарное отношение на решетке $\mathbb{F}$. Для того, чтобы R являлось логической редукцией, необходимо и достаточно, чтобы R не содержало ни одной такой пары (A, B), для которой выполнено соотношение $A \xrightarrow{R \setminus \{(A,B)\}} B$.

Доказательство. Пусть отношение R представляет собой логическую редукцию, то есть является минимальным логически эквивалентным себе отношением. Если бы при этом в R существовала пара $A \xrightarrow{R \setminus \{(A,B)\}} B$, то в силу леммы 2.4.1 ее можно было бы исключить, получив при этом меньшее эквивалентное отношение, что невозможно по условию леммы.

Для доказательства обратного утверждения предположим, что не существует ни одной пары $(A, B) \in R$, для которой справедливо $A \xrightarrow{R \setminus \{(A,B)\}} B$. Необходимо доказать, что в этом случае R представляет собой логическую редукцию. Предположим противное – пусть существует отношение $R_0 \subset R$, эквивалентное R, и $(A, B) \in R \setminus R_0$. Тогда в силу эквивалентности $R \sim R_0$ справедливо $A \xrightarrow{R_0} B$. Так как отношение R_0 не содержит пару (A, B), то $R_0 \subseteq R \setminus \{(A, B)\}$, и логическая связь $A \xrightarrow{R_0} B$ противоречит сделанному предположению о том, что таких пар (A, B) в R нет. Полученное противоречие доказывает искомое утверждение. □

Для произвольного отношения R рассмотрим отношение $\underset{\sim}{R}$, построенное по данному R последовательным выполнением шагов, обратных построению $\tilde{R}$, а именно:

• исключить из R содержащиеся в нем пары вида $A \supset B$ и обозначить новое отношение R_{-1};

• исключить из R_{-1} всевозможные пары (A,B) с элементами вида $A = \bigcup\limits_i A_i$, $B = \bigcup\limits_i B_i$, где все (A_i, B_i) ($i = 1,\ldots,n$) принадлежат R_{-1} и не совпадают с (A,B);

• исключить из полученного отношения все рефлексивные пары.

Замечание 2. Из леммы 2.4.1 следует, что отношение $\underset{\sim}{R}$ логически эквивалентно R.

Заметим, что подобный же подход к построению *транзитивной редукции* («удалить все транзитивные пары») был бы ошибочным. Причина в том, что в некоторых ситуациях (наличие в отношении циклов) транзитивная редукция не единственна [2], и одновременное удаление всех имеющихся транзитивных пар может привести к потере связей. Однако, поскольку решетка ациклична, удаление всех указанных выше «объединенных» пар (A,B) приводит к одному и тому же результату независимо от порядка удаления. По этой причине можно говорить об одновременном удалении всех таких пар.

Следующая теорема указывает достаточное условие существования и способ построения логической редукции данного отношения.

Теорема 2. Пусть для бинарного отношения R, заданного на решетке $\mathbb{F}$, построено соответствующее отношение $\tilde{R}$ (см. выше). Тогда, если для $\tilde{R}$ существует транзитивная редукция R^0, то соответствующее ей отношение $\underset{\sim}{R}^0$ представляет собой логическую редукцию исходного отношения R.

Доказательство. Из замечаний 1–2 следует, что указанное в теореме отношение $\underset{\sim}{R}^0$ логически эквивалентно R. Осталось показать, что $\underset{\sim}{R}^0$ является логической редукцией. Для этого достаточно проверить выполнение условия леммы 1 для $\underset{\sim}{R}^0$.

Предположим противное, а именно – что для некоторой пары $(A,B) \in \underset{\sim}{R}^0$ имеет место $A \xrightarrow{\underset{\sim}{R}^0 \setminus \{(A,B)\}} B$. Обозначим $R_0 = \underset{\sim}{R}^0 \setminus \{(A,B)\}$ и рассмотрим соответствующую этой связи логическую цепочку $\vec{r}_{AB} = (B_1,\ldots,B_m) \in R_0$. Согласно следствию 1, эта цепочка может быть составлена таким образом, что каждая пара $(B_{k-1}, B_k) \in \tilde{R}_0$, причем справедливо $B_{k-1} \supset B_k$ либо $B_{k-1} = \bigcup\limits_i A_i^k$

и $B_k = \bigcup_i B_i^k$, где $A_i^k = B_i^k$ или $(A_i^k, B_i^k) \in R_0$ $(1 \le k \le m+1$, $B_0 = A$, $B_{m+1} = B)$. В рассматриваемом случае, поскольку отношение $\underset{\sim}{R}^0$ вложено в транзитивную редукцию отношения $\tilde{R}$, оно не может содержать транзитивной в $\tilde{R}_0 \subseteq \tilde{R}$ пары (A,B). Этот факт означает, что в рассматриваемой логической цепочке $m = 0$ и для пары (A,B) остаются варианты $A \supset B$ либо $A = \bigcup_i A_i^1$ и $B = \bigcup_i B_i^1$, где $A_i^1 = B_i^1$ или $(A_i^1, B_i^1) \in R_0$. Поскольку процесс построения отношения $\underset{\sim}{R}^0$ исключает также вложенные ($A \supset B$) и объединенные ($A = \bigcup_i A_i^1$ и $B = \bigcup_i B_i^1$) пары, то остается лишь вариант «унарного» объединения, когда $A = B$ либо $(A,B) \in R_0$. Наконец, замечаем, что рефлексивных пар ($A = B$) в $\underset{\sim}{R}^0$ также нет по построению, а пары (A,B) нет в $R_0 = \underset{\sim}{R}^0 \setminus \{(A,B)\}$.

Таким образом, приходим к заключению, что вывод $A \xrightarrow{\underset{\sim}{R}^0 \setminus \{(A,B)\}} B$ невозможен, и, как следствие, отношение $\underset{\sim}{R}^0$ является логической редукцией R. □

Заметим, что требование существования транзитивной редукции отношения $\tilde{R}$ при определенных условиях может оказаться избыточным для существования логической редукции исходного отношения R. Очевидно, что при конечном множестве R из него всегда можно последовательно удалить «лишние» пары, получив в результате логическую редукцию. Однако, если при этом сама решетка $\mathbb{F}$ бесконечна и имеет соответствующую структуру, то, объединяя R с отношением $\supseteq$, можно построить не имеющее транзитивной редукции отношение $\tilde{R}$. Таким образом, возникает вопрос об усилении теоремы 2. Одно из возможных направлений такого усиления – использование в представленных построениях вместо $\supseteq$ отношения вида $\supseteq_R$, содержащего лишь необходимое для получения логической редукции R подмножество $\supseteq$. В этом случае утверждения и доказательства настоящей главы будут более громоздкими, однако принципиальных новшеств не принесут. По этой причине оставим данные идеи лишь в качестве рекомендации для конкретных приложений.

Отметим также, что на практике при построении $\tilde{R}$ не обязательно физически добавлять указанные пары. Достаточно реализовать эффективный механизм проверки того, относится ли заданная пара к теоретически добавляемым.

Что касается оценок сложности алгоритмов построения логической редукции, то здесь прогнозы не всегда оптимистичны. Как показано в [39], задача минимизации функций Хорна является NP-полной. В настоящей работе описывается процедура нахождения не наименьшего, а минимального множества правил. Использование при этом быстрого алгоритма построения транзитивной редукции [2] дает сложность $O(N^3)$, если при этом удастся эффективно реализовать построение отношений $\tilde{R}$ и $\underset{\sim}{R}$. Заметим также, что число элементов решетки N может оказаться сопоставимым с мощностью булеана 2^n, где n – количество точек решетки.

2.6. Логические уравнения на решетках

В данном разделе вводится и изучается связанный с LP-структурами специальный класс уравнений. Рассматриваются вопросы о разрешимости и количестве решений этих уравнений, а также методы их решения. Нахождение решения продукционно-логического уравнения эквивалентно обратному логическому выводу на решетке. Этот процесс решения обобщает рассмотренные в п.2.1.3 методы нахождения минимальных порождающих множеств в продукционных системах.

Пусть дано некоторое отношение R на решетке $\mathbb{F}$ и имеет место $A \xrightarrow{R} B$. Тогда B называется образом A, а A – прообразом B при отношении $\xrightarrow{R}$. Однако каждый элемент из $\mathbb{F}$ может иметь много образов и прообразов. Более того, в данном случае в силу определения логического отношения любой $B_1 \subset B$ является образом A и каждый $A_1 \supset A$ является прообразом B. По этой причине при изучении образов и прообразов логических отношений необходимо уточнение рассматриваемых понятий.

Для данного $B \in \mathbb{F}$ минимальным прообразом при отношении $\xrightarrow{R}$ называется такой элемент $A \in \mathbb{F}$, что $A \xrightarrow{R} B$ и A является минимальным, то есть не содержит никакого другого $A_1 \in \mathbb{F}$, для которого $A_1 \xrightarrow{R} B$.

В оставшейся части этого раздела рассматриваются лишь точечные решетки.

Определение 1. Точка x решетки $\mathbb{F}$ называется начальной при отношении R, если в R нет ни одной такой пары (A,B), что x содержится в B и не содержится в A. Элемент X точечной решетки $\mathbb{F}$ называется начальным, если все его точки являются начальными (при отношении R). Подмножество $\mathbb{F}_0(R)$ (будем обозначать $\mathbb{F}_0$, если это не вызовет неоднозначностей) точечной решетки $\mathbb{F}$, состоящее из всех начальных элементов $\mathbb{F}$, называется начальным множеством решетки $\mathbb{F}$ (при отношении R).

Очевидно, начальное множество $\mathbb{F}_0$ образует подрешетку в $\mathbb{F}$.

Рассмотрим уравнение

$$R(X) = B\,, \tag{1}$$

где $B \in \mathbb{F}$ – заданный элемент, $X \in \mathbb{F}$ – неизвестный.

Определение 2. Частным решением X уравнения (1) называется любой минимальный прообраз элемента B, содержащийся в $\mathbb{F}_0$. Приближенным (частным) решением X уравнения (1) называется любой прообраз элемента B, содержащийся в $\mathbb{F}_0$. Общим решением уравнения (1) называется совокупность всех его частных решений $\{X_s\}$, $s \in S$.

По определению точное решение является и приближенным. Очевидно, приближенное решение всегда содержит хотя бы одно точное решение.

Уравнения вида (1) будем называть продукционно-логическими (в рамках данной главы – просто логическими) уравнениями на решетках.

Вначале рассмотрим вопрос о том, как меняется общее решение уравнений вида (1) при объединении их правых частей. Точнее, нельзя ли для одного и того же отношения R вместо исходного уравнения решать несколько уравнений с правыми частями, объединение которых равно данному B?

Лемма 1. Пусть X_1 – частное решение уравнения вида (1) с правой частью B_1, а Y_1 – частное решение уравнения того же вида с правой частью B_2. Тогда $X_1 \cup Y_1$ является приближенным решением уравнения

$$R(X) = B_1 \cup B_2\,. \tag{2}$$

Доказательство. Действительно, поскольку $\xrightarrow{R}$ содержит отношение включения, то $X_1 \cup Y_1 \xrightarrow{R} X_1$ и $X_1 \cup Y_1 \xrightarrow{R} Y_1$. Отсюда, так как по условию леммы $X_1 \xrightarrow{R} B_1$ и $Y_1 \xrightarrow{R} B_2$, в силу транзитивности $\xrightarrow{R}$ имеем $X_1 \cup Y_1 \xrightarrow{R} B_1$ и $X_1 \cup Y_1 \xrightarrow{R} B_2$. Наконец, из последних двух соотношений, пользуясь дистрибутивностью $\xrightarrow{R}$, получим $X_1 \cup Y_1 \xrightarrow{R} B_1 \cup B_2$. □

Теорема 1. Пусть $\{X_s\}$, $s \in S_1$ – общее решение уравнения вида (1) с правой частью B_1, а $\{Y_p\}$, $p \in S_2$ – общее решение уравнения того же вида с правой частью B_2. Тогда общее решение уравнения (2) представляет собой множество всех элементов вида $X_s \cup Y_p$, из которого исключены элементы, содержащие другие элементы этого же множества.

Доказательство. По лемме 1, каждый элемент $X_t \cup Y_p$ является прообразом $B_1 \cup B_2$, то есть содержит хотя бы одно частное решение (2). Покажем теперь, что уравнение (2) не имеет частных решений, отличных от вида $X_t \bigcup Y_p$. Предположим противное – пусть некоторый $Z \in \mathbb{F}_0$, являясь частным решением (2), не совпадает ни с одним элементом вида $X_t \bigcup Y_p$. При этом Z не может содержать ни одного элемента вида $X_t \bigcup Y_p$, иначе он не был бы частным решением (оно по определению минимально). Отсюда следует, что Z не содержит ни одного X_t (или ни одного Y_p, что равнозначно). С другой стороны, поскольку $Z \xrightarrow{R} B_1 \bigcup B_2$, то $Z \xrightarrow{R} B_1$ и $Z \xrightarrow{R} B_2$. Этот факт означает, что Z содержит хотя бы по одному частному решению уравнений с правыми частями B_1 и B_2, а все эти решения содержатся соответственно в множествах $\{X_s\}$ и $\{Y_p\}$. Таким образом, приходим к противоречию. □

Рассмотрим методы решения логических уравнений вида (1).

Далее в этом разделе будем предполагать, что R является конечным каноническим отношением на точечной решетке $\mathbb{F}$ (см. п. 2.4), не содержащим пар отношения $\supseteq$, а правая часть B уравнения (1) представляет собой конечное объединение точек.

Введем понятие структурного расслоения исходного отношения R на виртуальные слои (частичные отношения) $\{R_t \mid t \in T\}$. Оно позволит упростить изучение свойств отношения $\xrightarrow{R}$, а также облегчит построение и исследование ряда алгоритмов, связанных с решением соответствующих логических уравнений. С этой целью вначале разобьем R на непересекающиеся подмножества, каждое из которых образовано парами вида (A, x_p) с одним и тем же точечным элементом x_p в качестве правой части. Такое разбиение имеет смысл благодаря тому, что R является каноническим. Обозначим эти подмножества R^p соответственно их элементу $x_p, p \in P$.

Определение 3. Слоем R_t в отношении R называется подмножество R, образованное упорядоченными парами, взятыми по одной из каждого непустого R^p, $p \in P$. Два слоя, отличающиеся хотя бы одной парой, считаются различными.

Замечание 1. Каждый слой содержит максимально возможное подмножество пар из R с уникальными правыми частями. Добавление к слою еще одной пары нарушило бы это условие.

Замечание 2. Любое подмножество пар в R с уникальными правыми частями принадлежит некоторому слою.

Замечание 3. В общем случае слои имеют непустые пересечения. Объединение всех слоев равно R.

Замечание 4. Нетрудно заметить, что общее количество слоев N определяется равенством $N = \prod_{p \in P} N_p$, где N_p – мощность подмножества пар отношения R с правой частью x_p.

Будем констатировать, что логическая цепочка $\vec{r}_{AB}$ принадлежит в R некоторому слою R_t, если все нерефлексивные пары, определяющие логическую связь $A \xrightarrow{R} B$ (нерефлексивные элементы соответствующей матрицы вывода (2.3.2)), содержатся в одном и том же слое R_t.

Лемма 2. Если $A \xrightarrow{R} B$, то существует логическая цепочка $\vec{r}_{AB}$, принадлежащая в R некоторому слою R_t.

Утверждение этой леммы непосредственно вытекает из следствия 2.3.2 и замечания 2. □

Следствие 1. Логическое замыкание канонического отношения R на точечной решетке $\mathbb{F}$ равно объединению логических замыканий его слоев, то есть $\xrightarrow{R} = \bigcup_{t \in T} \xrightarrow{R_t}$.

Доказательство. Поскольку $R_t \subseteq R$, то согласно следствию 2.5.2 имеем $\xrightarrow{R_t} \subseteq \xrightarrow{R} (\forall t \in T)$. Обратно, если $A \xrightarrow{R} B$, то по лемме 2 справедливо $A \xrightarrow{R_t} B$ при некотором $t \in T$. Следовательно, $(A, B) \in \bigcup_{t \in T} \xrightarrow{R_t}$. □

Будем считать, что решение X уравнения (1) (точное или приближенное) порождается в R некоторым слоем R_t, если $X \xrightarrow{R_t} B$.

Замечание 5. Согласно лемме 2 и следствию 1, любое частное решение уравнения (1) порождается в R некоторым слоем R_t.

Замечание 6. Из следствия 1 вытекает, что для нахождения решения уравнения (1) в некотором слое R_t достаточно вместо (1) решить аналогичное уравнение с отношением R_t.

Последние замечания не гарантируют, что два различных слоя не могут порождать одного и того же решения. Кроме того, могут существовать слои, дающие частное решение для R_t, но приближенное для R. Некоторые слои могут вообще не давать решений. Однако, справедливо утверждение о том, что один слой не может порождать более одного точного решения. Чтобы установить этот факт, вначале докажем вспомогательные утверждения.

Лемма 3. Если X – частное решение уравнения (1), то существует матрица вывода M_{XB}, соответствующая паре $X \xrightarrow{R} B$, следующего вида

$$M_{XB} = \begin{pmatrix} (A_{1,1}, b_{1,1}) & (A_{1,2}, b_{1,2}) & \dots & (A_{1,m+1}, b_{1,m+1}) \\ (A_{2,1}, b_{2,1}) & (A_{2,2}, b_{2,2}) & \dots & (A_{2,m+1}, b_{2,m+1}) \\ \dots & \dots & \dots & \dots \\ (A_{p_1,1}, b_{p_1,1}) & (A_{p_2,2}, b_{p_2,2}) & \dots & (A_{p_{m+1},m+1}, b_{p_{m+1},m+1}) \\ (C_1, C_1) & (C_2, C_2) & \dots & (C_{m+1}, C_{m+1}) \end{pmatrix}, \qquad (3)$$

где $(A_{i,k}, b_{i,k}) \in R$, $b_{i,k}$ – попарно различные точки $\mathbb{F}$ $(1 \le i \le p_k;\ 1 \le k \le m+1)$; $C_k \in \mathbb{F}$ $(1 \le k \le m+1)$;

$$X = \bigcup_{1 \le i \le p_1} A_{i,1} \cup C_1\ ; \qquad (4)$$

$$B = \bigcup_{1 \le i \le p_{m+1}} b_{i,m+1} \cup C_{m+1}\ ; \qquad (5)$$

$$\bigcup_{1 \le i \le p_k} b_{i,k} \cup C_k = \bigcup_{1 \le i \le p_{k+1}} A_{i,k+1} \cup C_{k+1} \quad (1 \le k \le m); \qquad (6)$$

Доказательство. Покажем, каким образом может быть получена матрица (3).

Вначале, согласно замечанию 5, построим для пары (X, B) обычную матрицу вывода вида (2.3.2), образованную парами некоторого слоя R_t (с уникальными правыми частями) и рефлексивными парами элементов $\mathbb{F}$. Затем в каждом столбце заменим все рефлексивные пары единственной рефлексивной парой, содержащей объединение всех рефлексивных пар столбца (объединяются отдельно левые и правые части). Поскольку в данном случае имеем дело с точечной решеткой $\mathbb{F}$, то полученная матрица уже имеет вид (3). Для доказательства леммы остается лишь добиться выполнения условий (4)–(6), которые представляют собой усиленный вариант соотношений (2.3.3)–(2.3.4).

Заметим, что условие (4) выполняется автоматически в силу минимальности X, декларированной в определении 5 (см. также соотношение вида (2.3.3) при $k = 1$). Для доказательства соотношений (5)–(6) будем последовательно просматривать столбцы матрицы справа налево. Вначале положим $k = m + 1$, $B_k = B$.

Исключим из k-го столбца элементы вида (A, b) такие, что $b \notin B_k$. Исключим такие же элементы b и из обеих частей рефлексивной пары этого столбца. Очевидно, что условия вида (2.3.3)–(2.3.4) при этом не нарушатся, однако будет выполняться и еще одно из условий (5)–(6) (при данном значении k). Далее, переходя к очередному столбцу матрицы (справа налево), уменьшим k на единицу и положим $B_k = \bigcup_i A_{i,k+1} \cup C_{k+1}$.

Описанную в последнем абзаце процедуру проведем для всех столбцов матрицы. Таким образом получим искомую матрицу M_{XB}. □

Замечание 7. Согласно замечаниям 5–6, утверждение леммы остается верным, если в ее условии отношение R заменить на R_t.

Определение 4. Матрицу вида (3), имеющую нерефлексивные элементы с уникальными правыми частями и удовлетворяющую условиям (4)–(6), будем называть канонической матрицей вывода для пары $X \xrightarrow{R_t} B$.

Лемма 4. Если для некоторых $X_1, X_2 \in \mathbb{F}_0$ и $B \in \mathbb{F}$ построены канонические матрицы M_{X_1B} и M_{X_2B}, все нерефлексивные элементы которых содержатся в одном и том же слое R_t, то $X_1 = X_2$.

Доказательство. Предположим противное, что $X_1 \neq X_2$. Пусть, для определенности, существует точка $a_0 \in X_1$ и $a_0 \notin X_2$. Рассмотрим вначале матрицу M_{X_1B}. Будем просматривать ее столбцы слева направо и при этом строить некоторую специальную последовательность нерефлексивных пар.

Согласно условию (4), точка a_0, как и любая точка элемента X_1, содержится в левой части хотя бы одной пары первого столбца матрицы. Если существует такая нерефлексивная пара (A_1, a_1) ($a_0 \in A_1$), то добавим эту пару к формируемой последовательности и двигаясь далее будем искать аналогичную пару для a_1. Если же это (C_1, C_1), то по условию (6) a_0 аналогично перейдет в следующий столбец. Пройдя все столбцы матрицы, получим конечную последовательность нерефлексивных пар отношения R (возможно, пустую) $\{(A_i, a_i) \mid i = 1, \dots, n\}$, причем $a_i \in A_{i+1}$, $a_i \notin \mathbb{F}_0$ $(0 \le i \le n)$, $A_{n+1} = B$. Если эта последовательность окажется пустой ($n = 0$), то будем иметь $a_0 \in B$.

Рассмотрим теперь матрицу M_{X_2B}. Ее столбцы будем просматривать справа налево, пытаясь в этой матрице выделить построенную выше последовательность $\{(A_i, a_i)\}$. Поскольку $a_n \in B$, то в силу (5) точка a_n содержится в правой части по крайней мере одной пары последнего столбца матрицы M_{X_2B}. Если существует такая нерефлексивная пара, то это может быть лишь (A_n, a_n), так как слой R_t не может содержать двух пар с общей правой частью a_n. В этом случае переходим в предыдущий столбец матрицы с новой отправной точкой $a_{n-1} \in A_n$ – правой частью предыдущего элемента последовательности. Если же это (C_{m+1}, C_{m+1}), то по условию (6) в предыдущий столбец аналогично переходит a_n. Таким образом будем продвигаться справа налево в матрице и в последовательности $\{A_i, a_i\}$.

Этот процесс может завершиться в одном из трех случаев. Если столбцы матрицы закончатся одновременно с последовательностью, то получим $a_0 \in X_2$, что противоречит первоначальному предположению (см. начало доказательства). Если же столбцы матрицы закончатся раньше, чем исчерпается последовательность, то в этом случае соотношение $a_i \in X_2$, $i > 0$ будет противоречить тому, что $a_i \notin \mathbb{F}_0$, так как по условию X_2 целиком состоит из начальных точек. Наконец, в последнем варианте, когда последовательность закончится раньше столбцов матрицы, в матрице $M_{X_2 B}$ окажется пара вида (A_0, a_0). Этот факт противоречит тому, что $a_0 \in \mathbb{F}_0$. Итак, в каждом возможном случае приходим к противоречию, что и доказывает лемму. □

Следствие 2. Один слой R_t не может содержать двух различных частных решений уравнения (1).

Доказательство. Предположим, что существуют два решения. Тогда по лемме 3 и замечанию 7 для этих решений существуют канонические матрицы вывода, нерефлексивные элементы которых принадлежат R_t. Отсюда по лемме 4 получаем, что эти решения совпадают. □

Объединяя полученные результаты, можем сформулировать следующее утверждение.

Теорема 2. Для нахождения общего решения уравнения (1) достаточно найти частное решение X_t в каждом слое R_t, если оно существует. Далее из полученного множества решений необходимо исключить элементы, содержащие другие элементы этого же множества.

Утверждение теоремы сразу следует из замечания 5 и леммы 4. □

Следствие 3. Количество частных решений уравнения (1) оценивается сверху выражением $N = \prod_{p \in P} N_p$, где N_p – мощность подмножества пар отношения R с правой частью x_p, а P определяет все такие подмножества (см. замечание 4). □

Теорема 2 и замечание 6 позволяют свести вопрос о решении уравнения (1) к задаче нахождения частного решения уравнения

$$R_t(X) = B\ , \tag{7}$$

где B – неначальный элемент решетки $\mathbb{F}$, R_t – произвольный слой в R.

Согласно следствию 2, обратное к $\xrightarrow{R_t}$ отношение является однозначным отображением $f_t : \mathbb{F} \to \mathbb{F}$ с некоторой областью определения $D(f_t) \subseteq \mathbb{F}$. Рассмотрим некоторые важные его свойства.

Лемма 5. Отображение f_t является $\bigcup$-гомоморфизмом [93], то есть $f_t(B_1 \bigcup B_2) = f_t(B_1) \bigcup f_t(B_2)$, $\forall B_1, B_2 \in D(f_t)$.

Доказательство. Пусть $M_{X_1B_1}$ и $M_{X_2B_2}$ – канонические матрицы вывода соответственно для $X_1 \xrightarrow{R_t} B_1$ и $X_2 \xrightarrow{R_t} B_2$. Предположим для определенности, что количество столбцов $M_{X_1B_1}$ больше либо равно количеству столбцов $M_{X_2B_2}$ (это можно сделать, так как матрицы равноправны). Объединим эти матрицы следующим образом. К элементам каждого столбца $M_{X_1B_1}$ допишем элементы соответствующего по номеру столбца $M_{X_2B_2}$. В каждом оставшемся столбце $M_{X_1B_1}$ (их по нашему предположению могло быть больше) к рефлексивному элементу вида (C_k, C_k) добавим рефлексивные пары, образованные правыми частями элементов последнего столбца матрицы $M_{X_2B_2}$. Затем в новой матрице исключим повторяющиеся нерефлексивные пары, как это описано в лемме 4. В результате получим матрицу $M_{X_{12}B_{12}}$, которая очевидным образом является канонической матрицей вывода для пары $(X_1 \bigcup X_2, B_1 \bigcup B_2)$. □

Следствие 4. Отображение f_t является изотонным [93], Это означает, что если $B_1 \supseteq B_2$, то $f_t(B_1) \supseteq f_t(B_2)$. □

Таким образом, основываясь на лемме 5, для решения уравнения (7) достаточно решить уравнение с каждой точкой элемента B в качестве правой части. Принимая во внимание это обстоятельство, рассмотрим задачу нахождения частного решения следующего уравнения:

$$R_t(X) = b, \tag{8}$$

где b – неначальная точка решетки $\mathbb{F}$, R_t – произвольный слой в R.

Рассмотрим методы решения этой задачи. Покажем, что она эквивалентна задаче на ориентированном графе перечисления всех входных вершин, из которых достижима данная вершина.

Вначале построим такой граф G_{R_t}. Каждой точке решетки $\mathbb{F}$, участвующей в отношении R, сопоставим вершину графа. Далее для каждой пары (A, a) рассматриваемого слоя R_t построим дуги, ведущие из всех вершин, соответствующих точкам A, в вершину, соответствующую данной a. Иногда для краткости будем отождествлять точки $\mathbb{F}$ и соответствующие им вершины в графе.

В полученном графе G_{R_t} выберем вершину b. Рассмотрим подграф $G_{R_t,b} \subseteq G_{R_t}$, содержащий все вершины, из которых достижима вершина b (включая саму b) и все дуги, соединяющие такие вершины.

Лемма 6. Если подграф $G_{R_t,b}$ не содержит ориентированных циклов, то множество всех его входных вершин (не имеющих входящих дуг) соответствует частному решению уравнения (8).

Доказательство. Предположим вначале, что граф $G_{R_t,b}$ ациклический. Покажем, как по нему построить соответствующую каноническую матрицу вывода M_{Xb}. Проведем построение, используя индукцию по столбцам, справа налево.

Последний столбец матрицы сформируем из единственного элемента – $(A_{m+1}, b) \in R_t$, где элемент A_{m+1} состоит из всех точек, непосредственно связанных дугами с вершиной b. Очевидно, условие (5) выполнено. Предположим теперь, что построен очередной столбец с элементами $(A_{i,k}, b_{i,k}) \in R_t$, $b_{i,k}$ – попарно различные точки $\mathbb{F}$ $(1 \le i \le p_k)$ и (C_k, C_k), $C_k \in \mathbb{F}$ $(1 < k \le m+1)$, причем выполнены условия (5)-(6). Тогда для каждой точки a каждого элемента $A_{i,k}$ найдем все вершины графа, дуги из которых входят в a, сформируем из них (в виде объединения) элемент $A \in \mathbb{F}$ и поместим пару $(A, a) \in R_t$ в следующий столбец (с номером $k-1$). Если же точка $a \in A_{i,k}$ окажется начальной (соответствует входной вершине графа), то добавим ее к рефлексивному элементу старого столбца, то есть в новый столбец вместо элемента (C_k, C_k) войдет элемент вида $(C_k \bigcup a, C_k \bigcup a)$. По построению рассматриваемого столбца условие (6) окажется выполненным. Так как граф не содержит циклов, то через конечное число шагов окажется, что все точки левых частей нерефлексивных элементов очередного столбца окажутся начальными. На этом процесс построения матрицы M_{Xb} заканчивается.

Поскольку таким способом будут пройдены все дуги графа $G_{R_t,b}$, то можно утверждать, что объединение X левых частей элементов первого столбца матрицы состоит из *всех* начальных точек, соответствующих входным вершинам графа $G_{R_t,b}$. С другой стороны, X – решение уравнения (8). По лемме 4 оно является точным решением. □

Замечание 8. Описанный в доказательстве процесс построения матрицы M_{Xb} напоминает процедуру обхода всех вершин графа $G_{R_t,b}$. Однако он не полностью соответствует ей, поскольку некоторые точки (вершины) могут обрабатываться более одного раза. Данный процесс приведен для доказательства леммы и носит лишь теоретический характер. Для практического же применения можно использовать именно указанный обход из вершины b (против стрелок, «в ширину») вершин графа $G_{R_t,b}$.

Лемма 7. Если X – решение уравнения (8), то в любой его канонической матрице вывода M_{Xb} содержатся точки, соответствующие всем вершинам графа $G_{R_t,b}$.

Доказательство. Предположим противное, а именно – для некоторой вершины y_0 графа $G_{R_t,b}$ соответствующая точка y_0 не содержится ни в одной паре матрицы M_{Xb}. По построению графа существует путь $y_1,...,y_{n-1}$ из вершины y в вершину b (можно считать $b = y_n$). Подобно доказательству леммы 4, будем просматривать столбцы матрицы M_{Xb} справа налево, пытаясь выделить в ней отмеченную выше последовательность $\{y_i\}$. Положим $i = n$.

По условию (6) (при $i = n$ – (5)) y_i содержится в правой части пары очередного (при $i = n$ – последнего) столбца матрицы M_{Xb}. Если существует такая нерефлексивная пара (Y_i, y_i) (при $i = n$ это именно так), то в этом случае переходим в предыдущий столбец матрицы с новой отправной точкой $y_{i-1} \in Y_i$ (эта принадлежность верна, так как Y_i содержит все вершины, предшествующие y_i , в том числе и y_{i-1}). Если же это (C_{k_i}, C_{k_i}), то по условию (6) в предыдущий столбец аналогично переходит y_i . Продолжая этот процесс, будем продвигаться справа налево в матрице и в последовательности $\{y_i\}$.

Как и ранее, рассмотрим возможные варианты завершения процесса. Если последовательность закончится ранее столбцов матрицы или одновременно с ними, то получим, что y_0 содержится в матрице. Данный факт будет противоречить сделанному в начале доказательства предположению. Если же столбцы матрицы закончатся раньше, чем исчерпается последовательность, то тогда соотношение $y_i \in X$, $i > 0$ будет противоречить тому, что $y_i \notin \mathbb{F}_0$, так как по определению X целиком состоит из начальных точек. Таким образом, в обоих возможных случаях приходим к противоречию, что и доказывает лемму. □

Лемма 8. Если в графе $G_{R_t,b}$ есть ориентированный цикл, то уравнение (8) не имеет решений.

Доказательство. Предположим противное, что существует решение X уравнения (8), и тогда построим соответствующую ему каноническую матрицу вывода M_{Xb}. Пусть y – некоторая вершина графа $G_{R_t,b}$, содержащаяся в его цикле. По предыдущей лемме в матрице M_{Xb} существует элемент, в левой или правой части которого присутствует соответствующая точка y .

Предположим для определенности, что в столбце с номером k находится элемент матрицы, содержащий в своей правой части точку y (вариант нахождения y в левой части некоторой пары с помощью условия (6) сводится к случаю правой части). Если это (C_k, C_k), то согласно (6) получим, что и в столбце номер $k-1$ есть элемент с точкой y в правой части. Если же это нерефлексивная пара (Y, y), то рассмотрим вершину y_1, предшествующую y в цикле графа. Соответственно имеем $y_1 \in Y$, поскольку Y содержит все вершины графа, непосредственно предшествующие y. В силу свойства (6), в $k-1$ столбце матрицы M_{Xb} содержится элемент, содержащий в правой части y_1.

Таким образом, за один шаг переместимся в матрице на один столбец влево, а в графе – не более чем на одну вершину назад в цикле. Продолжая этот процесс, через конечное число шагов (учитывая еще и условие (4)) выйдем из первого столбца в матрице, т.е. очередная рассматриваемая точка y_i окажется в X, но в графе вершина y_i останется в том же цикле. Это означает противоречие, т.к. в X содержатся только начальные точки, а в цикле графа – лишь невходные вершины. □

Объединяя утверждения доказанных лемм 6 и 8, можно сформулировать следующий факт, завершающий обоснование пошагового процесса решения исходного логического уравнения (1).

Теорема 3. Уравнение (8) имеет не более одного решения. Если граф $G_{R_t,b}$ не содержит циклов, то единственное решение уравнения состоит из всех точек, соответствующих входным вершинам графа. Если $G_{R_t,b}$ содержит хотя бы один цикл, то уравнение решений не имеет. □

Итак, в настоящей главе изложены основные положения общей теории LP-структур. Результаты последующих глав основываются на аналогичных положениях, развивая их в том или ином специальном направлении. В главе 3 будут рассмотрены LP-структуры, определенные на полных решетках.

Глава 3
LP-СТРУКТУРЫ НА ПОЛНЫХ РЕШЕТКАХ

В данной главе вводятся LP-структуры, образованные на основе полных решеток. Полученные результаты представляют обобщение основной части материала главы 2 и могут быть применены для моделирования стандартных продукционных систем с бесконечными правилами. Некоторые определения по форме выглядят подобными соответствующим определениям, которые даны в главе 2. Однако теперь их содержание связано с полными решетками, и это обстоятельство усложняет как смысл данных понятий, так и доказательство их свойств. Как и в предыдущей главе, по аналогичным соображениям, для «решеточных» операций вместо знаков $\leq$, $\geq$, и $\vee$ используются обозначения теории множеств $\subseteq$, $\supseteq$, и $\bigcup$.

3.1. Определение LP-структуры на полной решетке

В настоящем разделе определяется понятие LP-структуры на полной решетке и исследуются ее свойства. Вводятся бинарные отношения, содержащие семантику продукционно-логического вывода со свойством нетеровости, то есть транзитивной завершаемости. Приводится обоснование необходимости введения соответствующих ограничений модели. Доказывается существование логического замыкания нетерового бинарного отношения на полной решетке, что позволяет определить понятие эквивалентного отношения.

Бинарное отношение R на полной решетке $\mathbb{F}$ называется вполне $\bigcup$-дистрибутивным (в рамках данной главы – просто дистрибутивным), если для любого подмножества элементов решетки $\{B_t \mid t \in T\}$, такого, что $(A, B_t) \in R, \forall t \in T$, справедливо $(A, \bigcup\limits_{t \in T} B_t) \in R$. Как и выше (глава 2), определяющее решетку отношение включения $\supseteq$ является дистрибутивным.

Заметим, что в рамках теории, изложенной в главе 2, продукционно-логическая связь элементов решетки $A \xrightarrow{R} B$ осуществлялась логической цепочкой $\vec{r}_{AB}$ конечной длины с матрицей вывода (2.3.2)[1] конечных размеров. В данном разделе свойство полной дистрибутивности (при дальнейших построениях, аналогичных представленным в главе 2) формально допускает бесконечные столбцы, и, более того, несчетные множества взамен столбцов матрицы. Однако и здесь планируется рассматривать лишь логические цепочки конечной длины. Причина в том, что они соответствуют реалиям практических задач и могут быть реализованы программными средствами. К сожалению, такой подход не может быть осуществлен только с использованием свойств дистрибутивности и транзитивности рассматриваемых отношений. Покажем это.

Предположим, что множество пар исходного бинарного отношения R на полной решетке $\mathbb{F}$ образует счетную ацикличную цепочку вида $(A,B_1),(B_1,B_2),\ldots,(B_{k-1},B_k),(B_k,B_{k+1}),\ldots$. Тогда, в силу транзитивности отношения $\xrightarrow{R}$, имеем $A\xrightarrow{R}B_1, A\xrightarrow{R}B_2,\ldots,$ $A\xrightarrow{R}B_k, A\xrightarrow{R}B_{k+1},\ldots$ Рассматривая совокупность всех таких пар, с помощью дистрибутивности $\xrightarrow{R}$ (определение 1) получим $A\xrightarrow{R}B$, где $B=\bigcup\limits_{1\le k<\infty}B_k$. С другой стороны, попытка построить для данной связи $A\xrightarrow{R}B$ матрицу вывода с необходимыми свойствами оказывается неудачной. Претендующая на это матрица

$$M_{AB}=\begin{pmatrix} (B_1,B_1) & (B_1,B_1) & (B_1,B_1) & \ldots \\ (B_1,B_2) & (B_2,B_2) & (B_2,B_2) & \ldots \\ & (B_2,B_3) & (B_3,B_3) & \ldots \\ & & (B_3,B_4) & \ldots \\ & & & \ldots \end{pmatrix} \tag{1}$$

имеет бесконечными не только столбцы, что естественно отражает свойство полной дистрибутивности, но и строки.

Таким образом, при определении продукционно-логического отношения на полной решетке появляется необходимость в формулировке дополнительного свойства, гарантирующего конечность длин транзитивных цепочек, реализующих логические связи элементов. В некоторых алгебраических системах подобное свойство (отсутствие неограниченных транзитивных последовательностей) называется нетеровостью [17]. Для того, чтобы сформулировать аналогичное свойство отношения в рассматриваемом случае, вначале определим понятие логической связи.

---

[1] Как обычно, это ссылка на формулу (2) из п. 2.3.

Пусть задано бинарное отношение R на полной решетке $\mathbb{F}$ и выбраны два элемента $A, B \in \mathbb{F}$. Пусть существует некоторое (конечное или бесконечное) множество пар $\{(A_p, B_p) \mid p \in P\}$, где для любого p справедливо $A_p = B_p$ либо $(A_p, B_p) \in R$. Если при этом $A \supseteq A_p, p \in P$ и $\bigcup_p B_p \supseteq B$, то будем считать, что упорядоченная пара элементов (A, B) (вполне) дистрибутивно связана отношением R.

Замечание 1. Пусть задано произвольное отношение R на полной решетке $\mathbb{F}$ и имеется множество пар $\{(A_t, B_t) \mid t \in T\}$, каждая из которых дистрибутивно связана отношением R на основе собственного множества $\{(A_p^t, B_p^t) \mid p \in P_t\}$. Тогда пара $(A, B) = (\bigcup_{t \in T} A_t, \bigcup_{t \in T} B_t)$ дистрибутивно связана отношением R на основе множества $\{(A_p^t, B_p^t) \mid p \in P_t,\ t \in T\}$. Этот факт следует из понятия дистрибутивной связи и элементарных свойств операций объединения на полной решетке [93].

Определение 1. Пусть R – произвольное отношение на полной решетке $\mathbb{F}$ и выбраны два элемента $A, B \in \mathbb{F}$. Пусть также существует упорядоченный конечный набор элементов $\vec{r}_{AB} = (B_1, \ldots, B_m)$ ($B_1, \ldots, B_m \in \mathbb{F}$) такой, что в последовательности $(B_0, B_1), (B_1, B_2), \ldots, (B_m, B_{m+1})$, где $B_0 = A,\ B_{m+1} = B$, каждая пара дистрибутивно связана R (если дистрибутивно связана сама пара (A, B), полагаем $m = 0$). Тогда указанный набор $\vec{r}_{AB}$ будем называть логической цепочкой (длины m), соединяющей A и B. Пару (A, B) при этом будем называть логически связанной отношением R и обозначать этот факт $A \xrightarrow{R} B$.

Совокупность пар элементов полной решетки $\mathbb{F}$, участвующих в определении логической цепочки $\vec{r}_{AB} = (B_1, \ldots, B_m)$, можно записать в виде специального вектора вывода:

$$\vec{V}_{AB} = (\{(A_{p_1,1}, B_{p_1,1})\}, \{(A_{p_2,2}, B_{p_2,2})\}, \ldots, \{(A_{p_{m+1},m+1}, B_{p_{m+1},m+1})\}), \qquad (2)$$

где каждый элемент с номером k $(1 \le k \le m+1)$ представляет собой множество пар $\{(A_{p_k,k}, B_{p_k,k}) \mid p_k \in P_k\}$ и осуществляет дистрибутивную связь пары (B_{k-1}, B_k) в цепочке $\vec{r}_{AB}$. Согласно определению 2, справедливы следующие соотношения:

$$B_{k-1} \supseteq A_{p_k,k}\ (p_k \in P_k) \ \text{ и } \bigcup_{p_k \in P_k} B_{p_k,k} \supseteq B_k \ \ (1 \le k \le m+1); \qquad (3)$$

$$(A_{p_k,k}, B_{p_k,k}) \in R \ \text{ либо } \ A_{p_k,k} = B_{p_k,k} \ \ (p_k \in P_k;\ 1 \le k \le m+1). \qquad (4)$$

Замечание 2. Не ограничивая общности, можно считать, что каждый элемент вектора вывода содержит по крайней мере одну неподчиненную пару (отличную от вида $A_{p_k,k} \supseteq B_{p_k,k}$). В противном случае соответствующее множество (элемент) можно удалить из вектора, не нарушая условий вида (3)–(4).

Замечание 3. Нетрудно заметить, что при заданном векторе $\vec{V}_{AB}$ вместо элементов B_{k-1} логической цепочки, соединяющей (A,B), можно выбрать $\breve{B}_{k-1} = \bigcup_{p_k \in P_k} A_{p_k,k}$ $(1 \le k \le m+1)$. Тогда при $k=1$ вместо A можно получить меньший исходный элемент $\breve{B}_0 \subseteq A$. Рассмотрим также ряд других свойств вектора (2), которые будут полезны в этой главе.

Лемма 1. Пусть дан некоторый вектор вывода $\vec{V}_{AB}$, соответствующий логической цепочке $\vec{r}_{AB} = (B_1,\dots,B_m)$ и удовлетворяющий условиям (3)–(4). Пусть при некоторых j, p_j справедливо $B_{p_j,j} \supseteq X$, $X \in \mathbb{F}$. Тогда вектор $\vec{V}'_{AB}$, полученный из $\vec{V}_{AB}$ добавлением к его $j+1$ элементу (как множеству пар) рефлексивной пары (X,X), определяет логическую цепочку $\vec{r}'_{AB} = (B_1,\dots,B_{j-1},B_j \cup X, B_{j+1} \cup X, B_{j+2},\dots,B_m)$, также осуществляющую связь $A \xrightarrow{R} B$.

Доказательство. Достаточно для нового вектора $\vec{V}'_{AB}$ показать справедливость условий (3)-(4) при тех значениях k, при которых имеются изменения в логической цепочке и ее векторе. Поскольку $\bigcup_{p_k \in P_k} B_{p_k,k} \supseteq B_k$ и при некотором j выполнено $B_{p_j,j} \supseteq X$, то $\bigcup_{p_j \in P_j} B_{p_j,j} \supseteq B_j \cup X$. Далее, так как $B_j \supseteq A_{p_{j+1},j+1}$, то $B_j \cup X \supseteq A_{p_j,j+1}$. Очевидно также, что $B_j \cup X \supseteq X$. Наконец, из $\bigcup_{p_{j+1} \in P_{j+1}} B_{p_{j+1},j+1} \supseteq B_{j+1}$ следует $\bigcup_{p_{j+1} \in P_{j+1}} B_{p_{j+1},j+1} \cup X \supseteq B_{j+1} \cup X$, а из $B_{j+1} \supseteq A_{p_{j+2},j+2}$ вытекает, что $B_{j+1} \cup X \supseteq A_{p_{j+2},j+2}$. Таким образом, условие (3) остается выполненным. Условие (4) также сохраняется, так как к элементам вектора добавлена рефлексивная пара. □

Следствие 1. По индукции нетрудно показать, что описанному в лемме 1 преобразованию можно подвергнуть любое количество элементов вектора подряд, начиная с $k+1$-го. При этом новая матрица также будет осуществлять логическую связь $A \xrightarrow{R} B$.

Лемма 2. Если справедливо $A \xrightarrow{R} B$, то эту связь можно реализовать подмножеством R, в котором все правые части используемых нерефлексивных пар уникальны.

Доказательство. Пусть дан вектор вида (2), удовлетворяющий для некоторой пары (A,B) условиям (3)–(4). Не ограничивая общности, можно считать, что каждый элемент вектора (множество пар) не содержит пар с одинаковыми правыми частями, в противном случае «лишние» можно исключить без ущерба для выполнения условий (3)–(4). Действительно, удаление пар с повторными правыми частями в одном и том же множестве не нарушает условия (3): $B_{k-1} \supseteq A_{p_k,k}$ для всех $p_k \in P_k$ означает, что B_{k-1} содержит и любое меньшее количество элементов $A_{p_k,k}$. С другой стороны, после удаления $B_{p_k,k}$ объединение $\bigcup_{p_k} B_{p_k,k}$ не изменится, так как $B_{p_k,k}$ встречается в другой паре этого же столбца. При удалении пары из любого элемента не может нарушиться и условие (4).

Преобразуем вектор $\vec{V}_{AB}$ следующим образом. Будем последовательно слева направо просматривать его элементы и искать в них такую нерефлексивную пару $(A_{p_k,k}, B_{p_k,k})$, что $B_{p_k,k}$ появляется также в качестве правой части пары в некотором предыдущем множестве с номером l. При успешном поиске к каждому множеству с $l+1$ по k добавим пару $(B_{p_k,k}, B_{p_k,k})$, после чего из множества k удалим пару $(A_{p_k,k}, B_{p_k,k})$. В результате получим новый вектор вида (2), который, согласно следствию 1, также будет удовлетворять условиям (3)-(4) для (A, B), однако в своих компонентах с номерами $\leq k$ будет содержать единственную пару с правой частью $B_{p_k,k}$. Если в результате удаления пары в некотором множестве не окажется ни одной неподчиненной пары (отличной от вида $A_{p_k,k} \supseteq B_{p_k,k}$, см. замечание 2), удалим этот элемент вектора. Просматривая таким образом весь вектор, получим, что $B_{p_k,k}$ будет встречаться в правых частях пар вектора не более одного раза.

Заметим, что аналогичные действия по удалению пар с любой другой правой частью никак не пересекутся с перечисленными выше действиями в отношении $B_{p_k,k}$. Этот факт означает корректность постановки вопроса об одновременном исключении всех подобных пар при любом имеющемся их количестве. □

Определение 2. Вектор вывода $\vec{V}_{AB}$ называется каноническим, если он удовлетворяет замечанию 2, а также содержит нерефлексивные пары лишь с уникальными правыми частями, причем правая часть любой пары элемента вектора с номером k не содержится в правых частях других пар в элементах вектора с номерами $1, \ldots, k$.

Лемма 3. Для любой имеющейся связи $A \xrightarrow{R} B$ можно построить канонический вектор вывода.

Доказательство. Утверждение леммы 3 следует из замечания 2 и леммы 2. Наличие последнего указанного в определении 4 свойства канонического вектора устанавливается аналогично доказательству леммы 2. □

Как уже отмечалось ранее, в ряде известных алгебраических систем формулируется свойство отсутствия неограниченных транзитивных последовательностей. В нашем случае аналог такого свойства дает следующее определение.

Определение 3. Бинарное отношение на полной решетке называется нетеровым, если оно не порождает канонического вектора вывода неограниченной длины.

Заметим, что условие нетеровости LP-структуры слабее аналогичного свойства в системах переписывания термов [17], так как в отличие от них допускает циклы.

Рассмотрим свойства операций над логическими цепочками.

Пусть существуют цепочки $\vec{r}_{AB}$ и $\vec{r}_{BC}$, логически соединяющие отношением R соответственно пары (A,B) и (B,C). Цепочка $\vec{r}_{AC} = \vec{r}_{AB}\vec{r}_{BC}$, составленная последовательно из $\vec{r}_{AB}, B, \vec{r}_{BC}$, называется транзитивным объединением цепочек $\vec{r}_{AB}$ и $\vec{r}_{BC}$.

Замечание 4. Очевидно, $\vec{r}_{AC} = \vec{r}_{AB}\vec{r}_{BC}$ является логической цепочкой, реализующей связь $A \xrightarrow{R} C$. Таким образом, если имеют место $A \xrightarrow{R} B$ и $B \xrightarrow{R} C$, то справедливо и $A \xrightarrow{R} C$.

Операция транзитивного объединения легко распространяется на конечное число операндов с сохранением аналогичных свойств.

Для того чтобы ввести операцию суммы цепочек, понадобится следующее утверждение.

Лемма 4. Пусть R – нетерово отношение на полной решетке $\mathbb{F}$ и имеется множество пар $\{(A_t, B_t) \mid t \in T\}$, каждая из которых логически связана отношением R. Тогда существует логическая связь $\bigcup_{t\in T} A_t \xrightarrow{R} \bigcup_{t\in T} B_t$.

Доказательство. Для каждой связи $A_t \xrightarrow{R} B_t$ соответствующую ей логическую цепочку обозначим $\vec{r}_{A_tB_t} = (B_1^t, \ldots, B_{m_t}^t)$. Основываясь на лемме 3, можно считать, что все цепочки $\vec{r}_{A_tB_t}$ реализуются каноническими векторами вывода. Рассмотрим цепочку $\vec{r} = (\bigcup_{t\in T} B_1^t, \bigcup_{t\in T} B_2^t, \ldots, \bigcup_{t\in T} B_m^t, \ldots)$, составленную из соответствующих по номерам компонент цепочек $\vec{r}_{A_tB_t}$. При этом цепочки $\vec{r}_{A_tB_t}$, оказавшиеся короче других, продолжим их последним элементом.

Покажем, что при нетеровом отношении R цепочка $\vec{r}$ конечна. С этой целью будем строить цепочку $\vec{r}$ последовательно слева направо, одновременно формируя ее канонический вектор вывода $\vec{V}$. Очередную компоненту $\vec{V}$ получим как объединение соответствующих компонент векторов вывода цепочек $\vec{r}_{A_tB_t}$. Легко проверить, что при этом условия (3)–(4) для $\vec{V}$ окажутся выполненными. После добавления к $\vec{V}$ каждой компоненты будем производить его «чистку» аналогично доказательству леммы 2 (см. также доказательство леммы 3) с тем, чтобы он оставался каноническим. В силу нетеровости отношения R, процесс увеличения длины вектора $\vec{V}$ рано или поздно завершится. Соответственно прекратится и процесс формирования цепочки $\vec{r}$. Такое событие может произойти в одном из следующих случаев: завершатся все цепочки $\vec{r}_{A_tB_t}$; в цепочке $\vec{r}$ перестанут появляться элементы решетки $B_k = \bigcup_{t\in T} B_k^t$, не содержащиеся в ее предыдущих элементах.

Таким образом, полученная конечная цепочка $\vec{r}$ реализует логическую связь $\bigcup_{t\in T} A_t \xrightarrow{R} \bigcup_{t\in T} B_t$. □

Пусть R – нетерово отношение на полной решетке $\mathbb{F}$ и дано такое множество пар $\{(A_t, B_t) \mid t \in T\}$, что каждая из них логически связана отношением R на основе цепочки $\vec{r}_{A_tB_t} = (B_1^t, \ldots, B_{m_t}^t)$. Суммой цепочек $\sum_{t\in T} \vec{r}_{A_tB_t}$ называется цепочка $\vec{r}_{AB} = (\bigcup_{t\in T} B_1^t, \bigcup_{t\in T} B_2^t, \ldots, \bigcup_{t\in T} B_M^t)$, конечная в силу леммы 4.

Замечание 5. Из леммы 4 следует, что $\sum_{t\in T} \vec{r}_{A_tB_t}$ является логической цепочкой, реализующей связь $\bigcup_{t\in T} A_t \xrightarrow{R} \bigcup_{t\in T} B_t$.

Пусть R – нетерово отношение на полной решетке $\mathbb{F}$ и даны логические цепочки $\vec{r}_{AB_t} = (B_1^t, \ldots, B_{m_t}^t),\ t \in T$, соответствующие связям $A \xrightarrow{R} B_t,\ t \in T$. Дистрибутивным объединением цепочек $\{\vec{r}_{AB_t}\}$ будем называть цепочку $\vec{r}_{A\sum B_t} = \sum_{t\in T} \vec{r}_{AB_t}$.

Замечание 6. Поскольку дистрибутивное объединение – это частный случай суммы, то $\vec{r}_{A\sum B_t}$ является логической цепочкой, реализующей логическую связь $A \xrightarrow{R} \bigcup_{t\in T} B_t$. Таким образом, из $A \xrightarrow{R} B_t,\ t \in T$ следует $A \xrightarrow{R} \bigcup_{t\in T} B_t$.

В дальнейшем потребуется также следующая лемма.

Лемма 5. Пусть R – нетерово отношение, заданное на полной решетке $\mathbb{F}$. Пусть также для некоторой пары элементов (A, B) существует множество логически связанных пар $\{(A_t, B_t) \mid t \in T\}$ такое, что $A \supseteq A_t,\ t \in T$ и $\bigcup_{t\in T} B_t \supseteq B$. Тогда имеет место связь $A \xrightarrow{R} B$.

Доказательство. Пусть $\vec{r}_t$ – логическая цепочка, связывающая пару (A_t, B_t) ($t \in T$). Обозначим $\tilde{A} = \bigcup_t A_t$ и $\tilde{B} = \bigcup_t B_t$. Тогда, согласно замечанию 5, цепочка $\vec{r} = \sum \vec{r}_t$ логически связывает пару $(\tilde{A}, \tilde{B})$ в R. Отсюда в силу вложений $A \supseteq \tilde{A}$ и $\tilde{B} \supseteq B$ следует, что пара (A, B) логически связана цепочкой, составленной последовательно из $\tilde{A}, \vec{r}, \tilde{B}$. □

Отношение на полной решетке называется продукционно-логическим (в рамках данной главы – просто логическим), если оно нетерово, а также содержит $\supseteq$, дистрибутивно и транзитивно. Логическим замыканием нетерового отношения R называется наименьшее логическое отношение, содержащее R.

Нетрудно заметить, что отношение включения $\supseteq$ на полной решетке является наименьшим логическим отношением. Отметим также, что все канонические векторы вывода для отношения $\supseteq$ имеют длину, равную 1.

Аналогично схеме действий, изложенной в главе 2, переходим к выяснению вопроса о структуре логических отношений на полной решетке.

Лемма 6. Если R – логическое отношение на полной решетке $\mathbb{F}$ и пара (A, B) дистрибутивно связана отношением R, то $(A, B) \in R$.

Доказательство. Поскольку логическое отношение R содержит $\supseteq$, то из определения 2 следует $(A, A_p) \in R$ при всех $p \in P$. Учитывая, что R рефлексивно, из того же определения имеем $(A_p, B_p) \in R$ при каждом $p \in P$. Отсюда из свойства транзитивности R вытекает $(A, B_p) \in R$. Наконец, пользуясь дистрибутивностью отношения R, имеем $(A, \bigcup_p B_p) \in R$ и, еще раз – транзитивностью R, получим $(A, B) \in R$. □

Лемма 7. Если R – логическое отношение на полной решетке $\mathbb{F}$ и пара (A, B) логически связана отношением R, то $(A, B) \in R$.

Доказательство. Это утверждение автоматически вытекает из определения 1, леммы 6 и транзитивности логического отношения R. □

Перейдем к доказательству существования логического замыкания.

Теорема 1. Для нетерового отношения R на полной решетке $\mathbb{F}$ логическое замыкание существует и представляет собой множество $\xrightarrow{R}$ всех упорядоченных пар $A, B \in \mathbb{F}$, логически связанных отношением R.

Доказательство. Очевидно, если $A \supseteq B$ либо $(A, B) \in R$, то пара (A, B) дистрибутивно связана отношением R (на основе множества пар $\{(A,B)\}$). Этот факт означает, что отношение $\xrightarrow{R}$ содержит $\supseteq$ и R. Из замечания 6 следует его дистрибутивность, а из замечания 4 – транзитивность. Таким образом, отношение $\xrightarrow{R}$ – логическое.

Покажем, что это наименьшее логическое отношение для R. Пусть R_1 – любое другое логическое отношение, содержащее R. Пусть также $A \xrightarrow{R} B$. Тогда из $R \subseteq R_1$ следует $A \xrightarrow{R_1} B$. Отсюда по лемме 7 получаем, что $(A,B) \in R_1$. Следовательно, $\xrightarrow{R} \subseteq R_1$. □

Таким образом, показано, что для нетерового бинарного отношения на полной решетке логическое замыкание существует. Этот факт позволяет ввести понятие эквивалентных нетеровых отношений, что лежит в основе формальных преобразований и оптимизации продукционных баз знаний с бесконечными правилами.

3.2. Эквивалентные преобразования

В данном разделе рассмотрим вопросы эквивалентных преобразований логических отношений на полных решетках. Такие преобразования могут быть использованы для приведения базы знаний к каноническому виду, более удобному для исследования и практической реализации. Для произвольных решеток соответствующие задачи были изучены в п. 2.4.

Два нетеровых отношения R_1 и R_2, заданные на общей полной решетке $\mathbb{F}$, называются логически эквивалентными (или, в контексте данной главы, просто эквивалентными), если их логические замыкания совпадают. Для таких отношений, как обычно, будем использовать обозначение $R_1 \sim R_2$.

Лемма 1. Пусть R – нетерово отношение на решетке $\mathbb{F}$ и $A_t \xrightarrow{R} B_t$, $t \in T$. Тогда отношение $R' = R \bigcup\limits_{t \in T} \{(A_t, B_t)\}$ логически эквивалентно R.

Доказательство. По определению любое логическое отношение является собственным логическим замыканием. Это утверждение в силу теоремы 3.1.1[1] справедливо и для отношения $\xrightarrow{R}$. Далее, из определения 3.1.3[2] следует, что для любых $R_1 \subseteq R_2$ справедливо $\xrightarrow{R_1} \subseteq \xrightarrow{R_2}$. В данном случае по построению отношения R' выполнены включения $R \subseteq R' \subseteq \xrightarrow{R}$. Переходя к логическим замыканиям и учитывая отмеченное выше, получим, что отношения R и R' имеют общее логическое замыкание $\xrightarrow{R}$. □

[1] Теорема 1 из п. 3.1.

[2] Определение 3 из п. 3.1.

Лемма 1 обосновывает такие эквивалентные преобразования отношения, которые состоят в добавлении или исключении множества пар, логически связанных меньшим отношением.

Теорема 1. Пусть R_1, R_2, R_3, R_4 – нетеровы отношения на полной решетке. Если при этом $R_1 \sim R_2$ и $R_3 \sim R_4$, то $R_1 \bigcup R_3 \sim R_2 \bigcup R_4$.

Доказательство. Согласно теореме 3.1.1, достаточно доказать равенство $\xrightarrow{R_1 \cup R_3} = \xrightarrow{R_2 \cup R_4}$. Пусть $A \xrightarrow{R_1 \cup R_3} B$. Это означает, что существует цепочка $\vec{r}_{AB} = (B_1, ..., B_m)$, логически соединяющая (A, B) посредством отношения $R_1 \bigcup R_3$. Для произвольного k ($1 \le k \le m+1$) рассмотрим пару (B_{k-1}, B_k) (для единообразия полагаем $B_0 = A,\ B_{m+1} = B$). По определению логической цепочки эта пара дистрибутивно связана отношением $R_1 \bigcup R_3$. Этот факт означает (определение 3.1.2), что существует множество пар $\{(A_p^k, B_p^k),\ p \in P\}$ таких, что $A_p^k = B_p^k$ либо $(A_p^k, B_p^k) \in R_1 \bigcup R_3$, причем $B_{k-1} \supseteq A_p^k,\ p \in P$ и $\bigcup_{p \in P} B_p^k \supseteq B_k$.

Если $A_p^k = B_p^k$, то пара (A_p^k, B_p^k) логически связана любым отношением R. Пусть $(A_p^k, B_p^k) \in R_1 \bigcup R_3$. Предположим для определенности, что эта пара принадлежит R_1. Отсюда по условию теоремы ($R_1 \sim R_2$) имеем $A_p^k \xrightarrow{R_2} B_p^k$. Таким образом, все пары (A_p^k, B_p^k) логически связаны отношением $R_2 \bigcup R_4$. По лемме 3.1.5 получаем, что и пара (B_{k-1}, B_k) логически связана отношением $R_2 \bigcup R_4$. Поскольку k – произвольное, то в силу транзитивности отношения $\xrightarrow{R_2 \cup R_4}$ приходим к выводу, что и пара (A, B) логически связана отношением $R_2 \bigcup R_4$, то есть $A \xrightarrow{R_2 \cup R_4} B$.

Таким образом, установлено включение $\xrightarrow{R_1 \cup R_3} \subseteq \xrightarrow{R_2 \cup R_4}$. Обратное включение доказывается аналогично. □

Следствие 1. Пусть R_1, R_2, R – нетеровы отношения на полной решетке. Если при этом $R_1 \sim R_2$, то $R_1 \bigcup R \sim R_2 \bigcup R$.

Теорема 1 и следствие 1 обосновывают принцип локальности эквивалентных преобразований нетеровых отношений на полной решетке. Его суть в том, что заменив некоторую часть данного отношения на эквивалентную, получим эквивалентное отношение.

Теорема 2. Пусть R – нетерово отношение на полной решетке $\mathbb{F}$. Пусть также $\{(A_t, B_t) \in R,\ t \in T\}$, причем при каждом $t \in T$ справедливо $B_t = \bigcup_{p_t \in P_t} B_t^p$ – конечное или бесконечное объединение элементов $B_t^p \in \mathbb{F}$ $(p \in P_t,\ t \in T)$. Тогда отношение R', полученное из R заменой пар $\{(A_t, B_t) \in R,\ t \in T\}$ совокупностью всех пар $\{(A_t, B_t^p),\ p \in P_t,\ t \in T\}$, эквивалентно R.

Доказательство. Введем обозначения $R_1 = \{(A_t, B_t),\ t \in T\}$, $R_2 = \{(A_t, B_t^p),\ p \in P_t,\ t \in T\}$ и $R^- = R \setminus R_1$. Таким образом, $R = R^- \bigcup R_1; R' = R^- \bigcup R_2$. Рассмотрим логические замыкания $\xrightarrow{R_1}$ и $\xrightarrow{R_2}$ отношений R_1 и R_2 соответственно. Поскольку по определению $\xrightarrow{R_1}$ содержит отношение $\supseteq$, то в силу $B_t \supseteq B_t^p$, $p \in P_t$, $t \in T$ и своей транзитивности оно включает также R_2, то есть $R_2 \subseteq \xrightarrow{R_1}$. С другой стороны, отношение $\xrightarrow{R_2}$ из-за дистрибутивности включает R_1. Таким образом, R_1 и R_2 как элементы решетки отношений удовлетворяют лемме 2.2.1. По этой лемме имеем $\xrightarrow{R_1} = \xrightarrow{R_2}$, то есть отношения R_1 и R_2 эквивалентны. Применяя теперь к R_1, R_2, R^- следствие 1, получим, что $R \sim R'$. □

В качестве применения теоремы 2 рассмотрим частный случай – приведение отношения к каноническому виду. Отношение R на полной точечной решетке $\mathbb{F}$ называется каноническим, если оно задано множеством пар вида (A, a), где $A \in \mathbb{F}$, a – точка в $\mathbb{F}$.

Следствие 2. Согласно теореме 2, для любого нетерова отношения на полной точечной решетке существует эквивалентное ему каноническое отношение.

3.3. Логическая редукция

В настоящем разделе исследуются возможности приведения нетеровых отношений на полной решетке к минимальному эквивалентному виду. Доказана теорема о существовании минимальной LP-структуры и указан эффективный способ ее построения. Полученные результаты обосновывают возможности автоматической оптимизации продукционных систем с бесконечными предпосылками и заключениями правил. Соответствующие вопросы для отношений на общих решетках рассматривались ранее в п. 2.5.

Как обычно, логической редукцией данного нетерового отношения R на полной решетке будем называть эквивалентное ему отношение R_0 с минимальным множеством пар. Как и ранее, минимальность здесь понимается в смысле частично упорядоченных множеств – из R_0 невозможно исключить ни одной пары, не нарушив свойства эквивалентности.

Попытаемся для случая полной решетки показать, что логическое замыкание данного нетерового отношения R является транзитивным замыканием некоторого другого отношения $\tilde{R} \supseteq R$. Положительный ответ на этот вопрос дает возможность разработки эффективных алгоритмов построения логического замыкания и редукции, используя на определенном этапе алгоритмы транзитивного замыкания и редукции.

Для нетерового бинарного отношения R на полной решетке $\mathbb{F}$ рассмотрим отношение $\tilde{R}$, построенное по R последовательным выполнением следующих действий (шагов):

• добавить к R все пары вида (A, A), где $A \in \mathbb{F}$ (рефлексивные пары), и обозначить новое отношение R_1;

• добавить к R_1 всевозможные пары (A, B) с элементами вида $A = \bigcup_{t \in T} A_t$, $B = \bigcup_{t \in T} B_t$, где все (A_t, B_t) ($t \in T$) принадлежат R_1;

• объединить полученное отношение с отношением включения $\supseteq$.

Лемма 1. Отношение $\tilde{R}$ логически эквивалентно R.

Доказательство. На каждом указанном выше шаге к исходному отношению добавляются логически связанные пары. Действительно, связь пар, добавляемых на первом и третьем шагах, тривиальна. Из леммы 3.1.4 также следует, что при $(A_t, B_t) \in R_1$, $t \in T$ справедливо $\bigcup_{t \in T} A_t \xrightarrow{R_1} \bigcup_{t \in T} B_t$. Таким образом, эквивалентность нового отношения $\tilde{R}$ получается на основе условия леммы 3.2.1. □

Теорема 1. Для нетерового отношения R на полной решетке $\mathbb{F}$ логическое замыкание представляет собой транзитивное замыкание $\tilde{R}^*$ соответствующего отношения $\tilde{R}$.

Доказательство. По теореме 3.1.1 логическое замыкание R совпадает с отношением $\xrightarrow{R}$.

Покажем вначале, что $\tilde{R} \subseteq \xrightarrow{R}$. Пусть $(A, B) \in \tilde{R}$. Если при этом $A \supseteq B$ либо $(A, B) \in R$, то по определению $A \xrightarrow{R} B$. Остается случай, когда $A = \bigcup_{t \in T} A_t$, $B = \bigcup_{t \in T} B_t$, где каждая из пар (A_t, B_t) содержится в $R \subseteq \xrightarrow{R}$ либо рефлексивна и поэтому также принадлежит $\xrightarrow{R}$. В этом случае по лемме 3.1.4 заключаем, что $A \xrightarrow{R} B$.

Таким образом, $\tilde{R} \subseteq \xrightarrow{R}$. Поскольку транзитивное замыкание $\tilde{R}^*$ – наименьшее транзитивное отношение, содержащее $\tilde{R}$, и $\xrightarrow{R}$ также транзитивно, то имеем включение $\tilde{R}^* \subseteq \xrightarrow{R}$.

Докажем теперь обратное включение. Предположим, что $A \xrightarrow{R} B$. Тогда по определению существует логическая цепочка $\vec{r}_{AB} = (B_1, \ldots, B_m)$, соединяющая (A, B). Для произвольного k ($1 \le k \le m + 1$) рассмотрим пару (B_{k-1}, B_k) (для единообразия полагаем $B_0 = A$, $B_{m+1} = B$). По определению логической цепочки эта пара дистрибутивно связана отношением R. Этот факт означает (см.

определение 3.1.2), что существует множество пар $\{(A_p^k, B_p^k),\ p \in P\}$ таких, что $A_p^k = B_p^k$ либо $(A_p^k, B_p^k) \in R$, причем $B_{k-1} \supseteq A_p^k,\ p \in P$ и $\bigcup_p B_p^k \supseteq B_k$. Обозначим $\tilde{A}_k = \bigcup_p A_p^k$ и $\tilde{B}_k = \bigcup_p B_p^k$. Тогда по построению отношения $\tilde{R}$ имеем $(\tilde{A}_k, \tilde{B}_k) \in \tilde{R}$. Так как $\tilde{R}$ содержит отношение включения $\supseteq$, то справедливо также $(B_{k-1}, \tilde{A}_k) \in \tilde{R}$ и $(\tilde{B}_k, B_k) \in \tilde{R}$. Таким образом, пары $(B_{k-1}, \tilde{A}_k), (\tilde{A}_k, \tilde{B}_k), (\tilde{B}_k, B_k)$ принадлежат $\tilde{R}$. Этими тремя парами заменим в цепочке $\vec{r}_{AB}$ пару (B_{k-1}, B_k). Таким образом поступим для всех $1 \le k \le m+1$. В результате получим из $\vec{r}_{AB}$ транзитивную цепочку в $\tilde{R}$, соединяющую (A, B). Следовательно, по лемме 2.2.3 имеем $(A, B) \in \tilde{R}^*$. □

Следствие 1. Для любой связи $A \xrightarrow{R} B$ элементы соответствующей логической цепочки $\vec{r}_{AB} = (B_1, \ldots, B_m)$ могут быть подобраны специальным образом: каждая пара $(B_{k-1}, B_k) \in \tilde{R}$, причем справедливо $B_{k-1} \supset B_k$ либо $B_{k-1} = \bigcup_{p \in P} A_p^k$ и $B_k = \bigcup_{p \in P} B_p^k$, где $A_p^k = B_p^k$ или $(A_p^k, B_p^k) \in R$ (здесь $1 \le k \le m+1$, $B_0 = A,\ B_{m+1} = B$). Данный факт был попутно установлен при доказательстве второй части теоремы 1. □

Следствие 2. Если $R_1 \subseteq R_2$, то $\xrightarrow{R_1} \subseteq \xrightarrow{R_2}$.

Доказательство. Во-первых, из описания процесса построения $\tilde{R}$ легко заметить, что если $R_1 \subseteq R_2$, то $\tilde{R}_1 \subseteq \tilde{R}_2$. Далее, из свойств транзитивного замыкания (замечание 2.2.1) имеем $\tilde{R}_1^* \subseteq \tilde{R}_2^*$, что в силу теоремы 1 и означает доказываемый факт. □

Переходим непосредственно к вопросу о существовании и построении логической редукции отношений рассматриваемого класса.

Лемма 2. Пусть R – нетерово отношение на полной решетке $\mathbb{F}$. Для того, чтобы R являлось логической редукцией, необходимо и достаточно, чтобы R не содержало ни одной такой пары (A, B), для которой выполнено соотношение $A \xrightarrow{R \setminus \{(A,B)\}} B$.

Доказательство. Пусть отношение R представляет собой логическую редукцию, то есть является минимальным логически эквивалентным себе отношением. Если бы при этом в R существовала пара $A \xrightarrow{R \setminus \{(A,B)\}} B$, то в силу леммы 3.2.1 ее можно было бы исключить, получив при этом меньшее эквивалентное отношение, что невозможно по условию леммы.

Для доказательства обратного утверждения предположим, что не существует ни одной пары $(A, B) \in R$, для которой справедливо $A \xrightarrow{R \setminus \{(A,B)\}} B$. Необходимо доказать, что в этом случае R есть логическая редукция. Предположим противное – пусть существует отношение $R_0 \subset R$, эквивалентное R, и $(A,B) \in R \setminus R_0$. Тогда в силу эквивалентности $R \sim R_0$ справедливо $A \xrightarrow{R_0} B$. Так как отношение R_0 не содержит пару (A,B), то $R_0 \subseteq R \setminus \{(A,B)\}$, и логическая связь $A \xrightarrow{R_0} B$ противоречит сделанному предположению о том, что таких пар (A,B) в R нет. Полученное противоречие доказывает требуемое утверждение. □

Для нетерового отношения R на полной решетке рассмотрим отношение $\underset{\sim}{R}$, построенное по данному R последовательным выполнением шагов, обратных построению $\tilde{R}$, а именно:

- исключить из R содержащиеся в нем пары вида $A \supset B$ и обозначить новое отношение R_{-1};
- исключить из R_{-1} всевозможные пары (A,B) с элементами вида $A = \bigcup\limits_{t \in T} A_t$, $B = \bigcup\limits_{t \in T} B_t$, где все (A_t, B_t) ($t \in T$) принадлежат R_{-1} и не совпадают с (A,B);
- исключить из полученного отношения все рефлексивные пары.

Лемма 3. Отношение $\underset{\sim}{R}$ логически эквивалентно R.

Доказательство аналогично доказательству леммы 2. □

Как и в случае отношений на общих решетках (глава 2), поскольку решетка ациклична, удаление всех указанных выше «объединенных» пар (A,B) приводит к одному и тому же результату независимо от порядка удаления. По этой причине можно утверждать о возможности одновременного удаления всех таких пар.

Следующая теорема указывает достаточное условие существования и способ построения логической редукции данного отношения.

Теорема 2. Пусть для нетерового отношения R, заданного на полной решетке $\mathbb{F}$, построено соответствующее отношение $\tilde{R}$. Тогда, если для $\tilde{R}$ существует транзитивная редукция R^0, то соответствующее ей отношение $\underset{\sim}{R}^0$ представляет собой логическую редукцию исходного отношения R.

Доказательство. Из лемм 1, 3, а также леммы 3.2.1, следует, что указанное в теореме отношение $\underset{\sim}{R}^0$ логически эквивалентно R. Осталось показать, что $\underset{\sim}{R}^0$ является логической редукцией. Для этого достаточно проверить выполнение для $\underset{\sim}{R}^0$ условия леммы 2.

Предположим противное, что для некоторой пары $(A,B) \in \underset{\sim}{R}^0$ имеет место $A \xrightarrow{\underset{\sim}{R}^0 \setminus \{(A,B)\}} B$. Обозначим $R_0 = \underset{\sim}{R}^0 \setminus \{(A,B)\}$ и рассмотрим соответствующую этой связи логическую цепочку $\vec{r}_{AB} = (B_1, \ldots, B_m) \in R_0$. Согласно следствию 1, эта цепочка может быть составлена таким образом, что каждая пара $(B_{k-1}, B_k) \in \tilde{R}_0$, причем справедливо $B_{k-1} \supset B_k$ либо $B_{k-1} = \bigcup_{p\in P} A_p^k$ и $B_k = \bigcup_{p\in P} B_p^k$, где $A_p^k = B_p^k$ или $(A_p^k, B_p^k) \in R_0$ ($1 \le k \le m+1$, $B_0 = A$, $B_{m+1} = B$). В нашем случае, поскольку отношение $\underset{\sim}{R}^0$ вложено в транзитивную редукцию отношения $\tilde{R}$, оно не может содержать транзитивной в $\tilde{R}_0 \subseteq \tilde{R}$ пары (A,B). Это означает, что в рассматриваемой логической цепочке $m = 0$ и для пары (A,B) остаются варианты $A \supset B$ либо $A = \bigcup_{p\in P} A_p^1$ и $B = \bigcup_{p\in P} B_p^1$, где $A_p^1 = B_p^1$ или $(A_p^1, B_p^1) \in R_0$. Поскольку процесс построения отношения $\underset{\sim}{R}^0$ исключает также вложенные ($A \supset B$) и объединенные ($A = \bigcup_{p\in P} A_p^1$ и $B = \bigcup_{p\in P} B_p^1$) пары, то остается лишь вариант «унарного» объединения, когда $A = B$ либо $(A,B) \in R_0$. Наконец, замечаем, что рефлексивных пар ($A = B$) также нет в $\underset{\sim}{R}^0$ по построению, а пары (A,B) нет в $R_0 = \underset{\sim}{R}^0 \setminus \{(A,B)\}$.

С учетом изложенного выше, приходим к заключению, что вывод $A \xrightarrow{\underset{\sim}{R}^0 \setminus \{(A,B)\}} B$ невозможен, и, таким образом, отношение $\underset{\sim}{R}^0$ является логической редукцией R. □

Заметим, что, как и в главе 2 (п.2.5), в данном разделе требование существования транзитивной редукции отношения $\tilde{R}$ может оказаться избыточным для существования логической редукции исходного отношения R. При конечном множестве R из него можно последовательно удалить «лишние» пары, получив в результате логическую редукцию. Если же решетка $\mathbb{F}$ бесконечна, то, объединяя R с отношением $\supseteq$, можно построить не имеющее транзитивной редукции отношение $\tilde{R}$. Для усиления теоремы 2 в LP-структуре можно использовать вместо $\supseteq$ отношение вида $\supseteq_R$, содержащее лишь необходимое для получения логической редукции R подмножество $\supseteq$. Оставим эту идею в качестве рекомендации для конкретных приложений.

Заметим также, что бесконечность рассматриваемых множеств правил не означает на практике хранение бесконечного количества данных. Можно реализовать эффективные функции, реализующие правила продукционной системы с переменными и возвращающие псевдо-бесконечные наборы пар при повторных вызовах. Существует также возможность реализовать механизмы проверок того, относится ли заданная пара к теоретически добавляемым, например, при построении отношения $\tilde{R}$.

3.4. Логические уравнения на полных решетках

В настоящем разделе рассматривается класс продукционно-логических уравнений, определяемый LP-структурами на полных решетках с нетеровыми отношениями. Как и в п.2.6, исследуются вопросы о разрешимости, о количестве решений, а также методы решения этих уравнений. Нахождение решения продукционно-логического уравнения эквивалентно обратному логическому выводу в системе с бесконечными предпосылками и заключениями правил.

Вначале напомним несколько понятий, которые в данном разделе будут относиться к нетеровым отношениям на полной решетке.

Пусть дано нетерово отношение R на полной решетке $\mathbb{F}$. Тогда запись $A \xrightarrow{R} B$ означает, что B является образом A, а A – прообразом B при отношении $\xrightarrow{R}$. В этом контексте будем использовать обозначение $R(A) = B$.

Для данного $B \in \mathbb{F}$ минимальным прообразом при отношении $\xrightarrow{R}$ называется такой элемент $A \in \mathbb{F}$, что $A \xrightarrow{R} B$ и A является минимальным, а именно – не содержит никакого другого $A_1 \in \mathbb{F}$, для которого $A_1 \xrightarrow{R} B$.

В оставшейся части текущего раздела рассматриваются лишь полные точечные решетки и заданные на них нетеровы отношения.

Определение 1. Точка x решетки $\mathbb{F}$ называется начальной при отношении R, если в R нет ни одной такой пары (A,B), что x содержится в B и не содержится в A. Элемент X точечной решетки $\mathbb{F}$ называется начальным, если все его точки являются начальными (при отношении R). Подмножество $\mathbb{F}_0(R)$ (будем обозначать $\mathbb{F}_0$, если это не вызовет неоднозначностей) точечной решетки $\mathbb{F}$, состоящее из всех начальных элементов $\mathbb{F}$, называется начальным множеством решетки $\mathbb{F}$ (при отношении R).

Начальное множество $\mathbb{F}_0$ образует подрешетку в $\mathbb{F}$.

Рассматривается уравнение

$$R(X) = B, \tag{1}$$

где $B \in \mathbb{F}$ – заданный элемент, $X \in \mathbb{F}$ – неизвестный.

Определение 2. Частным решением X уравнения (1) называется любой минимальный прообраз элемента B, содержащийся в $\mathbb{F}_0$. Приближенным (частным) решением X уравнения (1) называется любой прообраз элемента B, содержащийся в $\mathbb{F}_0$. Общим решением уравнения (1) называется совокупность всех его частных решений $\{X_s\}$, $s \in S$.

По определению точное решение является и приближенным. Очевидно, в полной решетке приближенное решение всегда содержит хотя бы одно точное решение.

Уравнения вида (1) будем называть продукционно-логическими (в рамках данной главы – просто логическими) уравнениями на полных решетках.

Аналогично методологии, представленной в главе 2, рассмотрим вопрос о возможности «расщепления» уравнения (1) на некоторое множество уравнений с более простыми правыми частями.

Лемма 1. Пусть X_p – частное решение уравнения вида (1) с правой частью B_p, $p \in P$. Тогда элемент $\bigcup_{p\in P} X_p$ является приближенным решением уравнения

$$R(X) = \bigcup_{p\in P} B_p \qquad (2)$$

Доказательство. При сформулированном условии непосредственно из леммы 3.1.4 следует логическая связь $\bigcup_{p\in P} X_p \xrightarrow{R} \bigcup_{p\in P} B_p$, что и означает справедливость доказываемого утверждения. □

Теорема 1. Пусть дано нетерово отношение R на полной решетке $\mathbb{F}$. Пусть также $\{X_p^{s_p},\, s_p \in S_p\}$ – общее решение уравнения вида (1) с правой частью B_p, $p \in P$. Тогда общее решение уравнения (2) представляет собой множество всех элементов вида $\bigcup_{p\in P} X_p^{s_p}$, $s_p \in S_p$, из которого исключены элементы, содержащие другие элементы этого же множества.

Доказательство. По утверждению леммы 1, каждый элемент $\bigcup_{p\in P} X_p^{s_p}$, $s_p \in S_p$ является прообразом элемента $\bigcup_{p\in P} B_p$, то есть содержит хотя бы одно частное решение уравнения (2). Покажем теперь, что это уравнение не имеет частных решений, отличных от вида $\bigcup_{p\in P} X_p^{s_p}$. Предположим противное – пусть некоторый $Z \in \mathbb{F}_0$, являясь частным решением (2), не совпадает ни с одним элементом вида $\bigcup_{p\in P} X_p^{s_p}$. При этом Z не может также содержать ни одного элемента вида $\bigcup_{p\in P} X_p^{s_p}$, иначе он был бы приближенным, но не частным решением, которое по определению минимально. Отсюда следует, что Z хотя бы при одном некотором p_0 не содержит ни одного элемента множества $\{X_{p_0}^{s},\, s \in S_{p_0}\}$.

С другой стороны, поскольку $Z \xrightarrow{R} \bigcup_{p\in P} B_p$, то $Z \xrightarrow{R} B_p,\ p \in P$, то есть Z содержит хотя бы по одному частному решению уравнений с правой частью $B_p,\ p \in P$, включая и B_{p_0} . Поскольку все частные решения при правой части B_{p_0} содержатся в множестве $\{X_{p_0}^s,\ s \in S_{p_0}\}$ (по условию – это общее решение), то Z обязательно содержит хотя бы один элемент этого множества.

Таким образом, получено противоречие, которое опровергает сделанное выше предположение о том, что уравнение (2) имеет решение Z , отличное от вида $\bigcup_{p\in P} X_p^s$. □

Рассмотрим методы решения логических уравнений вида (1).

Далее в настоящем разделе будем предполагать, что R является каноническим нетеровым отношением на полной точечной решетке $\mathbb{F}$, не содержащим пар отношения $\supseteq$. Соответственно, правая часть B уравнения (1) представляет собой объединение ее точек.

Подобно используемому в главе 2, введем структурное расслоение исходного отношения R на виртуальные слои (частичные отношения) $\{R_t \mid t \in T\}$. Для этого предварительно разобьем R на непересекающиеся подмножества, каждое из которых образовано парами вида (A, x_p) с одним и тем же точечным элементом x_p в качестве правой части. Такое разбиение возможно, поскольку R является каноническим. Обозначим эти подмножества R^p , что соответственно их элементу $x_p, p \in P$.

Определение 3. Слоем R_t в нетеровом отношении R на полной точечной решетке называется подмножество отношения R , образованное парами, взятыми по одной из каждого непустого $R^p,\ p \in P$.

Замечание 1. Два слоя, отличающиеся хотя бы одной парой, считаются различными.

Замечание 2. Каждый слой содержит максимально возможное подмножество пар из R с уникальными правыми частями. Добавление к слою еще одной пары нарушило бы это условие.

Замечание 3. Любое подмножество пар в R с уникальными правыми частями принадлежит некоторому слою.

Замечание 4. В общем случае слои имеют непустые пересечения. Объединение всех слоев равно R .

Общее количество (кардинальное число) слоев определяется выражением $\prod_{p\in P} N_p$, где N_p – кардинальное число подмножества пар отношения R с правой частью x_p .

Будем констатировать, что логическая цепочка $\vec{r}_{AB}$ принадлежит в R некоторому слою R_t, если все нерефлексивные пары, определяющие логическую связь $A \xrightarrow{R} B$ (нерефлексивные пары, содержащиеся в элементах соответствующего вектора вывода), содержатся в одном и том же слое R_t.

Лемма 2. Если $A \xrightarrow{R} B$, то существует логическая цепочка $\vec{r}_{AB}$, принадлежащая в R некоторому слою R_t.

Утверждение этой леммы непосредственно вытекает из леммы 3.1.2 и замечания 3. □

Следствие 1. Логическое замыкание канонического нетерового отношения R на полной точечной решетке $\mathbb{F}$ равно объединению логических замыканий его слоев, то есть $\xrightarrow{R} = \bigcup_{t\in T} \xrightarrow{R_t}$.

Доказательство. Поскольку $R_t \subseteq R$, то согласно следствию 3.3.2 имеем $\xrightarrow{R_t} \subseteq \xrightarrow{R} (\forall t \in T)$. Обратно, если $A \xrightarrow{R} B$, то по лемме 2 справедливо $A \xrightarrow{R_t} B$ при некотором $t \in T$. Следовательно, $(A, B) \in \bigcup_{t\in T} \xrightarrow{R_t}$. □

Будем считать, что решение X уравнения (1) (точное или приближенное) порождается в R некоторым слоем R_t, если $X \xrightarrow{R_t} B$.

Замечание 5. Согласно лемме 2 и следствию 1, любое частное решение уравнения (1) порождается в R некоторым слоем R_t.

Замечание 6. Из следствия 1 вытекает, что для нахождения решения уравнения (1) в некотором слое R_t достаточно вместо (1) решить аналогичное уравнение с отношением R_t.

Последние замечания не гарантируют, что два различных слоя не могут порождать одного и того же решения. Кроме того, могут существовать слои, дающие частное решение для R_t, но приближенное для R. Некоторые слои могут вообще не давать решений. Однако, справедливо утверждение о том, что один слой не может порождать более одного точного решения. Чтобы установить этот факт, вначале докажем вспомогательные утверждения.

Лемма 3. Если X – частное решение уравнения (1), то существует соответствующий паре $X \xrightarrow{R} B$ вектор вывода $\vec{V}_{XB}$, имеющий вид

$$\begin{aligned}\vec{V}_{AB} = \Big(&\{(C_1, C_1), (A_{p_1,1}, b_{p_1,1})\}, \{(C_2, C_2), (A_{p_2,2}, b_{p_2,2})\}, \ldots, \\ &\{(C_{m+1}, C_{m+1}), (A_{p_{m+1},m+1}, b_{p_{m+1},m+1})\}\Big),\end{aligned} \tag{3}$$

где каждый элемент с номером k $(1 \le k \le m+1)$ образован одной рефлексивной парой (C_k, C_k) и непустым множеством пар

$\{(A_{p_k,k}, b_{p_k,k}) \mid p_k \in P_k\}$. При этом $b_{p_k,k}$ – попарно различные точки $\mathbb{F}$ $(p_k \in P_k;\ 1 \le k \le m+1)$;

$$X = \bigcup_{p_1 \in P_1} A_{p_1,1} \cup C_1 \ ; \tag{4}$$

$$B = \bigcup_{p_{m+1} \in P_{m+1}} b_{p_{m+1},m+1} \cup C_{m+1} \ ; \tag{5}$$

$$\bigcup_{p_k \in P_k} b_{p_k,k} \cup C_k = \bigcup_{p_{k+1} \in P_{k+1}} A_{p_{k+1},k+1} \cup C_{k+1} \ \ (1 \le k \le m); \tag{6}$$

Заметим, что вектор (3) является также каноническим в смысле определения 3.1.4.

Доказательство. Опишем, как может быть получен вектор (3).

Вначале, согласно замечанию 5, построим для пары (X, B) обычный канонический вектор вывода (вида (3.1.2)), образованный парами некоторого слоя R_t и рефлексивными парами элементов $\mathbb{F}$. Затем в каждом элементе вектора (множестве пар) заменим все рефлексивные пары единственной рефлексивной парой, содержащей их объединение (объединяются отдельно левые и правые части). Поскольку рассматриваются точечная решетка $\mathbb{F}$ и каноническое отношение R, то полученный вектор имеет вид (3). Для доказательства леммы осталось добиться выполнения условий (4)–(6), которые представляют собой усиленный вариант соотношений (3.1.3)–(3.1.4).

Условие (4) выполняется сразу в силу минимальности X, который по условию леммы является частным решением (см. также соотношение вида (3.1.3) при $k = 1$). Для доказательства соотношений (5)–(6) будем последовательно просматривать элементы вектора (3) справа налево. Вначале положим $k = m + 1$, $B_k = B$.

Исключим из k-го множества пары вида (A, b) такие, что $b \notin B_k$. Исключим такие же элементы b и из обеих частей рефлексивной пары этого множества. Очевидно, что условия вида (3.1.3)–(3.1.4) при этом не нарушатся, однако будет выполнено еще одно из условий (5)–(6) (при данном значении k). Далее, переходя к очередному элементу вектора (множеству пар, справа налево), уменьшим k на единицу и положим $B_k = \bigcup\limits_{p_{k+1} \in P_{k+1}} A_{p_{k+1},k+1} \cup C_{k+1}$.

Описанную в последнем абзаце процедуру проведем для всех элементов вектора вывода. Таким образом получим искомый вектор $\vec{V}_{XB}$. □

Замечание 7. Согласно замечаниям 5–6, утверждение леммы остается верным, если в ее условии отношение R заменить на R_t.

Лемма 4. Если для некоторых $X_1, X_2 \in \mathbb{F}_0$ и $B \in \mathbb{F}$ построены векторы вывода $\vec{V}_{X_1B}$ и $\vec{V}_{X_2B}$ вида (3), все нерефлексивные пары которых содержатся в одном и том же слое R_t, то $X_1 = X_2$.

Доказательство. Предположим противное, что $X_1 \neq X_2$. Пусть, для определенности, существует точка $a_0 \in X_1$ и $a_0 \notin X_2$. Рассмотрим вначале вектор $\vec{V}_{X_1B}$. Будем просматривать его элементы слева направо и при этом строить некоторую специальную последовательность нерефлексивных пар.

Согласно условию (4), точка a_0, как и любая точка элемента X_1, содержится в левой части хотя бы одной пары первого элемента вектора. Если существует такая нерефлексивная пара (A_1, a_1) ($a_0 \in A_1$), то добавим эту пару к формируемой последовательности и, двигаясь в векторе вправо, будем искать аналогичную пару для a_1. Если же это (C_1, C_1), то по условию (6) a_0 аналогично перейдет в следующий элемент вектора. Пройдя весь вектор, получим конечную последовательность нерефлексивных пар отношения R (возможно, пустую) $\{(A_i, a_i) \mid i = 1, \ldots, n\}$, причем $a_i \in A_{i+1},\ a_i \notin \mathbb{F}_0\ (0 \leq i \leq n)$, $A_{n+1} = B$. Если эта последовательность окажется пустой ($n = 0$), то будем иметь $a_0 \in B$.

Рассмотрим теперь вектор $\vec{V}_{X_2B}$. Его элементы будем просматривать справа налево, пытаясь в этом векторе выделить построенную выше последовательность $\{(A_i, a_i)\}$. Поскольку $a_n \in B$, то в силу (5) точка a_n содержится в правой части по крайней мере одной пары последнего элемента вектора $\vec{V}_{X_2B}$. Если существует такая нерефлексивная пара, то это может быть лишь (A_n, a_n), так как слой R_t не может содержать двух пар с общей правой частью a_n. В этом случае переходим в предыдущий элемент вектора с новой отправной точкой $a_{n-1} \in A_n$ – правой частью предыдущего элемента последовательности. Если же это (C_{m+1}, C_{m+1}), то по условию (6) в предыдущий столбец аналогично переходит a_n. Таким образом будем продвигаться справа налево в векторе $\vec{V}_{X_2B}$ и в последовательности $\{A_i, a_i\}$.

Этот процесс может завершиться в одном из трех случаев. Если элементы вектора $\vec{V}_{X_2B}$ закончатся одновременно с последовательностью, то получим $a_0 \in X_2$, что будет противоречить первоначальному предположению о том, что $a_0 \notin X_2$ (см. начало доказательства). Если же элементы вектора закончатся раньше, чем исчерпается последовательность, то в этом случае соотношение $a_i \in X_2,\ i > 0$ будет противоречить тому, что $a_i \notin \mathbb{F}_0$, так как по условию X_2 целиком состоит из начальных точек, а a_i таковой не является. Наконец, в последнем варианте, когда последовательность закончится раньше элементов вектора, в векторе $\vec{V}_{X_2B}$ окажется пара вида (A_0, a_0). Этот факт противоречит тому, что a_0 – начальная точка. Итак, в каждом возможном случае приходим к противоречию, что и доказывает лемму. □

Следствие 2. Один слой R_t не может содержать двух различных частных решений уравнения (1).

Доказательство. Предположим, что существуют два решения. Тогда по лемме 3 и замечанию 7 для этих решений существуют векторы вывода (3), нерефлексивные пары которых принадлежат R_t. Отсюда по лемме 4 получаем, что эти решения совпадают. □

Объединяя полученные результаты, можно сформулировать следующее утверждение.

Теорема 2. Для нахождения общего решения уравнения (1) достаточно найти частное решение X_t в каждом слое R_t, если оно существует, после чего из полученного множества решений исключить элементы, содержащие другие элементы этого же множества.

Утверждение теоремы сразу следует из замечания 5 и леммы 4. □

Следствие 3. Количество частных решений уравнения (1) оценивается сверху выражением $\prod_{p\in P} N_p$, где N_p – кардинальное число подмножества пар отношения R с правой частью x_p, а P определяет все такие подмножества (см. замечание 4). □

Теорема 2 и замечание 6 позволяют свести решение уравнения (1) к задаче нахождения частного решения уравнения

$$R_t(X) = B\,, \tag{7}$$

где B – неначальный элемент решетки $\mathbb{F}$, R_t – произвольный слой в R.

Согласно следствию 2, отношение, обратное к $\xrightarrow{R_t}$, является однозначным отображением $f_t : \mathbb{F} \to \mathbb{F}$ с некоторой областью определения $D(f_t) \subseteq \mathbb{F}$. Рассмотрим некоторые его свойства.

Лемма 5. Отображение f_t является полным $\bigcup$-гомоморфизмом, то есть $f_t(\bigcup_{p\in P} B_p) = \bigcup_{p\in P} f_t(B_p),\ \forall B_p \in D(f_t),\ p\in P$.

Доказательство. Пусть $\vec{V}_{X_pB_p}$, $p\in P$ – канонические векторы вывода вида (3) соответственно для связей $X_p \xrightarrow{R_t} B_p,\ p\in P$. На начальном этапе каждый вектор, имеющий не самую большую длину, дополним справа повторяющимся элементом – множеством, состоящим из единственной рефлексивной пары, сформированной на основе объединения правых частей всех пар его последнего элемента. Объединим полученные векторы описанным далее способом.

Будем строить последовательно слева направо результирующий канонический вектор вывода $\vec{V}$. Очередную компоненту $\vec{V}$ получим как объединение всех соответствующих по номеру компонент векторов $\vec{V}_{X_p B_p}$, $p \in P$ (рефлексивные пары совмещаются). Как и при доказательстве леммы 3.1.4, условия (3.1.3)–(3.1.4) для $\vec{V}$ окажутся выполненными. После добавления к $\vec{V}$ каждой компоненты будем производить его «чистку» аналогично доказательству леммы 3.1.2 (см. также доказательство леммы 3.1.3) с тем, чтобы он оставался каноническим. В силу нетеровости отношения R, процесс увеличения длины вектора $\vec{V}$ рано или поздно завершится. Аналогично доказательству леммы 3.1.4, полученный вектор будет осуществлять вывод $\bigcup_{p\in P} X_p \xrightarrow{R} \bigcup_{p\in P} B_p$.

Затем в новом векторе исключим лишние нерефлексивные пары подобно тому, как это описано в доказательстве леммы 4. В результате получим вектор $\vec{V}_{X_{12}B_{12}}$, который является вектором вида (3) для пары $(\bigcup_{p\in P} X_p, \bigcup_{p\in P} B_p)$. □

Следствие 4. Отображение f_t является изотонным, то есть если $B_1 \supseteq B_2$, то $f_t(B_1) \supseteq f_t(B_2)$. □

Таким образом, основываясь на лемме 5, для решения уравнения (7) достаточно решить уравнение с каждой точкой элемента B в качестве правой части. В развитие этой идеи рассмотрим задачу нахождения частного решения следующего уравнения:

$$R_t(X) = b\,, \tag{8}$$

где b – неначальная точка полной решетки $\mathbb{F}$, R_t – произвольный слой в R.

Рассмотрим методы решения этой задачи. Покажем, что она эквивалентна задаче на ориентированном графе (возможно, бесконечном, но с конечной высотой) перечисления всех входных вершин, из которых достижима данная вершина.

Вначале построим такой граф G_{R_t}. Каждой точке решетки $\mathbb{F}$, участвующей в отношении R, сопоставим вершину графа. Далее для каждой пары (A, a) рассматриваемого слоя R_t построим дуги, ведущие из всех вершин, соответствующих точкам A, в вершину, соответствующую данной a. Для краткости будем отождествлять точки решетки $\mathbb{F}$ и соответствующие им вершины в графе. Конечность высоты графа обеспечивается свойством нетеровости отношения R.

В полученном графе G_{R_t} выберем вершину b. Рассмотрим подграф $G_{R_t,b} \subseteq G_{R_t}$, содержащий все вершины, из которых достижима вершина b (включая саму b), и все дуги, соединяющие такие вершины.

Лемма 6. Если подграф $G_{R_t,b}$ не содержит ориентированных циклов, то множество всех его входных вершин (не имеющих входящих дуг) соответствует частному решению уравнения (8).

Доказательство. Предположим вначале, что граф $G_{R_t,b}$ ациклический. Покажем, как по нему построить соответствующий вектор $\vec{V}_{Xb}$ вида (3). Построение проведем по индукции по компонентам вектора, справа налево.

Последний компонент (множество пар) вектора сформируем из единственной пары – $(A_{m+1}, b) \in R_t$, где элемент A_{m+1} состоит из всех точек, непосредственно связанных дугами с вершиной b. Очевидно, условие (5) при этом выполнено. Предположим теперь, что построен очередной элемент вектора с парами $(A_{p_k,k}, b_{p_k,k}) \in R_t$, $b_{p_k,k}$ – попарно различные точки $\mathbb{F}$ $(p_k \in P_k)$ и (C_k, C_k), $C_k \in \mathbb{F}$ $(1 < k \leq m+1)$, причем выполнены условия (5)–(6). Тогда для каждой точки a каждого элемента $A_{p_k,k}$ найдем все вершины графа, дуги из которых входят в a. Сформируем из этих вершин (в виде объединения) элемент $A \in \mathbb{F}$ и поместим пару $(A, a) \in R_t$ в следующий элемент вектора (с номером $k-1$). Если же точка $a \in A_{p_k,k}$ окажется начальной (соответствует входной вершине графа), то добавим ее к рефлексивной паре старого элемента вектора. Это означает, что в новый элемент вместо пары (C_k, C_k) войдет пара вида $(C_k \bigcup a, C_k \bigcup a)$. По построению нового элемента условие (6) окажется выполненным. Так как граф не содержит циклов и имеет конечную высоту, то через конечное число шагов окажется, что все точки левых частей нерефлексивных пар очередного элемента окажутся начальными. На этом процесс построения вектора $\vec{V}_{Xb}$ заканчивается.

Поскольку таким способом будут пройдены все дуги графа $G_{R_t,b}$, то можно утверждать, что объединение X левых частей пар первого элемента вектора состоит из *всех* начальных точек, соответствующих входным вершинам графа $G_{R_t,b}$. С другой стороны, X – решение уравнения (8). По лемме 4 оно является точным решением. □

Замечание 8. Описанный в доказательстве процесс построения вектора $\vec{V}_{Xb}$ напоминает процедуру обхода всех вершин графа $G_{R_t,b}$. Однако такой процесс не полностью соответствует ей, поскольку некоторые точки (вершины) могут обрабатываться более одного раза. Он приведен для доказательства леммы и носит теоретический характер.

Лемма 7. Если X – решение уравнения (8), то в любом его каноническом векторе вывода $\vec{V}_{Xb}$ вида (3) содержатся точки, соответствующие всем вершинам графа $G_{R_t,b}$.

Доказательство. Предположим противное, а именно – что для некоторой вершины y_0 графа $G_{R_t,b}$ соответствующая точка y_0 не содержится ни в одной паре вектора $\vec{V}_{Xb}$. По построению графа существует путь $y_1,\dots,y_{n-1}$ из вершины y в вершину b (можно считать $b=y_n$). Подобно доказательству леммы 4, будем просматривать элементы вектора $\vec{V}_{Xb}$ справа налево, пытаясь выделить в ней указанную выше последовательность $\{y_i\}$. Положим $i=n$.

По условию (6) (при $i=n$ – (5)) y_i содержится в правой части пары очередного (при $i=n$ – последнего) элемента вектора $\vec{V}_{Xb}$. Если существует такая нерефлексивная пара (Y_i, y_i) (при $i=n$ это именно так), то в этом случае переходим в предыдущий элемент вектора с новой отправной точкой $y_{i-1}\in Y_i$ (эта принадлежность верна, так как Y_i содержит все вершины, предшествующие y_i, в том числе и y_{i-1}). Если же это (C_{k_i}, C_{k_i}), то по условию (6) в предыдущий элемент аналогично переходит y_i. Продолжая этот процесс, будем продвигаться справа налево в векторе и в последовательности $\{y_i\}$.

Как и ранее, рассмотрим возможные варианты завершения процесса. Если последовательность закончится ранее элементов вектора или одновременно с ними, то получим, что y_0 содержится в векторе, что будет противоречить сделанному в начале доказательства предположению. Если же элементы вектора закончатся раньше, чем исчерпается последовательность, то тогда соотношение $y_i\in X,\ i>0$ будет противоречить тому, что $y_i\notin\mathbb{F}_0$, так как по определению X целиком состоит из начальных точек. Таким образом, в обоих возможных случаях приходим к противоречию, что и доказывает лемму. □

Лемма 8. Если в графе $G_{R_t,b}$ есть ориентированный цикл, то уравнение (8) не имеет решений.

Доказательство. Предположим противное – пусть существует решение X уравнения (8). Тогда построим соответствующий ему канонический вектор вывода $\vec{V}_{Xb}$ вида (3). Пусть y – некоторая вершина графа $G_{R_t,b}$, содержащаяся в его цикле. По предыдущей лемме в векторе $\vec{V}_{Xb}$ существует пара, в левой или правой части которой присутствует соответствующая точка y.

Предположим для определенности, что в элементе вектора с номером k находится пара, содержащая в своей правой части точку y (вариант нахождения y в левой части некоторой пары с помощью условия (6) сводится к случаю правой части). Если это (C_k, C_k), то согласно (6) получим, что и в элементе с номером $k-1$ присутствует пара с точкой y в правой части. Если же это нерефлексивная пара (Y, y), то рассмотрим вершину y_1, предшествующую y в цикле графа. Соответственно имеем $y_1 \in Y$, поскольку Y содержит все вершины графа, непосредственно предшествующие y. В силу свойства (6), в $k-1$ элементе вектора $\vec{V}_{Xb}$ содержится пара, содержащая в правой части y_1.

Таким образом, за один шаг перемещаемся в векторе на один элемент влево, а в графе – не более чем на одну вершину назад в цикле. Продолжая этот процесс, через конечное число шагов (учитывая еще и условие (4)) выйдем из первого элемента в векторе. Это будет означать, что очередная рассматриваемая точка y_i окажется в X, но в графе вершина y_i останется в том же цикле. Таким образом, приходим к противоречию, так как в X содержатся только начальные точки, а в цикле графа – лишь невходные вершины. □

Объединяя утверждения лемм 6 и 8, можно считать доказанным следующее утверждение.

Теорема 3. Уравнение (8) имеет не более одного решения. Если граф $G_{R_t,b}$ не содержит циклов, то единственное решение уравнения состоит из всех точек, соответствующих входным вершинам графа. Если $G_{R_t,b}$ содержит хотя бы один цикл, то уравнение решений не имеет. □

Данная теорема завершает обоснование пошагового процесса решения исходного логического уравнения (1).

В настоящей главе формально введены и исследованы LP-структуры на полных решетках. Полученные результаты обобщают основную часть материала предыдущей главы и могут быть применены для моделирования стандартных продукционных систем с бесконечными предпосылками и заключениями правил. В следующей главе будут рассмотрены LP-структуры, соответствующие моделям продукционных систем с более сложной логикой.

Глава 4
РАСШИРЕННЫЕ МОДЕЛИ

Данная глава посвящена развитию предложенной ранее общей теории LP-структур применительно к некоторым расширенным моделям продукционных систем (см. также пп. 1.4–1.6). Эти модели являются более сложными, так как поддерживают большее число операций в выражениях предпосылки и заключения правил, и, соответственно, более разнообразные правила дедукции. В связи с тем, что механизмы продукционно-логического вывода в расширенных моделях усложняются, для определения соответствующих понятий и доказательства их свойств в данной главе используется рекурсивный подход, аналогичный принятому в пропозициональном исчислении. В пп. 4.1–4.2 для операций на LP-структурах используются традиционные для теории решеток обозначения $\leq$, $\geq$, $\wedge$ и $\vee$.

4.1. LP-структуры нулевого порядка

В настоящем разделе определяется и исследуется LP-структура, логика которой расширена до модели полного набора логических связок пропозиционального языка [187] – импликации, конъюнкции, дизъюнкции и отрицания. Отсюда происходит название такой алгебраической системы – «LP-структура нулевого порядка». В качестве ее основы используется булева решетка (см. п. 1.2). Исследуются следующие основные вопросы: существование LP-замыкания; эквивалентные преобразования; структура логических связей. Доказывается теорема о существовании логической редукции произвольного отношения и описан процесс ее построения. Один из этапов данного процесса связан с нахождением транзитивной редукции бинарного отношения.

4.1.1. Логическое замыкание и эквивалентные преобразования

В текущем подразделе рассматриваются бинарные отношения на булевой решетке. Введем несколько базовых понятий.

Отношение R на решетке с дополнениями $\mathbb{F}$ называется контрапозиционным, если из $(a,b) \in R$ следует $(b',a') \in R$, где $a,b \in \mathbb{F}$.

Заданное на решетке отношение R называется:

• дистрибутивным относительно операции $\wedge$ (или $\wedge$-дистрибутивным), если для любых пар вида $(a,b_1),(a,b_2) \in R$ справедливо $(a,b_1 \wedge b_2) \in R$;

• дистрибутивным относительно операции $\vee$ (или $\vee$-дистрибутивным), если для любых $(a_1,b),(a_2,b) \in R$ выполнено $(a_1 \vee a_2,b) \in R$;

• дистрибутивным, если оно одновременно обладает обоими свойствами.

Отношение на булевой решетке именуется продукционно-логическим (в данном разделе – просто логическим), если оно содержит $\leq$, а также является контрапозиционным, дистрибутивным и транзитивным. Логическим замыканием произвольного отношения R называется наименьшее логическое отношение, содержащее R. Как обычно, под LP-структурой подразумевается решетка с заданным на ней логическим отношением.

Замечание 1. Для логического отношения R справедливо следующее утверждение: если $(a,b) \in R$, то $(a,a \wedge b) \in R$ и $(a \vee b,b) \in R$. Оно следует из дистрибутивности R и принадлежности $(a,a),(b,b)$ отношению R.

Очевидно, что базовое отношение $\leq$ на булевой решетке само является логическим.

Два отношения R_1 и R_2, определенные на общей решетке, называются логически эквивалентными (в контексте текущего раздела – просто эквивалентными), если их логические замыкания совпадают. Этот факт, как обычно, обозначаем $R_1 \sim R_2$. Логической редукцией данного отношения R называется эквивалентное ему минимальное отношение R_0.

Определение 1. Пусть задано отношение R на булевой решетке $\mathbb{F}$. Будем констатировать, что упорядоченная пара $a,b \in \mathbb{F}$ логически связана отношением R (обозначим этот факт $a \xrightarrow{R} b$), если выполнено одно из следующих условий:

1) $(a,b) \in R$;
2) $a \leq b$;
3) $b' \xrightarrow{R} a'$;
4) существуют такие $b_1,b_2 \in \mathbb{F}$, что $b_1 \wedge b_2 = b$, причем $a \xrightarrow{R} b_1$, $a \xrightarrow{R} b_2$;
5) существуют такие $a_1,a_2 \in \mathbb{F}$, что $a_1 \vee a_2 = a$, причем $a_1 \xrightarrow{R} b$, $a_2 \xrightarrow{R} b$;

6) существует элемент $c \in \mathbb{F}$ такой, что $a \xrightarrow{R} c$ и $c \xrightarrow{R} b$.

Определение 1 по данному R задает новое отношение $\xrightarrow{R}$ на решетке $\mathbb{F}$, которое содержит отношения R, $\leq$, а также обладает некоторыми дополнительными свойствами. Условия 1)–6) определения 1 будем также называть правилами вывода.

При получении некоторой логической связи шагом вывода будем называть применение ровно одного правила, возможно, одновременно к некоторому конечному множеству элементов решетки. Например:

- если $b_t \xrightarrow{R} a_t$, $t \in T$, то $a_t' \xrightarrow{R} b_t'$, $t \in T$;
- если $a_t \xrightarrow{R} c_t$, $c_t \xrightarrow{R} b_t$, $t \in T$, то $a_t \xrightarrow{R} b_t$, $t \in T$.

Уровнем рекурсии в отношении $a \xrightarrow{R} b$ будем называть количество шагов вывода, необходимое для получения этой связи. При этом учитываются лишь применения правил 3)–6). Для связи, основанной только на правиле 1) или 2), уровень рекурсии считается равным нулю. Поскольку в общем случае связь $a \xrightarrow{R} b$ может быть получена не единственным набором правил, обычно будем лишь оценивать ее уровень рекурсии, не указывая его точного значения.

Применительно к шагам вывода соотношения $a \xrightarrow{R} b$ будем употреблять слова «начальный», «последний», а также «предыдущий», «следующий» и так далее. При этом имеется в виду продвижение в направлении прямого логического вывода, то есть от пар исходного отношения ($R \cup \leq$) к требуемой паре отношения $\xrightarrow{R}$. Заметим, что рекурсивное определение 1 сформулировано в рамках методологии обратного вывода.

Лемма 1. Пусть R – логическое отношение на булевой решетке $\mathbb{F}$ и $a, b \in \mathbb{F}$. Тогда, если справедливо $a \xrightarrow{R} b$, то имеет место $(a, b) \in R$.

Доказательство. Проведем его с помощью индукции по m – оценке уровня рекурсии в соотношении $a \xrightarrow{R} b$. При $m = 0$ имеет место одно из условий 1)–2) определения 1. Случай 1) непосредственно означает справедливость доказываемого утверждения. Если же выполнено 2), то и в этом случае $(a, b) \in R$, поскольку логическое отношение R содержит и $\leq$.

Предположим далее, что лемма верна для некоторого $m \geq 0$, и докажем ее утверждение при уровне рекурсии $m + 1$. В этом случае новые для рассмотрения варианты могут дать правила 3)–6).

Если имеет место 3), то по предположению индукции уровень рекурсии в соответствующем соотношении $b' \xrightarrow{R} a'$ не превосходит m, поэтому $(b', a') \in R$. Тогда, в силу контрапозиционности логического отношения R, получим $(a, b) \in R$.

Рассмотрим вариант, когда соотношение $a \xrightarrow{R} b$ происходит из условия 4) определения 1. При этом по предположению индукции соотношения $a \xrightarrow{R} b_1, a \xrightarrow{R} b_2$ имеют уровень рекурсии $\leq m$, то есть $(a,b_1),(a,b_2) \in R$. Тогда, в силу $\wedge$-дистрибутивности R, получим $(a,b) \in R$. Если же выполнено 5), то аналогичный факт $(a,b) \in R$ следует из свойства $\vee$-дистрибутивности отношения R.

Наконец, пусть справедливо 6). И в этом случае по предположению индукции базовые соотношения $a \xrightarrow{R} c$ и $c \xrightarrow{R} b$ имеют уровень рекурсии $\leq m$. Следовательно, $(a,c),(c,b) \in R$. Отсюда, в силу транзитивности логического отношения R, имеем $(a,b) \in R$. □

Теорема 1. Для произвольного отношения R на булевой решетке $\mathbb{F}$ логическое замыкание существует и совпадает с множеством $\xrightarrow{R}$ всех упорядоченных пар, логически связанных отношением R.

Доказательство. Заметим вначале, что при произвольном R соответствующее отношение $\xrightarrow{R}$ является логическим. Действительно, в силу условия 2) определения 1 оно содержит $\leq$, из 3) следует его контрапозиционность, из 4)–5) – дистрибутивность, а условие 6) означает транзитивность данного отношения. Далее, в силу условия 1) определения 1, отношение $\xrightarrow{R}$ содержит R. Для доказательства теоремы осталось показать, что это – наименьшее из отношений, обладающих этим свойством.

Пусть R' – любое другое логическое отношение, содержащее R. Тогда очевидно, что если $a \xrightarrow{R} b$, то $a \xrightarrow{R'} b$. Отсюда по лемме 1 имеем $(a,b) \in R'$. Следовательно, отношение $\xrightarrow{R}$ содержится в произвольно выбранном R', и, таким образом, является наименьшим логическим отношением, содержащим R. □

Следствие 1. Пусть R – бинарное отношение на булевой решетке $\mathbb{F}$ и $a_t \xrightarrow{R} b_t$, $\forall t \in T$. Тогда отношение $R' = R \bigcup \{(a_t,b_t) \mid t \in T\}$ эквивалентно R.

Доказательство. Заметим вначале, что по определению любое логическое отношение является собственным логическим замыканием. Такое же замечание в силу теоремы 1 относится и к отношению $\xrightarrow{R}$. Далее, из определения 1 очевидным образом следует, что для любых $R_1 \subseteq R_2$ справедливо $\xrightarrow{R_1} \subseteq \xrightarrow{R_2}$. В нашем случае по построению отношения R' выполнены включения $R \subseteq R' \subseteq \xrightarrow{R}$. Переходя к логическим замыканиям и учитывая сказанное выше, получаем, что отношения R и R' имеют общее логическое замыкание $\xrightarrow{R}$. □

Далее рассмотрим вопросы эквивалентных преобразований рассматриваемых в данном разделе LP-структур нулевого порядка.

Пусть дано произвольное бинарное отношение R на решетке $\mathbb{F}$. Его эквивалентным преобразованием называется такая замена всего множества упорядоченных пар R или его части, что полученное в результате новое отношение P логически эквивалентно R, то есть $P \sim R$.

Теорема 2. Пусть R_1, R_2, R_3, R_4 – отношения на общей булевой решетке $\mathbb{F}$. Если при этом $R_1 \sim R_2$ и $R_3 \sim R_4$, то $R_1 \bigcup R_3 \sim R_2 \bigcup R_4$.

Доказательство. Необходимо доказать равенство двух множеств $\xrightarrow{R_1 \cup R_3} = \xrightarrow{R_2 \cup R_4}$. Пусть вначале $a \xrightarrow{R_1 \cup R_3} b$. Докажем, что при этом справедливо $a \xrightarrow{R_2 \cup R_4} b$. Для этого, как и выше, применим метод индукции по m – оценке уровня рекурсии в соотношении $a \xrightarrow{R_1 \cup R_3} b$.

При $m = 0$ имеет место одно из условий 1)–2) определения 1. Если справедливо 2), то в этом случае $a \xrightarrow{R_2 \cup R_4} b$, поскольку логическое отношение $\xrightarrow{R_2 \cup R_4}$ содержит и $\le$. Если выполнено условие 1), то имеем $(a,b) \in R_1 \bigcup R_3$. Пусть для определенности $(a,b) \in R_1$ (вариант $(a,b) \in R_3$ симметричен). В этом случае имеет место $a \xrightarrow{R_1} b$, откуда в силу условия эквивалентности $R_1 \sim R_2$ получим $a \xrightarrow{R_2} b$. Следовательно, справедливо и $a \xrightarrow{R_2 \cup R_4} b$. Таким образом, при $m = 0$ утверждение теоремы доказано.

Предположим далее, что оно верно для некоторого $m \ge 0$, и докажем его при уровне рекурсии $m+1$. В этом случае новые для рассмотрения варианты для связи $a \xrightarrow{R_1 \cup R_3} b$ могут дать условия 3)–6) определения 1.

Если имеет место 3), то по предположению индукции уровень рекурсии в соответствующем соотношении $b' \xrightarrow{R_1 \cup R_3} a'$ не превосходит m, поэтому $b' \xrightarrow{R_2 \cup R_4} a'$. В этом случае, в силу контрапозиционности логического отношения $\xrightarrow{R_2 \cup R_4}$, получим $a \xrightarrow{R_2 \cup R_4} b$.

Рассмотрим вариант, когда соотношение $a \xrightarrow{R_1 \cup R_3} b$ происходит из условия 4) определения 1. Тогда по предположению индукции соотношения $a \xrightarrow{R_1 \cup R_3} b_1, a \xrightarrow{R_1 \cup R_3} b_2$ имеют уровень рекурсии $\le m$, поэтому $a \xrightarrow{R_2 \cup R_4} b_1, a \xrightarrow{R_2 \cup R_4} b_2$. Далее, в силу $\wedge$-дистрибутивности $\xrightarrow{R_{21} \cup R_4}$, получим $a \xrightarrow{R_2 \cup R_4} b$. Если же выполнено 5), то подлежащий доказательству факт аналогично следует из предположения индукции и свойства $\vee$-дистрибутивности логического отношения $\xrightarrow{R_{21} \cup R_4}$.

Наконец, пусть справедливо 6). И в этом случае базовые соотношения $a \xrightarrow{R_1 \cup R_3} c$ и $c \xrightarrow{R_1 \cup R_3} b$ имеют уровень рекурсии $\le m$. Следовательно, по предположению индукции получаем $a \xrightarrow{R_2 \cup R_4} c$ и $c \xrightarrow{R_2 \cup R_4} b$. Отсюда, в силу транзитивности логического отношения $\xrightarrow{R_{21} \cup R_4}$, имеем $a \xrightarrow{R_2 \cup R_4} b$. □

Следствие 2. Пусть R_1, R_2, R – отношения на общей булевой решетке $\mathbb{F}$. Если при этом $R_1 \sim R_2$, то $R_1 \cup R \sim R_2 \cup R$.

Следствия 1 и 2 обосновывают принцип локальности эквивалентных преобразований логических отношений.

4.1.2. Структура логических связей

В предыдущих главах, посвященных исследованию свойств LP-структур с более простой логикой, было показано, что логическое замыкание произвольного отношения R совпадает с транзитивным замыканием некоторого другого отношения $\tilde{R} \supseteq R$, построенного по данному R в виде «дистрибутивного многообразия». Это обстоятельство позволяет свести изучение некоторых важных вопросов, касающихся логических отношений, к соответствующим задачам транзитивных отношений. В частности, построение логического замыкания или редукции можно выполнить с помощью быстрых алгоритмов (типа Уоршолла), разработанных для транзитивных отношений [2]. В рассматриваемом случае также удается разделить процесс построения логического замыкания на два этапа, второй из которых соответствует транзитивному замыканию.

С целью получения указанного результата докажем предварительно некоторые вспомогательные утверждения, касающиеся устройства логических связей.

Лемма 1. Пусть R – отношение на булевой решетке и $a_t \xrightarrow{R} b_t,\ \forall t \in T$ (T –конечное множество). Тогда справедливы логические связи $\bigwedge\limits_{t\in T} a_t \xrightarrow{R} \bigwedge\limits_{t\in T} b_t$ и $\bigvee\limits_{t\in T} a_t \xrightarrow{R} \bigvee\limits_{t\in T} b_t$.

Доказательство. Чтобы установить первое из доказываемых соотношений, введем обозначения $\tilde{a} = \bigwedge\limits_{t\in T} a_t,\ \tilde{b} = \bigwedge\limits_{t\in T} b_t$. Поскольку $\tilde{a} \le a_t,\ \forall t \in T$, то в силу $a_t \xrightarrow{R} b_t$ и условия 6) определения 4.1.1.1[1] справедливо $\tilde{a} \xrightarrow{R} b_t,\ \forall t \in T$. Применяя к последнему соотношению правило 4), получим $\tilde{a} \xrightarrow{R} \tilde{b}$.

Для доказательства второй части леммы обозначим $\tilde{a} = \bigvee\limits_{t\in T} a_t$, $\tilde{b} = \bigvee\limits_{t\in T} b_t$. Далее, пользуясь неравенствами $b_t \le \tilde{b},\ \forall t \in T$, из $a_t \xrightarrow{R} b_t$ с помощью транзитивного условия 6) имеем $a_t \xrightarrow{R} \tilde{b},\ \forall t \in T$. Наконец, используя правило 5), приходим к соотношению $\tilde{a} \xrightarrow{R} \tilde{b}$. □

Следствие 1. На основе леммы 1 можно ввести обобщенные правила вывода, соответствующие правилам 4) и 5) определения 4.1.1.1. Упорядоченная пара $a, b \in \mathbb{F}$ логически связана

[1] Определение 1 из п.4.1.1.

отношением R ($a \xrightarrow{R} b$), если выполнено одно из следующих условий:

4′) существуют такие $a_t, b_t \in \mathbb{F}, t \in T$, что $\bigwedge_{t\in T} a_t = a, \bigwedge_{t\in T} b_t = b$, причем $a_t \xrightarrow{R} b_t, \forall t \in T$;

5′) существуют такие $a_t, b_t \in \mathbb{F}, t \in T$, что $\bigvee_{t\in T} a_t = a, \bigvee_{t\in T} b_t = b$, причем $a_t \xrightarrow{R} b_t, \forall t \in T$.

Очевидно, что правила 4) и 5) определения 4.1.1.1 являются частными случаями правил 4′) и 5′).

Лемма 2. Пусть R – отношение на булевой решетке. Тогда при выводе произвольной логической связи $a \xrightarrow{R} b$ все применения дистрибутивных правил 4′)–5′) могут быть произведены без участия правила 2) со строгим неравенством. Роль последнего можно свести исключительно к присутствию в качестве компонента для транзитивного правила 6).

Доказательство. Предположим, что при получении вывода $a \xrightarrow{R} b$ использовалось условие 4′). Разобьем имеющееся множество пар $\{(a_t, b_t) \mid t \in T\}$ на 2 подмножества. В первое из них войдут такие пары (a_t, b_t), для которых выполнено $a_t < b_t$. Соответствующее им множество индексов $\{t\}$ обозначим T_1. Ко второму подмножеству отнесем все остальные пары $(a_t, b_t), t \in T_2$. Тогда, вводя обозначения $\bigwedge_{t\in T_1} a_t = a^1, \bigwedge_{t\in T_1} b_t = b^1$, $\bigwedge_{t\in T_2} a_t = a^2, \bigwedge_{t\in T_2} b_t = b^2$, получим $a = a^1 \wedge a^2$, $b = b^1 \wedge b^2$. При этом $a^1 \le b^1$ (соответственно $a^1 \xrightarrow{R} b^1$) и по условию 4′) имеет место $a^2 \xrightarrow{R} b^2$. Очевидно, что указанное выше применение правила 4′) для $\{(a_t, b_t) \mid t \in T\}$ можно заменить аналогичным правилом для $(a^1, b^1), (a^2, b^2)$.

Если при этом окажется, что $T_1 = \varnothing$, то для данного вывода $a \xrightarrow{R} b$ сформулированное в лемме утверждение относительно правила 4′) выполнено сразу. Если же $T_2 = \varnothing$ либо $a^2 \le b^2$, то применение правила 4′) не потребуется вовсе. Рассмотрим оставшийся нетривиальный случай ($T_1 \ne \varnothing, T_2 \ne \varnothing$ и не имеет места $a^2 \le b^2$), в котором поступим следующим образом. Применим правило 4′) к соотношениям $b^1 \xrightarrow{R} b^1$, $a^2 \xrightarrow{R} b^2$, тогда получим $b^1 \wedge a^2 \xrightarrow{R} b^1 \wedge b^2$. Возвращаясь к имеющемуся неравенству $a^1 \le b^1$, заметим, что в силу свойств решетки $\mathbb{F}$ из него следует неравенство $a^1 \wedge a^2 \le b^1 \wedge a^2$. По этой причине, применяя к парам $a^1 \wedge a^2 \le b^1 \wedge a^2$ и $b^1 \wedge a^2 \xrightarrow{R} b^1 \wedge b^2$ транзитивное правило 6), приходим к соотношению $a^1 \wedge a^2 \xrightarrow{R} b^1 \wedge b^2$, т.е. $a \xrightarrow{R} b$. При этом мы исключили в применяемом правиле 4′) участие строгих неравенств вида $a_t < b_t$. Таким образом, для правила 4′) утверждение леммы доказано.

Рассмотрим другой вариант, когда при выводе $a \xrightarrow{R} b$ было применено правило 5′). Аналогично способу, примененному ранее, разобьем имеющееся множество $\{(a_t, b_t) \mid t \in T\}$ на 2 подмножества. Первое из них будет содержать пары $a_t < b_t$, соответствующее множество индексов обозначим T_1. Ко второму подмножеству отнесем все остальные пары (a_t, b_t), $t \in T_2$. Тогда, обозначая $\bigvee\limits_{t \in T_1} a_t = a^1$, $\bigvee\limits_{t \in T_1} b_t = b^1$, $\bigvee\limits_{t \in T_2} a_t = a^2$, $\bigvee\limits_{t \in T_2} b_t = b^2$, будем иметь $a = a^1 \vee a^2$, $b = b^1 \vee b^2$. При этом $a^1 \le b^1$ (соответственно $a^1 \xrightarrow{R} b^1$) и по правилу 5′) имеет место $a^2 \xrightarrow{R} b^2$. Указанное выше применение правила 5′) для $\{(a_t, b_t) \mid t \in T\}$ можно заменить аналогичным правилом для $(a^1, b^1), (a^2, b^2)$.

Если выяснится, что $T_1 = \varnothing$, то для вывода $a \xrightarrow{R} b$ сформулированное в лемме утверждение относительно правила 5′) будет выполнено сразу. Если же $T_2 = \varnothing$ либо $a^2 \le b^2$, то применение правила 5) вообще не потребуется. Остается нетривиальный случай ($T_1 \ne \varnothing, T_2 \ne \varnothing$ и не имеет места $a^2 \le b^2$), в котором применим к соотношениям $a^1 \xrightarrow{R} a^1, a^2 \xrightarrow{R} b^2$ правило 5′), соответственно получим $a^1 \vee a^2 \xrightarrow{R} a^1 \vee b^2$. Возвращаясь к имеющемуся неравенству $a^1 \le b^1$, заметим, что в силу свойств решетки $\mathbb{F}$ из него следует неравенство $a^1 \vee b^2 \le b^1 \vee b^2$. Применяя к соотношениям $a^1 \vee a^2 \xrightarrow{R} a^1 \vee b^2$ и $a^1 \vee b^2 \le b^1 \vee b^2$ транзитивное правило 6), получим справедливость $a^1 \vee a^2 \xrightarrow{R} b^1 \vee b^2$, то есть $a \xrightarrow{R} b$. Теперь утверждение леммы доказано и для случая применения правила 5′). □

Лемма 3. Пусть R – отношение на булевой решетке $\mathbb{F}$. Тогда при выводе любой логической связи $a \xrightarrow{R} b$ все применения контрапозиционного правила 3) определения 4.1.1.1 могут быть исключены либо перенесены в начальную стадию этого процесса.

Доказательство. Достаточно показать, что любое применение правила 3) можно исключить либо поменять местами с предшествующим ему применением каждого из правил 2), 4′), 5′), 6), получив при этом аналогичную логическую связь. Рассмотрим все возможные в данной ситуации случаи.

Предположим, что вывод $a \xrightarrow{R} b$ получен из соотношения $b' \le a'$ применением правила 3). Поскольку $b' \le a'$, то в силу свойств булевой решетки справедливо $a \le b$, откуда сразу следует $a \xrightarrow{R} b$. Таким образом, в данном случае применение правила 3) можно исключить.

Рассмотрим вариант, когда при получении некоторого вывода $a \xrightarrow{R} b$ непосредственно перед правилом 3) использовалось условие 4′). В этом случае существуют такие $a_t, b_t \in \mathbb{F}$, $t \in T$, что $\bigwedge_{t\in T} a_t = a'$, $\bigwedge_{t\in T} b_t = b'$, причем $b_t \xrightarrow{R} a_t$, $\forall t \in T$. Вместо указанного последовательного применения правил 4′) и 3) поступим по-другому – сразу применим к имеющимся соотношениям $b_t \xrightarrow{R} a_t$, $\forall t \in T$ правило 3). Тогда получим, что справедливо $a_t' \xrightarrow{R} b_t'$, $\forall t \in T$. Применяя далее правило 5′), приходим к соотношению $\bigvee_{t\in T} a_t' \xrightarrow{R} \bigvee_{t\in T} b_t'$, откуда с помощью закона де Моргана получим $a \xrightarrow{R} b$. Таким образом, в данном случае применение правила 3) удается переместить на шаг назад, получив при этом аналогичный результат $a \xrightarrow{R} b$.

Рассмотрим ситуацию, в которой при получении вывода $a \xrightarrow{R} b$ перед правилом 3) было применено условие 5′). В данной ситуации существуют такие $a_t, b_t \in \mathbb{F}$, $t \in T$, что $\bigvee_{t\in T} a_t = a'$, $\bigvee_{t\in T} b_t = b'$, причем $b_t \xrightarrow{R} a_t$, $\forall t \in T$. В этом случае вместо правил 5′) и 3) применим к имеющимся соотношениям $b_t \xrightarrow{R} a_t$, $\forall t \in T$ правило 3). В результате получим связи $a_t' \xrightarrow{R} b_t'$, $\forall t \in T$. Применяя теперь правило 4′), приходим к соотношению $\bigwedge_{t\in T} a_t' \xrightarrow{R} \bigwedge_{t\in T} b_t'$, откуда по закону де Моргана получим $a \xrightarrow{R} b$. И в этом случае применение правила 3) удалось переместить к началу вывода с сохранением результата $a \xrightarrow{R} b$.

Наконец, остается случай, когда при получении $a \xrightarrow{R} b$ перед правилом 3) было применено транзитивное правило 6). В этом случае существует элемент $c \in \mathbb{F}$ такой, что $b' \xrightarrow{R} c$ и $c \xrightarrow{R} a'$. Как и выше, к этим базовым соотношениям сразу применим правило 3). В результате получим, что $c' \xrightarrow{R} b$, $a \xrightarrow{R} c'$. Отсюда по правилу 6) приходим к соотношению $a \xrightarrow{R} b$.

Таким образом, в каждой возможной ситуации вывода применение правила 3) определения 4.1.1.1 можно произвести до применений остальных его правил. □

Лемма 4. Пусть R – отношение на булевой решетке. Тогда при выводе любой логической связи $a \xrightarrow{R} b$ все применения транзитивного правила 6) могут быть исключены либо перенесены в заключительную стадию данного процесса.

Доказательство. Докажем утверждение леммы с помощью индукции по m – уровню рекурсии в соотношении $a \xrightarrow{R} b$. При $m = 0$ для a, b имеет место одно из условий 1)–2) определения 4.1.1.1[1]. В этом случае вывод $a \xrightarrow{R} b$ вовсе не содержит транзитивных связей, а значит, утверждение леммы выполнено.

[1] Определение 1 из п. 4.1.1.

Предположим далее, что оно верно для некоторого $m \geq 0$, и докажем это утверждение при уровне рекурсии $m+1$. По предположению индукции, первые m шагов вывода можно организовать так, что транзитивное правило 6) будет применяться лишь в конце процесса. В этом случае все зависит от того, какое правило вывода применено на последнем ($m+1$) шаге. По лемме 2 правило 2) в этом плане можно не рассматривать, а по лемме 3 все применения контрапозиционного правила 3) можно считать реализованными в начале процесса вывода. Если же последним применено правило 6), то утверждение леммы сразу оказывается выполненным – все транзитивные связи использованы в заключительной стадии вывода. Таким образом, остается исследовать случаи, когда на ($m+1$) шаге применено правило вывода 4′) либо 5′).

Рассмотрим вариант, когда связь $a \xrightarrow{R} b$ в конечном счете получена из условия 4′). При этом каждое базовое соотношение $a_t \xrightarrow{R} b_t,\ t \in T$ имеет уровень рекурсии $\leq m$ и, по предположению индукции, все свои транзитивные связи использует лишь в конце вывода. Другими словами, при каждом $t \in T$ существует цепочка элементов $a_t = c_t^0, c_t^1, \ldots, c_t^{n_t} = b_t$ такая, что выполнены соотношения $c_t^{i-1} \xrightarrow{R} c_t^i$, $i = 1, \ldots, n_t$, при выводе которых не используется транзитивное правило 6). Ввиду конечности множества T, величины n_t ограничены в совокупности, то есть $n_t \leq N,\ t \in T.$ В связи с этим обстоятельством, будем считать все указанные выше цепочки элементов равными по длине N, дополнив каждую из них в случае необходимости справа повторяющимся элементом $c_t^{n_t} = b_t$. Тогда для цепочек $a_t = c_t^0, c_t^1, \ldots, c_t^N = b_t,\ t \in T$ построим элементы $\tilde{c}_i = \bigwedge_{t \in T} c_t^i,\ i = 0, \ldots, N$. Очевидно, что $\tilde{c}_0 = a$ и $\tilde{c}_N = b$. Кроме того, поскольку $c_t^{i-1} \xrightarrow{R} c_t^i$, то к этим парам можно применить правило 4′), то есть получаем $\tilde{c}_{i-1} \xrightarrow{R} \tilde{c}_i$, $i = 1, \ldots, N$.

Таким образом, удалось показать, что в рассматриваемом случае все применения транзитивного правила 6) в базовых соотношениях $a_t \xrightarrow{R} b_t,\ t \in T$ можно заменить его применением на заключительной стадии вывода к построенным элементам $\tilde{c}_i,\ i = 0, \ldots, N$.

Предположим теперь, что рассматриваемая связь $a \xrightarrow{R} b$ была выведена из условия 5′). Тогда все базовые соотношения $a_t \xrightarrow{R} b_t,\ t \in T$ имеют уровень рекурсии $\leq m$ и, по предположению индукции, все транзитивные связи использует только в конце вывода. По этой причине при каждом $t \in T$ существует цепочка элементов $a_t = c_t^0, c_t^1, \ldots, c_t^{n_t} = b_t$ такая, что выполнены соотношения $c_t^{i-1} \xrightarrow{R} c_t^i$, $i = 1, \ldots, n_t$, при выводе которых не используется транзитивное правило 6). Поскольку множество T конечно, то все величины n_t ограничены в совокупности, а именно – $n_t \le N$, $t \in T$. Как и выше, будем считать все цепочки элементов равными по длине N, расширив их в случае необходимости справа последними элементами $c_t^{n_t} = b_t$. Тогда для цепочек $a_t = c_t^0, c_t^1, \dots, c_t^N = b_t$, $t \in T$ построим элементы $\tilde{c}_i = \bigvee_{t \in T} c_t^i$, $i = 0, \dots, N$. Легко видеть, что $\tilde{c}_0 = a$ и $\tilde{c}_N = b$. Кроме того, поскольку $c_t^{i-1} \xrightarrow{R} c_t^i$, то к этим парам можно применить правило 5′), то есть получаем $\tilde{c}_{i-1} \xrightarrow{R} \tilde{c}_i$, $i = 1, \dots, N$.

Итак, в рассматриваемом случае все применения транзитивного правила 6) в базовых соотношениях $a_t \xrightarrow{R} b_t$, $t \in T$ также можно заменить его применением на заключительной стадии вывода к построенным элементам $\tilde{c}_i$, $i = 0, \dots, N$. □

4.1.3. Логическая редукция

Для произвольного отношения R на булевой решетке $\mathbb{F}$ введем отношение $\tilde{R}$, построенное по данному R последовательным выполнением следующих действий (шагов):

1) добавить к R все пары вида (a, a), где $a \in \mathbb{F}$ (рефлексивные пары), и обозначить новое отношение R_1;

2) добавить к R_1 все пары (a, b), для которых $(b', a') \in R$, и обозначить новое отношение R_2;

3) добавить к R_2 всевозможные пары вида $(a_1 o_1 a_2 o_2 \dots o_{m-1} a_m, b_1 o_1 b_2 o_2 \dots o_{m-1} b_m)$, где $(a_i, b_i) \in R_2$, символ o_i обозначает операцию $\wedge$ либо $\vee$, и обозначить новое отношение R_3;

4) объединить полученное отношение с отношением $\le$.

Заметим, что в силу дистрибутивности булевой решетки, шаг 3) можно заменить двумя более простыми последовательными шагами, а именно – вначале использовать лишь одну из операций $\wedge$ либо $\vee$, затем – только вторую.

Замечание 1. Применяя конечное число раз лемму 4.1.2.1[1] и следствие 4.1.1.1[2], нетрудно убедиться в том, что отношение $\tilde{R}$ логически эквивалентно R.

Лемма 1. Пусть R – отношение на булевой решетке $\mathbb{F}$. Тогда, если $a \xrightarrow{R} b$, и это соотношение может быть получено без использования транзитивного правила 6) определения 4.1.1.1, то $(a, b) \in \tilde{R}$.

Доказательство. Пусть имеет место $a \xrightarrow{R} b$, полученное без правила 6). Если указанное соотношение произошло непосредственно из условия 1) или 2) определения 4.1.1.1, то сразу имеем

[1] Лемма 1 из п.4.1.2.

[2] Следствие 1 из п.4.1.1.

$(a,b) \in \tilde{R}$. Остается рассмотреть нетривиальный случай применения цепочки правил 2), 3), 4′), 5′).

Поскольку при выводе правило 6) не используется, то по лемме 4.1.2.2 не применяется и правило 2) со строгим неравенством. Далее, по лемме 4.1.2.3 все применения контрапозиционного правила 3) могут быть осуществлены на начальном этапе процесса вывода. Следовательно, правила 4′)–5′) завершают этот процесс.

Если сопоставить перечисленные этапы вывода с последовательностью построения отношения $\tilde{R}$, то окажется, что вывод $a \xrightarrow{R} b$ представляет собой построение некоторого подмножества $\tilde{R}$, что и доказывает включение $(a,b) \in \tilde{R}$. □

Теорема 1. Логическое замыкание отношения R на булевой решетке $\mathbb{F}$ совпадает с транзитивным замыканием $\tilde{R}^*$ соответствующего отношения $\tilde{R}$.

Доказательство. Как было отмечено выше, отношение $\tilde{R}$ эквивалентно R. Следовательно, по определению логического замыкания, имеем $\tilde{R} \subseteq \xrightarrow{R}$. Отсюда, поскольку отношение $\xrightarrow{R}$ транзитивно, получаем $\tilde{R}^* \subseteq \xrightarrow{R}$.

Докажем обратное включение. Предположим, что $a \xrightarrow{R} b$. По лемме 4.1.2.4, при получении этого вывода все применения транзитивного правила 6) определения 4.1.1.1 могут быть перенесены в заключительную стадию процесса. Этот факт означает следующее: существует цепочка элементов $a = c_0, c_1, \ldots, c_n = b$ такая, что выполнены соотношения $c_{i-1} \xrightarrow{R} c_i$, $i = 1, \ldots, n$, при выводе которых правило 6) не используется. Тогда по лемме 1 имеем $(c_{i-1}, c_i) \in \tilde{R}$, откуда сразу получаем $(a,b) \in \tilde{R}^*$. Итак, $\xrightarrow{R} \subseteq \tilde{R}^*$. □

Рассмотрим непосредственно вопрос о существовании и построении логической редукции LP-структур.

Для отношения R на булевой решетке $\mathbb{F}$ введем отношение $\underset{\sim}{R}$, построенное по данному R последовательным выполнением действий (шагов), обратных построению $\tilde{R}$, а именно:

1) исключить из R все пары (a,b), для которых $a < b$, и обозначить новое отношение R_{-1}.

2) исключить из R_{-1} все пары (a,b) вида $(a_1 o_1 a_2 o_2 \ldots o_{m-1} a_m, b_1 o_1 b_2 o_2 \ldots o_{m-1} b_m)$, где $(a_i, b_i) \in R_{-1}$ либо $(b_i', a_i') \in R_{-1}$, символ o_i обозначает операцию $\wedge$ либо $\vee$, причем (a,b) не совпадает ни с одной парой (a_i, b_i), и обозначить новое отношение R_{-2};

3) исключить из R_{-2} все контрапозиционные пары, то есть при $(a,b), (b', a') \in R_{-2}$ оставить в R_{-2} лишь одну из двух пар, и обозначить новое отношение R_{-3};

4) исключить из R_{-3} все пары вида (a,a), где $a \in \mathbb{F}$ (рефлексивные пары).

Как и выше для отношения $\tilde{R}$, в силу дистрибутивности булевой решетки, шаг 2) может быть заменен двумя более простыми последовательными шагами: вначале использовать лишь одну из операций $\wedge$ либо $\vee$, затем – только вторую.

Замечание 2. Так же с помощью леммы 4.1.2.1 и следствия 4.1.1.1 нетрудно убедиться в том, что отношение $\utilde{R}$ логически эквивалентно R.

Лемма 2. Пусть R – бинарное отношение на булевой решетке $\mathbb{F}$. Для того, чтобы R являлось логической редукцией, необходимо и достаточно, чтобы R не содержало ни одной такой пары (a,b), для которой выполнено соотношение $a \xrightarrow{R \setminus \{(a,b)\}} b$.

Доказательство. Пусть отношение R представляет собой логическую редукцию, то есть является минимальным логически эквивалентным себе отношением. Предположим противное, а именно, что существует пара $(a,b) \in R$, логически связанная отношением $R \setminus \{(a,b)\}$. Если это так, то в силу следствия 4.1.1.1 пару (a,b) можно исключить из R, получив при этом меньшее эквивалентное отношение. Таким образом, при сделанном предположении отношение R не может быть логической редукцией.

Для доказательства обратного утверждения предположим, что не существует ни одной пары $(a,b) \in R$, для которой справедливо $a \xrightarrow{R \setminus \{(a,b)\}} b$. Необходимо доказать, что в этом случае R есть логическая редукция. Вновь предположим противное – пусть существует отношение $R_0 \subset R$, эквивалентное R, и $(a,b) \in R \setminus R_0$. Тогда, поскольку $(a,b) \in R$, в силу эквивалентности рассматриваемых отношений справедливо $a \xrightarrow{R_0} b$. Так как отношение R_0 не содержит пару (a,b), то $R_0 \subseteq R \setminus \{(a,b)\}$, и логическая связь $a \xrightarrow{R_0} b$ противоречит сделанному ранее предположению о том, что таких пар (a,b) в R нет. Противоречие доказывает утверждение леммы. □

Следующая теорема указывает достаточное условие существования и способ построения логической редукции данного отношения.

Теорема 2. Пусть для отношения R на булевой решетке $\mathbb{F}$ построено соответствующее отношение $\tilde{R}$ (см. выше). Тогда, если для $\tilde{R}$ существует транзитивная редукция R^0, то соответствующее ей отношение $\utilde{R}^0$ представляет собой логическую редукцию исходного отношения R.

Доказательство. Из замечаний 1–2 следует, что указанное в теореме отношение $\utilde{R}^0$ логически эквивалентно R. Осталось показать, что $\underset{\sim}{R}^0$ является логической редукцией вообще. Для этого достаточно проверить выполнение для $\underset{\sim}{R}^0$ условия леммы 2.

Пусть (a,b) – произвольная пара отношения $\underset{\sim}{R}^0$. По условию теоремы необходимо показать невозможность логической связи $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$. Предположим противное, а именно, что эта связь существует. Тогда в силу следствия 4.1.1.1 отношение $\underset{\sim}{R}^0 \setminus \{(a,b)\}$ эквивалентно $\underset{\sim}{R}^0$. Сразу заметим, что применение правила 1) определения 4.1.1.1 для вывода $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$ невозможно, поскольку пара (a,b) не содержится в множестве $\underset{\sim}{R}^0 \setminus \{(a,b)\}$.

По лемме 4.1.2.4 любая логическая связь может быть построена таким образом, что все транзитивности будут использоваться лишь в завершающей стадии ее вывода. Этот факт означает, что существует цепочка элементов $a = c_0, c_1, \ldots, c_n = b$ такая, что выполнены соотношения $c_{i-1} \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} c_i$, $i = 1, \ldots, n$, при выводе каждого из которых правило 6) не используется. Отсюда по лемме 1 имеем $(c_{i-1}, c_i) \in \tilde{R}$. Таким образом, при $n > 1$ пара (a,b) оказывается транзитивной в $\tilde{R}$. Следовательно, она не может содержаться в $\underset{\sim}{R}^0$, являющемся подмножеством транзитивной редукции отношения $\tilde{R}$. Таким образом, получено противоречие исходному предположению $(a,b) \in \underset{\sim}{R}^0$.

Остается исследовать случай $n = 1$. В этой ситуации по лемме 4.1.2.3[1] рассматриваемая логическая связь $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$ может содержать применения правил 4′) и 5′) лишь в заключительной стадии вывода. В силу свойства дистрибутивности булевой решетки, любая полученная таким образом пара (a,b) описывается шагами 3) и 4) процесса построения отношения $\tilde{R}$. Соответственно при нахождении $\underset{\sim}{R}^0$ (обратный процесс) она будет исключена. Таким образом, снова приходим к противоречию исходному предположению $(a,b) \in \underset{\sim}{R}^0$.

При выводе $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$ не могло использоваться и правило 3) определения 4.1.1.1, поскольку в этом случае при $(a,b) \in \underset{\sim}{R}^0$ соответствующая контрапозиционная пара (b', a') должна быть исключена на шаге 4) построения $\underset{\sim}{R}^0$. Применение правила 2) при $a \neq b$ исключается леммой 4.1.2.2, а случай $a = b$ невозможен ввиду выполнения шага 5) процесса построения отношения $\underset{\sim}{R}^0$.

Таким образом, удалось исследовать возможные варианты предполагаемого логического вывода $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$. В результате установлено, что в каждом таком случае наличие связи $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$

[1] Лемма 3 из п.4.1.2.

противоречит факту $(a,b) \in \underset{\sim}{R}^0$. Следовательно, отношение $\underset{\sim}{R}^0$ представляет собой логическую редукцию. □

В следующем разделе рассматривается класс LP-структур, соответствующий моделям продукционных систем с еще более сложной логикой. Такие системы могут содержать в предпосылках и заключениях своих правил предикаты первого порядка.

4.2. LP-структуры первого порядка

В данном разделе определяются расширенные LP-структуры на полных булевых решетках. Наряду с рассмотренным выше набором логических связок пропозиционального исчисления, в таких алгебраических системах определены также бесконечноместные операции пересечения и объединения, реализующие для модели продукционной логики кванторы общности и существования [187]. Отсюда происходит название такой алгебраической системы – «LP-структура первого порядка». Как и в главе 3, переход к полным решеткам приводит к необходимости введения дополнительных ограничений модели, связанных с аналогом свойства нетеровости (завершаемости вывода).

Изучен стандартный набор основных вопросов, в их числе существование LP-замыкания, эквивалентные преобразования, структура логических связей. Доказана теорема о существовании логической редукции произвольного отношения и описан процесс ее построения, один из этапов которого представляет собой нахождение транзитивной редукции бинарного отношения.

4.2.1. Логическое замыкание и эквивалентные преобразования

В данном подразделе будем рассматривать бинарные отношения на полной булевой решетке $\mathbb{F}$. Введем несколько основных понятий.

Бинарное отношение R на полной решетке называется:

• дистрибутивным относительно операции $\bigwedge$ (или $\bigwedge$-дистрибутивным), если для любого подмножества элементов решетки $\{b_t \mid t \in T\}$, такого, что $(a,b_t) \in R, \forall t \in T$, справедливо $(a, \bigwedge\limits_{t\in T} b_t) \in R$;

• дистрибутивным относительно операции $\bigvee$ (или $\bigvee$-дистрибутивным), если для любого подмножества элементов решетки $\{a_t \mid t \in T\}$ из $(a_t,b) \in R, \forall t \in T$ следует $(\bigvee\limits_{t\in T} a_t, b) \in R$;

• вполне дистрибутивным, если оно одновременно обладает обоими отмеченными выше свойствами.

Очевидно, что из полной дистрибутивности отношения следует его дистрибутивность в смысле, описанном в п.4.1.1.

Как и в главе 3, при определении продукционно-логического отношения на полной решетке потребуется формулировка дополнительного свойства, гарантирующего конечность длин транзитивных цепочек, реализующих логические связи элементов – аналога свойства нетеровости. Для того, чтобы сформулировать это свойство, вначале определим понятие логической связи.

Определение 1. Пусть задано бинарное отношение R на полной булевой решетке $\mathbb{F}$. Будем считать, что упорядоченная пара $a, b \in \mathbb{F}$ логически связана отношением R (обозначим этот факт $a \xrightarrow{R} b$), если выполнено одно из следующих условий:

1) $(a,b) \in R$;
2) $a \le b$;
3) $b' \xrightarrow{R} a'$;
4) существуют такие $b_t \in \mathbb{F},\ t \in T$, что $\bigwedge\limits_{t \in T} b_t = b$, причем $a \xrightarrow{R} b_t,\ \forall t \in T$;
5) существуют такие $a_t \in \mathbb{F},\ t \in T$, что $\bigwedge\limits_{t \in T} a_t = a$, причем $a_t \xrightarrow{R} b,\ \forall t \in T$;
6) существует элемент $c \in \mathbb{F}$ такой, что $a \xrightarrow{R} c$ и $c \xrightarrow{R} b$.

Определение 1 по данному R задает новое отношение $\xrightarrow{R}$ на той же решетке $\mathbb{F}$, которое содержит отношения R , $\le$, а также обладает некоторыми дополнительными свойствами. Условия 1)–6) определения 1 будем также называть правилами вывода.

Определение 2. Отношение R на полной булевой решетке $\mathbb{F}$ называется нетеровым, если любое соотношение $a \xrightarrow{R} b$ $(a, b \in \mathbb{F})$ может быть получено таким образом, что длины всех применяемых для этого транзитивных цепочек (правило 6) ограничены в совокупности.

При получении некоторой логической связи шагом вывода будем называть применение ровно одного правила, возможно, одновременно к бесконечному множеству элементов решетки. Например:

- если $b_t \xrightarrow{R} a_t,\ t \in T$, то $a'_t \xrightarrow{R} b'_t,\ t \in T$;
- если $a_t \xrightarrow{R} c_t,\ c_t \xrightarrow{R} b_t,\ t \in T$, то $a_t \xrightarrow{R} b_t,\ t \in T$.

Таким образом, определение 1 для нетерова отношения R представляет лишь такие логические связи, которые могут быть получены за конечное число шагов. Соответственно, уровнем рекурсии в соотношении $a \xrightarrow{R} b$ будем называть количество шагов вывода, необходимое для получения этой связи. При этом учитываются лишь применения правил 3)–6). Для связи, основанной только на правиле 1) или 2), уровень рекурсии считается равным нулю. Поскольку в общем случае данная логическая связь $a \xrightarrow{R} b$ может быть получена не единственным набором шагов, будем лишь оценивать ее уровень рекурсии, не указывая его точного значения.

Применительно к шагам вывода некоторого соотношения $a \xrightarrow{R} b$ будем употреблять слова «начальный», «последний», а также «предыдущий», «следующий» и так далее. При этом имеется в виду продвижение в направлении прямого логического вывода, то есть от пар исходного отношения ($R \bigcup \leq$) к требуемой паре отношения $\xrightarrow{R}$. В то же время рекурсивное определение 1 сформулировано в соответствии с методологией обратного вывода.

Отношение R на полной булевой решетке называется продукционно-логическим (в данном разделе – просто логическим), если оно нетерово, содержит $\leq$, а также является контрапозиционным, вполне дистрибутивным и транзитивным. Логическим замыканием нетерового отношения R на полной булевой решетке называется наименьшее логическое отношение, содержащее R. Как и везде в данной работе, под LP-структурой подразумевается решетка с заданным на ней логическим отношением.

Замечание 1. При заданном логическом отношении R на полной булевой решетке $\mathbb{F}$ для каждого элемента $c \in \mathbb{F}$ совокупность всех его R-прообразов Δ_c образует полный идеал в $\mathbb{F}$, а множество всех R-образов ∇_c представляет собой полный фильтр в $\mathbb{F}$.

Очевидно также, что отношение $\leq$ на полной булевой решетке само является логическим.

Как и ранее, два отношения R_1 и R_2, определенные на полной булевой решетке, называются логически эквивалентными (в контексте данного раздела – просто эквивалентными), если их логические замыкания совпадают. Этот факт обозначается $R_1 \sim R_2$. Логической редукцией данного отношения R называется минимальное эквивалентное ему отношение R_0.

Перейдем к выяснению вопроса о существовании логических отношений на полной булевой решетке.

Лемма 1. Пусть R – логическое отношение на полной булевой решетке $\mathbb{F}$ и $a,b \in \mathbb{F}$. Тогда, если справедливо $a \xrightarrow{R} b$, то $(a,b) \in R$.

Доказательство. Проведем его с помощью индукции по m – оценке уровня рекурсии в соотношении $a \xrightarrow{R} b$. При $m = 0$ имеет место одно из условий 1)–2) определения 1. Случай 1) означает требуемое утверждение. Если же справедливо 2), то и тогда $(a,b) \in R$, поскольку логическое отношение R содержит и $\leq$.

Предположим далее, что лемма верна для некоторого $m \geq 0$, и докажем ее утверждение при уровне рекурсии $m+1$. В этом случае новые для рассмотрения варианты могут дать условия 3)–6) определения 1.

Если в качестве последнего шага применяется правило 3), то по предположению индукции уровень рекурсии в соответствующем соотношении $b' \xrightarrow{R} a'$ не превосходит m, поэтому $(b', a') \in R$. Тогда, в силу контрапозиционности логического отношения R, получим $(a, b) \in R$.

Рассмотрим вариант, когда соотношение $a \xrightarrow{R} b$ происходит из условия 4) определения 1. При этом по предположению индукции каждое соотношение $a \xrightarrow{R} b_t$ имеет уровень рекурсии $\leq m$, то есть $(a, b_t) \in R$, $t \in T$. Тогда, в силу $\bigwedge$-дистрибутивности R, получим $(a, b) \in R$. Если же выполнено 5), то аналогичный факт $(a, b) \in R$ следует из свойства $\bigvee$-дистрибутивности отношения R.

Наконец, пусть справедливо 6). И в этом случае по предположению индукции базовые соотношения $a \xrightarrow{R} c$ и $c \xrightarrow{R} b$ имеют уровень рекурсии $\leq m$. Следовательно, $(a, c), (c, b) \in R$. Отсюда, в силу транзитивности логического отношения R, имеем $(a, b) \in R$. □

Теорема 1. Для нетерового отношения R на полной булевой решетке $\mathbb{F}$ логическое замыкание существует и совпадает с множеством $\xrightarrow{R}$ всех упорядоченных пар, логически связанных отношением R.

Доказательство. Заметим вначале, что при нетеровом R соответствующее отношение $\xrightarrow{R}$ является логическим. Действительно, в силу условия 2) определения 1 оно содержит $\leq$, из 3) следует его контрапозиционность, из 4)–5) – дистрибутивность, а 6) означает транзитивность данного отношения. Далее, в силу условия 1) определения 1, отношение $\xrightarrow{R}$ содержит R. Для доказательства теоремы осталось показать, что это – наименьшее из таковых отношений.

Пусть R' – любое другое логическое отношение, содержащее R. Тогда очевидно, что если $a \xrightarrow{R} b$, то $a \xrightarrow{R'} b$. Отсюда по лемме 1 имеем $(a, b) \in R'$. Следовательно, отношение $\xrightarrow{R}$ содержится в произвольно выбранном R', и, таким образом, является наименьшим логическим отношением, содержащим R. □

Следствие 1. Пусть R – нетерово отношение на полной булевой решетке $\mathbb{F}$ и $a_t \xrightarrow{R} b_t$, $\forall t \in T$. Тогда отношение $R' = R \bigcup \{(a_t, b_t) \mid t \in T\}$ логически эквивалентно R.

Доказательство аналогично следствию 4.1.1.1[1]. □

[1] Следствие 1 из п. 4.1.1.

Теперь рассмотрим вопросы, связанные с возможностью эквивалентных преобразований рассматриваемых LP-структур. Пусть дано нетерово бинарное отношение R на полной булевой решетке $\mathbb{F}$. Его эквивалентным преобразованием называется такая замена множества упорядоченных пар R (всего или некоторой его части), что полученное в результате новое отношение P логически эквивалентно R, то есть $P \sim R$.

Теорема 2. Пусть R_1, R_2, R_3, R_4 – нетеровы отношения на полной булевой решетке $\mathbb{F}$. Если при этом $R_1 \sim R_2$ и $R_3 \sim R_4$, то $R_1 \bigcup R_3 \sim R_2 \bigcup R_4$.

Доказательство. Необходимо доказать равенство двух множеств $\xrightarrow{R_1 \cup R_3} = \xrightarrow{R_2 \cup R_4}$. Пусть вначале $a \xrightarrow{R_1 \cup R_3} b$. Докажем, что при этом справедливо $a \xrightarrow{R_2 \cup R_4} b$. Для этого применим метод индукции по m – верхней оценке уровня рекурсии в соотношении $a \xrightarrow{R_1 \cup R_3} b$.

При $m = 0$ имеет место одно из условий 1)–2) определения 1. Если справедливо 2), то тогда $a \xrightarrow{R_2 \cup R_4} b$, поскольку логическое отношение $\xrightarrow{R_2 \cup R_4}$ содержит и $\leq$. Если выполнено условие 1), то имеем $(a,b) \in R_1 \bigcup R_3$. Пусть для определенности $(a,b) \in R_1$ (вариант $(a,b) \in R_3$ симметричен). В этом случае имеет место $a \xrightarrow{R_1} b$, откуда в силу условия эквивалентности $R_1 \sim R_2$ получим $a \xrightarrow{R_2} b$. Следовательно, справедливо и $a \xrightarrow{R_2 \cup R_4} b$. Таким образом, при $m = 0$ искомое утверждение доказано.

Предположим далее, что оно верно для некоторого $m \geq 0$, и докажем его при оценке уровня рекурсии $m + 1$. В этом случае новые для рассмотрения варианты связи $a \xrightarrow{R_1 \cup R_3} b$ могут дать условия 3)–6) определения 1.

Если последним было применено правило 3), то по предположению индукции уровень рекурсии в соответствующем соотношении $b' \xrightarrow{R_1 \cup R_3} a'$ не превосходит m, поэтому $b' \xrightarrow{R_2 \cup R_4} a'$. В этом случае, в силу контрапозиционности логического отношения $\xrightarrow{R_2 \cup R_4}$, получим $a \xrightarrow{R_2 \cup R_4} b$.

Рассмотрим вариант, когда соотношение $a \xrightarrow{R_1 \cup R_3} b$ получено из условия 4) определения 1. В этом случае по предположению индукции все соотношения $a \xrightarrow{R_1 \cup R_3} b_t$ имеют уровень рекурсии $\leq m$, поэтому $a \xrightarrow{R_2 \cup R_4} b_t$, $\forall t \in T$. Тогда, в силу $\bigwedge$-дистрибутивности $\xrightarrow{R_2 \cup R_4}$, получим $a \xrightarrow{R_2 \cup R_4} b$. Если же выполнено 5), то требуемый факт аналогично следует из предположения индукции и свойства $\bigvee$-дистрибутивности логического отношения $\xrightarrow{R_2 \cup R_4}$.

Наконец, пусть справедливо 6). И в этом случае базовые соотношения $a \xrightarrow{R_1 \cup R_3} c$ и $c \xrightarrow{R_1 \cup R_3} b$ имеют уровень рекурсии $\le m$. Следовательно, по предположению индукции получаем $a \xrightarrow{R_2 \cup R_4} c$ и $c \xrightarrow{R_2 \cup R_4} b$. Отсюда, в силу транзитивности логического отношения $\xrightarrow{R_2 \cup R_4}$, имеем $a \xrightarrow{R_2 \cup R_4} b$. □

Следствие 2. Пусть R_1, R_2, R – отношения на полной булевой решетке $\mathbb{F}$. Если при этом $R_1 \sim R_2$, то $R_1 \cup R \sim R_2 \cup R$.

Следствия 1 и 2 обосновывают принцип локальности эквивалентных преобразований логических отношений в случае полных булевых решеток.

4.2.2. Структура логических связей

В настоящем подразделе проводится аналогичное представленному в п.4.1.2 исследование с целью заложить основу для разделения процессов построения логического замыкания и редукции на этапы, один из которых соответствует нахождению транзитивного замыкания и редукции. Для этого доказываются некоторые свойства логических связей на полных булевых решетках.

Лемма 1. Пусть R – нетерово отношение на полной булевой решетке $\mathbb{F}$ и $a_t \xrightarrow{R} b_t, \forall t \in T$. Тогда справедливы соотношения $\bigwedge_{t \in T} a_t \xrightarrow{R} \bigwedge_{t \in T} b_t$ и $\bigvee_{t \in T} a_t \xrightarrow{R} \bigvee_{t \in T} b_t$.

Доказательство. Чтобы установить первое из требуемых соотношений, введем обозначения $\tilde{a} = \bigwedge_{t \in T} a_t, \tilde{b} = \bigwedge_{t \in T} b_t$. Поскольку $\tilde{a} \le a_t, \forall t \in T$, то в силу $a_t \xrightarrow{R} b_t$ и условия 6) определения 4.2.1.1[1] справедливо $\tilde{a} \xrightarrow{R} b_t, \forall t \in T$. Применяя к последнему соотношению правило 4) определения 4.2.1.1, получим $\tilde{a} \xrightarrow{R} \tilde{b}$.

Для доказательства второй части леммы обозначим $\tilde{a} = \bigvee_{t \in T} a_t$, $\tilde{b} = \bigvee_{t \in T} b_t$. Далее, пользуясь неравенствами $b_t \le \tilde{b}, \forall t \in T$, из $a_t \xrightarrow{R} b_t$ с помощью транзитивного условия 6) имеем $a_t \xrightarrow{R} \tilde{b}, \forall t \in T$. Наконец, используя правило 5) определения 4.2.1.1, приходим к соотношению $\tilde{a} \xrightarrow{R} \tilde{b}$. □

Следствие 1. На основе леммы 1 можно ввести обобщенные правила вывода, соответствующие правилам 4) и 5) определения 4.2.1.1. Упорядоченная пара $a, b \in \mathbb{F}$ логически связана отношением R ($a \xrightarrow{R} b$), если выполнено одно из следующих условий:

4′) существуют такие $a_t, b_t \in \mathbb{F}, t \in T$, что $\bigwedge_{t \in T} a_t = a, \bigwedge_{t \in T} b_t = b$, причем $a_t \xrightarrow{R} b_t, \forall t \in T$;

5′) существуют такие $a_t, b_t \in \mathbb{F}, t \in T$, что $\bigvee_{t \in T} a_t = a, \bigvee_{t \in T} b_t = b$, причем $a_t \xrightarrow{R} b_t, \forall t \in T$.

[1] Определение 1 из п.4.2.1.

Правила 4) и 5) определения 4.2.1.1 являются частными случаями правил 4′) и 5′).

Лемма 2. Пусть R – нетерово отношение на полной булевой решетке $\mathbb{F}$. Тогда при выводе произвольной логической связи $a \xrightarrow{R} b$ любое применение правил 4′)–5′) может быть произведено без участия в их компонентах правила 2) определения 4.2.1.1 со строгим неравенством. Его роль можно свести исключительно к присутствию в качестве компонента для транзитивного правила 6).

Доказательство. Предположим, что при получении вывода $a \xrightarrow{R} b$ использовалось условие 4′). Разобьем имеющееся множество пар $\{(a_t, b_t) \mid t \in T\}$ на 2 подмножества. В первое из них войдут такие пары (a_t, b_t), для которых выполнено $a_t < b_t$. Соответствующее им множество индексов $\{t\}$ обозначим T_1. Ко второму подмножеству отнесем все остальные пары (a_t, b_t), $t \in T_2$. Тогда, вводя обозначения $\bigwedge_{t \in T_1} a_t = a^1$, $\bigwedge_{t \in T_1} b_t = b^1$, $\bigwedge_{t \in T_2} a_t = a^2$, $\bigwedge_{t \in T_2} b_t = b^2$, получим $a = a^1 \wedge a^2$, $b = b^1 \wedge b^2$. При этом $a^1 \leq b^1$ (соответственно $a^1 \xrightarrow{R} b^1$) и по условию 4′) имеет место $a^2 \xrightarrow{R} b^2$. Очевидно, что указанное выше применение правила 4′) для $\{(a_t, b_t) \mid t \in T\}$ можно заменить аналогичным правилом для $(a^1, b^1), (a^2, b^2)$.

Если при этом окажется, что $T_1 = \varnothing$, то для данного вывода $a \xrightarrow{R} b$ сформулированное в лемме утверждение относительно правила 4′) выполнено сразу. Если же $T_2 = \varnothing$ либо $a^2 \leq b^2$, то необходимости в применении правила 4′) не возникает вообще. Рассмотрим оставшийся нетривиальный случай ($T_1 \neq \varnothing, T_2 \neq \varnothing$ и не имеет места $a^2 \leq b^2$), в котором поступим следующим образом. Применим правило 4′) к соотношениям $b^1 \xrightarrow{R} b^1$, $a^2 \xrightarrow{R} b^2$, тогда получим $b^1 \wedge a^2 \xrightarrow{R} b^1 \wedge b^2$. Возвращаясь к имеющемуся неравенству $a^1 \leq b^1$, заметим, что в силу свойств решетки $\mathbb{F}$ из него следует неравенство $a^1 \wedge a^2 \leq b^1 \wedge a^2$. Применяя далее к парам $a^1 \wedge a^2 \leq b^1 \wedge a^2$ и $b^1 \wedge a^2 \xrightarrow{R} b^1 \wedge b^2$ транзитивное правило 6), приходим к соотношению $a^1 \wedge a^2 \xrightarrow{R} b^1 \wedge b^2$, то есть $a \xrightarrow{R} b$. При этом в применяемом правиле 4′) исключается участие строгих неравенств вида $a_t < b_t$. Таким образом, для правила 4′) утверждение леммы доказано.

Рассмотрим другой вариант, когда при выводе $a \xrightarrow{R} b$ было применено правило 5′). Как и выше, разобьем имеющееся множество $\{(a_t, b_t) \mid t \in T\}$ на два подмножества. Первое из них будет содержать пары $a_t < b_t$. Соответствующее множество индексов обозначим T_1. Ко второму подмножеству отнесем все остальные пары (a_t, b_t), $t \in T_2$. Тогда, обозначая $\bigvee_{t \in T_1} a_t = a^1$, $\bigvee_{t \in T_1} b_t = b^1$, $\bigvee_{t \in T_2} a_t = a^2$, $\bigvee_{t \in T_2} b_t = b^2$, будем иметь

$a = a^1 \vee a^2$, $b = b^1 \vee b^2$. При этом $a^1 \le b^1$ (соответственно $a^1 \xrightarrow{R} b^1$) и по правилу 5′) имеет место $a^2 \xrightarrow{R} b^2$. Указанное выше применение правила 5′) для $\{(a_t, b_t) \mid t \in T\}$ можно заменить аналогичным правилом для $(a^1, b^1), (a^2, b^2)$.

Если выяснится, что $T_1 = \varnothing$, то для вывода $a \xrightarrow{R} b$ сформулированное в лемме утверждение относительно правила 5′) будет выполнено сразу. Если же $T_2 = \varnothing$ либо $a^2 \le b^2$, то применение правила 5′) не потребуется вообще. Остается нетривиальный случай ($T_1 \ne \varnothing, T_2 \ne \varnothing$ и не имеет места $a^2 \le b^2$), в котором применим к соотношениям $a^1 \xrightarrow{R} a^1, a^2 \xrightarrow{R} b^2$ правило 5′). Соответственно получим $a^1 \vee a^2 \xrightarrow{R} a^1 \vee b^2$. Возвращаясь к имеющемуся неравенству $a^1 \le b^1$, заметим, что в силу свойств решетки $\mathbb{F}$ из него следует неравенство $a^1 \vee b^2 \le b^1 \vee b^2$. Применяя к соотношениям $a^1 \vee a^2 \xrightarrow{R} a^1 \vee b^2$ и $a^1 \vee b^2 \le b^1 \vee b^2$ транзитивное правило 6), получим справедливость $a^1 \vee a^2 \xrightarrow{R} b^1 \vee b^2$, то есть $a \xrightarrow{R} b$. Теперь утверждение леммы доказано и для случая применения правила 5′). □

Лемма 3. Пусть R – нетерово отношение на полной булевой решетке $\mathbb{F}$. Тогда при выводе любой логической связи $a \xrightarrow{R} b$ все применения контрапозиционного правила 3) определения 4.2.1.1 могут быть исключены либо перенесены в начальную стадию этого процесса.

Доказательство. Достаточно показать, что любое применение правила 3) можно исключить либо поменять местами с предшествующим ему применением каждого из правил 2), 4′), 5′), 6), получив при этом аналогичную логическую связь. Рассмотрим все потенциально возможные при этом случаи.

Предположим, что вывод $a \xrightarrow{R} b$ получен из соотношения $b' \le a'$ применением правила 3). Поскольку $b' \le a'$, то в силу свойств булевой решетки $\mathbb{F}$ справедливо $a \le b$, откуда сразу следует $a \xrightarrow{R} b$. Таким образом, в данном случае применение правила 3) можно исключить.

Рассмотрим вариант, когда при получении некоторого вывода $a \xrightarrow{R} b$ непосредственно перед правилом 3) использовалось условие 4’). В подобной ситуации существуют такие $a_t, b_t \in \mathbb{F}$, $t \in T$, что $\bigwedge_{t \in T} a_t = a'$, $\bigwedge_{t \in T} b_t = b'$, причем $b_t \xrightarrow{R} a_t$, $\forall t \in T$. Вместо указанного последовательного применения правил 4′) и 3) поступим по-другому, а именно – сразу применим к имеющимся соотношениям $b_t \xrightarrow{R} a_t$, $\forall t \in T$ правило 3). Тогда получим, что справедливо $a_t' \xrightarrow{R} b_t'$, $\forall t \in T$. Применяя далее правило 5′), приходим к соотношению $\bigvee_{t\in T} a'_t \xrightarrow{R} \bigvee_{t\in T} b'_t$, откуда с помощью закона де Моргана получим $a \xrightarrow{R} b$. Таким образом, в данном случае применения правила 3) удается переместить на шаг назад, получив при этом аналогичный результат $a \xrightarrow{R} b$.

Рассмотрим ситуацию, в которой при получении вывода $a \xrightarrow{R} b$ перед правилом 3) было применено условие 5′). В этой ситуации существуют такие $a_t, b_t \in \mathbb{F},\ t \in T$, что $\bigvee_{t\in T} a_t = a',\ \bigvee_{t\in T} b_t = b'$, причем $b_t \xrightarrow{R} a_t,\ \forall t \in T$. Тогда вместо правил 5′) и 3) применим к имеющимся соотношениям $b_t \xrightarrow{R} a_t,\ \forall t \in T$ правило 3). В результате получим связи $a'_t \xrightarrow{R} b'_t,\ \forall t \in T$. Применяя теперь правило 4′), приходим к соотношению $\bigwedge_{t\in T} a'_t \xrightarrow{R} \bigwedge_{t\in T} b'_t$, откуда по закону де Моргана получим $a \xrightarrow{R} b$. И в этом случае применение правила 3) удалось переместить к началу вывода с сохранением результата $a \xrightarrow{R} b$.

Наконец, остается случай, когда при получении $a \xrightarrow{R} b$ перед правилом 3) было применено транзитивное правило 6). В этом случае существует элемент $c \in \mathbb{F}$ такой, что $b' \xrightarrow{R} c$ и $c \xrightarrow{R} a'$. Как и выше, к этим базовым соотношениям сразу применим правило 3). В результате получим, что $c' \xrightarrow{R} b$, $a \xrightarrow{R} c'$. Отсюда по правилу 6) приходим к соотношению $a \xrightarrow{R} b$.

Таким образом, в каждой возможной ситуации вывода применение правила 3) определения 4.2.1.1 можно произвести до применений остальных его правил. □

Лемма 4. Пусть R – нетерово отношение на полной булевой решетке $\mathbb{F}$. Тогда при выводе любой логической связи $a \xrightarrow{R} b$ все применения транзитивного правила 6) определения 4.2.1.1[1] могут быть исключены либо перенесены в заключительную стадию данного процесса.

Доказательство. Докажем утверждение леммы с помощью индукции по m – оценке уровня рекурсии в соотношении $a \xrightarrow{R} b$. При $m = 0$ для a, b имеет место одно из условий 1)–2) определения 4.2.1.1. В этом случае вывод $a \xrightarrow{R} b$ вовсе не содержит транзитивных связей, а значит, требуемое утверждение выполнено.

Предположим далее, что оно верно для некоторого $m \geq 0$, и докажем это утверждение при уровне рекурсии $m+1$. По предположению индукции, первые m шагов вывода можно организовать так, что транзитивное правило 6) будет применяться лишь в конце процесса. В этом случае все зависит от того, какое правило вывода

[1] Следствие 1 из п. 4.1.1.

применено на последнем ($m+1$) шаге. По лемме 2 правило 2) в этом плане можно не рассматривать, а по лемме 3 все применения контрапозиционного правила 3) можно считать реализованными в начале процесса вывода. Если же последним применено правило 6), то утверждение леммы сразу оказывается выполненным – все транзитивные связи использованы в заключительной стадии вывода. Таким образом, остается исследовать случаи, когда на ($m+1$) шаге применено правило вывода 4′) либо 5′).

Рассмотрим вариант, когда связь $a \xrightarrow{R} b$ в конечном счете получена из условия 4′). При этом каждое базовое соотношение $a_t \xrightarrow{R} b_t,\ t \in T$ имеет уровень рекурсии $\leq m$ и, по предположению индукции, все свои транзитивные связи использует лишь в конце вывода. Другими словами, при каждом $t \in T$ существует цепочка элементов $a_t = c_t^0, c_t^1, \ldots, c_t^{n_t} = b_t$ такая, что выполнены соотношения $c_t^{i-1} \xrightarrow{R} c_t^i$, $i = 1, \ldots, n_t$, при выводе каждого из которых не используется транзитивное правило 6). В силу нетеровости отношения R, величины n_t ограничены в совокупности, то есть $n_t \leq N,\ t \in T.$ В связи с этим обстоятельством будем считать все указанные выше цепочки элементов равными по длине N, дополнив каждую в случае необходимости справа повторяющимся элементом $c_t^{n_t} = b_t$. Тогда для цепочек $a_t = c_t^0, c_t^1, \ldots, c_t^N = b_t,\ t \in T$ построим элементы $\tilde{c}_i = \bigwedge_{t \in T} c_t^i,\ i = 0, \ldots, N$. Очевидно, что $\tilde{c}_0 = a$ и $\tilde{c}_N = b$. Кроме того, поскольку $c_t^{i-1} \xrightarrow{R} c_t^i$, то к этим парам можно применить правило 4′), то есть получаем $\tilde{c}_{i-1} \xrightarrow{R} \tilde{c}_i$, $i = 1, \ldots, N$.

Таким образом, удалось показать, что в рассматриваемом случае все применения транзитивного правила 6) в базовых соотношениях $a_t \xrightarrow{R} b_t,\ t \in T$ можно заменить его применением на заключительной стадии вывода к построенным элементам $\tilde{c}_i,\ i = 0, \ldots, N$.

Предположим теперь, что рассматриваемая связь $a \xrightarrow{R} b$ была выведена из условия 5′). Тогда все базовые соотношения $a_t \xrightarrow{R} b_t,\ t \in T$ имеют уровень рекурсии $\leq m$ и, по предположению индукции, все транзитивные связи использует только в конце вывода. По этой причине при каждом $t \in T$ существует цепочка элементов $a_t = c_t^0, c_t^1, \ldots, c_t^{n_t} = b_t$ такая, что выполнены соотношения $c_t^{i-1} \xrightarrow{R} c_t^i$, $i = 1, \ldots, n_t$, при выводе которых не используется транзитивное правило 6). Поскольку отношение R нетерово, то все величины n_t ограничены в совокупности, а именно – $n_t \leq N,\ t \in T.$ Как и выше, будем считать все цепочки элементов равными по длине N, расширив их в случае необходимости справа последними элементами $c_t^{n_t} = b_t$. Тогда для цепочек $a_t = c_t^0, c_t^1, \ldots, c_t^N = b_t,\ t \in T$ построим элементы $\tilde{c}_i = \bigvee_{t \in T} c_t^i,\ i = 0, \ldots, N$. Легко видеть, что $\tilde{c}_0 = a$ и

$\tilde{c}_N = b$. Кроме того, поскольку $c_t^{i-1} \xrightarrow{R} c_t^i$, то к этим парам можно применить правило 5′), то есть получаем $\tilde{c}_{i-1} \xrightarrow{R} \tilde{c}_i$, $i = 1,\dots,N$.

Итак, в рассматриваемом случае все применения транзитивного правила 6) в базовых соотношениях $a_t \xrightarrow{R} b_t$, $t \in T$ также можно заменить его применением на заключительной стадии вывода к построенным элементам $\tilde{c}_i$, $i = 0,\dots,N$. □

4.2.3. Логическая редукция

Для нетерового отношения R на полной булевой решетке $\mathbb{F}$ введем отношение $\tilde{R}$, построенное последовательным выполнением следующих действий (шагов):

1) добавить к R все пары вида (a,a), где $a \in \mathbb{F}$ (рефлексивные пары), и обозначить новое отношение R_1;

2) добавить к R_1 все пары (a,b), для которых $(b',a') \in R$, и обозначить новое отношение R_2;

3) добавить к R_2 всевозможные пары вида $(a_1 o_1 a_2 o_2 \dots o_{m-1} a_m,$ $b_1 o_1 b_2 o_2 \dots o_{m-1} b_m)$, где $(a_i, b_i) \in R_2$, символ o_i обозначает операцию $\wedge$ либо $\vee$, и обозначить новое отношение R_3;

4) добавить к R_3 всевозможные пары вида $(Q_1 Q_2 \dots Q_m a_{t_1 t_2 \dots t_m},$ $Q_1 Q_2 \dots Q_m b_{t_1 t_2 \dots t_m})$, где $(a_{t_1 t_2 \dots t_m}, b_{t_1 t_2 \dots t_m}) \in R_3$, символ Q_i обозначает бесконечную операцию вида $\bigwedge\limits_{t_i \in T_i}$ либо $\bigvee\limits_{t_i \in T_i}$;

5) объединить полученное отношение с отношением $\le$.

Заметим, что в силу дистрибутивности булевой решетки, действия шага 3) можно разделить на два более простых последовательных шага, а именно – вначале использовать лишь одну из операций $\wedge$ либо $\vee$, затем – только вторую.

Замечание 1. Применяя конечное число раз лемму 4.2.2.1[1] и следствие 4.2.1.1[2], нетрудно убедиться в том, что отношение $\tilde{R}$ логически эквивалентно R.

Лемма 1. Пусть R – нетерово отношение на полной булевой решетке $\mathbb{F}$. Тогда, если $a \xrightarrow{R} b$, и это соотношение может быть получено без использования транзитивного правила 6) определения 4.2.1.1[3], то $(a,b) \in \tilde{R}$.

Доказательство. Пусть имеет место $a \xrightarrow{R} b$, полученное без применений упомянутого в условии леммы правила 6). Если указанное соотношение получено непосредственно из правила 1) или 2) определения 4.2.1.1, то сразу имеем $(a,b) \in \tilde{R}$. Остается рас-

[1] Лемма 1 из п. 4.2.2.

[2] Следствие 1 из п. 4.2.1.

[3] Определение 1 из п. 4.2.1.

смотреть нетривиальный случай применения цепочки правил 2), 3), 4′), 5′).

Поскольку при выводе правило 6) не используется, то по лемме 4.2.2.2[1] не применяется и правило 2) со строгим неравенством. Далее, по лемме 4.2.2.3[2] все применения контрапозиционного правила 3) могут быть выполнены на начальном этапе процесса вывода. Следовательно, правила 4′)–5′) завершают этот процесс. Заметим также, что все применения правил 4′)–5′) с бесконечными операциями $\bigwedge$ и $\bigvee$ могут быть произведены после всех применений правил 4′)–5′) с конечными операциями $\wedge$ и $\vee$. Данный факт вытекает из свойств ассоциативности и дистрибутивности операций $\bigwedge$ и $\bigvee$ по отношению к операциям $\wedge$ и $\vee$ (см. п.1.2). Он является алгебраической интерпретацией известной теоремы о предваренной форме предиката в логике первого порядка [187].

Если сопоставить перечисленные этапы вывода с последовательностью построения отношения $\tilde{R}$, то окажется, что вывод $a \xrightarrow{R} b$ представляет собой построение некоторого подмножества $\tilde{R}$, что и доказывает включение $(a,b) \in \tilde{R}$. □

Теорема 1. Логическое замыкание нетерового отношения R на полной булевой решетке $\mathbb{F}$ совпадает с транзитивным замыканием $\tilde{R}^*$ соответствующего отношения $\tilde{R}$.

Доказательство. В силу замечания 1, отношение $\tilde{R}$ эквивалентно R. Следовательно, по определению логического замыкания, имеем $\tilde{R} \subseteq \xrightarrow{R}$. Отсюда, поскольку отношение $\xrightarrow{R}$ транзитивно, получаем $\tilde{R}^* \subseteq \xrightarrow{R}$.

Докажем обратное включение. Предположим, что $a \xrightarrow{R} b$. По лемме 4.2.2.4, при получении этого вывода все применения транзитивного правила 6) определения 4.2.1.1 могут быть перенесены в заключительную стадию процесса. Это означает, что существует цепочка элементов $a = c_0, c_1, \ldots, c_n = b$ такая, что выполнены соотношения $c_{i-1} \xrightarrow{R} c_i$, $i = 1, \ldots, n$, при выводе которых правило 6) не используется. Тогда по лемме 1 имеем $(c_{i-1}, c_i) \in \tilde{R}$, откуда сразу получаем $(a,b) \in \tilde{R}^*$. Итак, $\xrightarrow{R} \subseteq \tilde{R}^*$. □

Рассмотрим непосредственно вопрос о существовании и построении логической редукции бинарных отношений. Для нетерового отношения R на полной булевой решетке $\mathbb{F}$ введем отношение $\underset{\sim}{R}$, построенное последовательным выполнением действий (шагов), обратных построению $\tilde{R}$, а именно:

---

[1] Лемма 2 из п. 4.2.2.

[2] Лемма 3 из п. 4.2.2.

1) исключить из R все пары (a,b), для которых $a < b$, и обозначить новое отношение R_{-1};

2) исключить из R_{-1} все пары (a,b) вида $(Q_1 Q_2 ... Q_m a_{t_1 t_2 ... t_m},$ $Q_1 Q_2 ... Q_m b_{t_1 t_2 ... t_m})$, где $(a_{t_1 t_2 ... t_m}, b_{t_1 t_2 ... t_m}) \in R_{-1}$ либо $(b'_{t_1 t_2 ... t_m}, a'_{t_1 t_2 ... t_m}) \in R_{-1}$, символ Q_i обозначает бесконечную операцию вида $\bigwedge_{t_i \in T_i}$ либо $\bigvee_{t_i \in T_i}$, причем (a,b) не совпадает ни с одной парой $(a_{t_1 t_2 ... t_m}, b_{t_1 t_2 ... t_m})$, и обозначить новое отношение R_{-2};

3) исключить из R_{-2} все пары (a,b) вида $(a_1 o_1 a_2 o_2 ... o_{m-1} a_m,$ $b_1 o_1 b_2 o_2 ... o_{m-1} b_m)$, где $(a_i, b_i) \in R_{-2}$ либо $(b'_i, a'_i) \in R_{-2}$, символ o_i обозначает операцию $\wedge$ либо $\vee$, причем (a,b) не совпадает ни с одной парой (a_i, b_i), и обозначить новое отношение R_{-3};

4) исключить из R_{-3} все контрапозиционные пары, то есть при $(a,b),(b',a') \in R_{-3}$ оставить в R_{-3} лишь одну из двух пар, и обозначить новое отношение R_{-4};

5) исключить из R_{-4} все пары вида (a,a), где $a \in \mathbb{F}$ (рефлексивные пары).

В силу дистрибутивности булевой решетки, шаг 3) может быть заменен двумя более простыми последовательными шагами, каждый из которых использует лишь одну из операций $\wedge$ либо $\vee$.

Замечание 2. С помощью леммы 4.2.2.1 и следствия 4.2.1.1 нетрудно убедиться в том, что отношение $\underset{\sim}{R}$ логически эквивалентно R.

Лемма 2. Пусть R – нетерово отношение на полной булевой решетке $\mathbb{F}$. Для того, чтобы R являлось логической редукцией, необходимо и достаточно, чтобы R не содержало ни одной такой пары (a,b), для которой выполнено соотношение $a \xrightarrow{R\setminus\{(a,b)\}} b$.

Доказательство. Пусть отношение R представляет собой логическую редукцию, то есть является минимальным логически эквивалентным себе отношением. Предположим противное, а именно, что существует пара $(a,b) \in R$, логически связанная отношением $R \setminus \{(a,b)\}$. Если это так, то в силу следствия 4.2.1.1 пару (a,b) можно исключить из R, получив при этом меньшее эквивалентное отношение. Таким образом, при сделанном предположении отношение R не может быть логической редукцией.

Для доказательства обратного утверждения предположим, что не существует ни одной пары $(a,b) \in R$, для которой справедливо $a \xrightarrow{R\setminus\{(a,b)\}} b$. Необходимо доказать, что в этом случае R есть логическая редукция. Вновь предположим противное, а именно – пусть существует отношение $R_0 \subset R$, эквивалентное R, и $(a,b) \in R \setminus R_0$. Тогда, поскольку $(a,b) \in R$, в силу эквивалентности рассматриваемых отношений справедливо $a \xrightarrow{R_0} b$. Так как отношение R_0 не содержит пару (a,b), то $R_0 \subseteq R \setminus \{(a,b)\}$. Таким образом, логическая связь $a \xrightarrow{R_0} b$ противоречит сделанному предположению о том, что таких пар (a,b) в R нет. Полученное противоречие доказывает утверждение леммы. □

Следующая теорема устанавливает достаточное условие существования и способ построения логической редукции данного отношения.

Теорема 2. Пусть для нетерового отношения R, заданного на полной булевой решетке $\mathbb{F}$, построено соответствующее отношение $\tilde{R}$. Тогда, если для $\tilde{R}$ существует транзитивная редукция R^0, то соответствующее ей отношение $\underset{\sim}{R}^0$ представляет собой логическую редукцию исходного отношения R.

Доказательство. Из замечаний 1–2 следует, что указанное в теореме отношение $\underset{\sim}{R}^0$ логически эквивалентно R. Осталось показать, что $\underset{\sim}{R}^0$ является логической редукцией вообще. Для этого достаточно проверить выполнение для $\underset{\sim}{R}^0$ условия леммы 2.

Пусть (a,b) – произвольная пара отношения $\underset{\sim}{R}^0$. Требуется показать невозможность логической связи $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$. Предположим противное, что эта связь существует. Тогда в силу следствия 4.2.1.1 отношение $\underset{\sim}{R}^0 \setminus \{(a,b)\}$ эквивалентно $\underset{\sim}{R}^0$. Сразу заметим, что правило 1) определения 4.2.1.1 для вывода $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$ применить невозможно, поскольку пара (a,b) не содержится в множестве $\underset{\sim}{R}^0 \setminus \{(a,b)\}$.

По лемме 4.2.2.4 любая логическая связь может быть построена таким образом, что все транзитивности будут использоваться лишь в завершающей стадии ее вывода. Этот факт свидетельствует о существовании цепочки элементов $a = c_0, c_1, \ldots, c_n = b$ такой, что выполнены соотношения $c_{i-1} \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} c_i$, $i = 1, \ldots, n$, при выводе каждого из которых правило 6) не используется. Отсюда по лемме 1 имеем $(c_{i-1}, c_i) \in \tilde{R}$. Таким образом, при $n > 1$ пара (a,b) оказывается транзитивной в $\tilde{R}$. Следовательно, она не может содержаться в $\underset{\sim}{R}^0$, которое является подмножеством транзитивной редукции отношения $\tilde{R}$. В результате получено противоречие исходному предположению $(a,b) \in \underset{\sim}{R}^0$.

Остается исследовать случай $n = 1$. В этой ситуации по лемме 4.2.2.3 рассматриваемая логическая связь $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$ может содержать применения правил 4′) и 5′) лишь в заключительной стадии вывода. В силу свойств дистрибутивности полной булевой решетки [192], любая полученная таким образом пара (a,b) описывается шагами 3) и 4) процесса построения отношения $\tilde{R}$. Соответственно, при нахождении $\underset{\sim}{R}^0$ (обратный процесс) она будет исключена. Таким образом, снова получено противоречие исходному предположению $(a,b) \in \underset{\sim}{R}^0$.

При выводе $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$ не могло использоваться и правило 3) определения 4.2.1.1, поскольку в этом случае при $(a,b) \in \underset{\sim}{R}^0$ соответствующая контрапозиционная пара (b', a') должна быть исключена на шаге 4) построения $\underset{\sim}{R}^0$. Применение правила 2) при $a \neq b$ исключается леммой 4.2.2.2, а случай $a = b$ невозможен ввиду выполнения шага 5) процесса построения отношения $\underset{\sim}{R}^0$.

Таким образом, удалось исследовать возможные варианты предполагаемого логического вывода $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$. В результате установлено, что в каждом рассматриваемом случае наличие связи $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$ противоречит факту $(a,b) \in \underset{\sim}{R}^0$. Следовательно, отношение $\underset{\sim}{R}^0$ представляет собой логическую редукцию. □

В разделах 4.1 и 4.2 рассмотрены LP-структуры, которые могут использоваться для моделирования расширенных продукционных систем. По отношению к моделям, описанным в главах 2–3, указанные расширения состоят в том, что в предпосылках и заключениях правил допускаются достаточно общие логические формулы (соответственно формулы исчислений нулевого и первого порядка). Модель следующего раздела также расширяет общую теорию LP-структур. Однако это реализуется не обобщением логических формул, а добавлением новых нестандартных операций.

4.3. Эквациональные LP-структуры

Данный раздел посвящен описанию еще одной новой модели – эквациональной LP-структуры. Такая структура возникает в качестве модели условной эквациональной теории и соответствующей условной системы переписывания термов (см. п 1.6).

Кроме разработки самой модели, основные результаты раздела состоят в исследовании стандартных для данной работы свойств подобных структур. К их числу относятся: существование логического замыкания; возможность локально-эквивалентных преобразований исходного отношения; архитектура логических связей; существование и построение логической редукции LP-структуры. В силу происхождения этой модели в данном разделе вновь используются обозначения операций из теории множеств. В заключительной части раздела подводятся некоторые итоги и указываются перспективы дальнейших исследований в данном направлении.

4.3.1. Модель условной эквациональной теории

Связанные с термами и эквациональными теориями базовые определения были приведены в п. 1.6.

В настоящем разделе вводится основанная на решетках алгебраическая модель условной эквациональной теории. Эта модель учитывает возможные связи между термами, обусловленные применениями к ним функций и подстановок. Исходная теория состоит из условных соотношений вида $s_1 = t_1, \ldots, s_n = t_n : u_1 = v_1, \ldots, u_m = v_m$ (если имеют место равенства термов $s_i = t_i$, $i = 1, \ldots, n$, то выполнены и все $u_j = v_j$, $j = 1, \ldots, m$). Будем называть такие соотношения (условными) эквациональными правилами, или просто правилами там, где это не будет вызывать недоразумений. Равенства между термами интерпретируются обычным для эквациональной теории способом: $s = t$, если данное равенство можно получить из имеющегося набора равенств с помощью рассматриваемой эквациональной дедукции. Ниже будут определены соответствующие ей аксиомы и правила вывода, которые естественным образом расширяют набор аксиом и правил вывода, описанный в [46] для условных равенств вида $s_1 = t_1, \ldots, s_n = t_n : s = t$.

В предлагаемой алгебраической модели условные эквациональные правила реализуются бинарным отношением R на решетке, порожденной множествами равенств $\{s_i = t_i\}$. Подчеркнем, что R связывает не отдельные термы (как в обычной эквациональной логике), а наборы равенств между термами. Этот факт означает, что каждому правилу $s_1 = t_1, \ldots, s_n = t_n : u_1 = v_1, \ldots, u_m = v_m$ соответствует пара $(a, b) \in R$, где $a = \{s_1 = t_1, \ldots, s_n = t_n\}$ и $b = \{u_1 = v_1, \ldots, u_m = v_m\}$. Рассматриваемая модель содержит логику продукционного вывода. Такой вывод можно назвать мета-выводом, поскольку он применяется не к самим термам, а к подмножествам равенств между термами.

Пусть дана эквациональная теория (Σ, E). Для множества равенств E рассмотрим множество конечных подмножеств $\lambda(E)$. В нем заданы отношения включения $\subseteq$, $\supseteq$, а также «решеточные» операции $\bigcap$ и $\bigcup$ – теоретико-множественные пересечение и объединение. Кроме них для описания рассматриваемой модели потребуются еще две группы операций, связанных соответственно с функциями и подстановками термов:

1) если $a = \{s_i = t_i \mid i = 1, \ldots, n\}$ и $f \in \Sigma_n$, то $f(a) = \{f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)\}$;

2) если $a = \{s_j = t_j \mid j = 1, \ldots, m\}$, то $\sigma(a) = \{\sigma(s_j) = \sigma(t_j) \mid j = 1, \ldots, m\}$ для любой подстановки σ.

Определение 1. Пусть задана эквациональная теория (Σ, E). Эквациональной решеткой будем называть решетку, полученную пополнением $\lambda(E)$ относительно определенных выше операций 1)–2).

Таким образом, предпосылки и заключения рассматриваемых правил являются элементами эквациональной решетки.

По аналогии с [46] введем аксиомы и правила условной эквациональной дедукции. Ее аксиомы порождаются перечисленными в п. 1.6 правилами вывода равенств. Правило вывода 2) означает наличие условного правила (аксиомы) $a : f(a)$ для любых $a = \{s_1 = t_1, \dots, s_n = t_n\}$ и $f \in \Sigma_n$; из 3) следует аксиома $a : \sigma(a)$ для любых $a \in \mathbb{F}$ и подстановки σ. Такие условные правила в рассматриваемой логике можно также назвать тавтологиями. Еще одной очевидной тавтологией (аксиомой) является правило $a : b,\ a, b \in \mathbb{F}$ при $a \supseteq b$.

Правила вывода в условной эквациональной логике таковы:

1) $a : b \vdash \sigma(a) : \sigma(b)\ (a, b \in \mathbb{F})$ для любой подстановки σ (см. аналогичное правило в [46] с исходными обозначениями);

2) $a : b,\ a : c \vdash a : b \bigcup c\ (a, b, c) \in \mathbb{F}$ (возможность вывода по частям);

3) $a : b,\ b : c \vdash a : c\ (a, b, c) \in \mathbb{F}$ (транзитивность).

4.3.2. Логическое замыкание и эквивалентные преобразования

В данном подразделе рассматриваются бинарные отношения на эквациональной решетке. Ниже будет введено понятие логического отношения, которое соответствует множеству правил условной эквациональной теории. Свойства такого отношения должны отражать сформулированные в п. 4.3.1 аксиомы и правила условной эквациональной дедукции.

Во-первых, логическое отношение R должно содержать все тавтологии. Введем для них общее обозначение: $a \succcurlyeq b$, если $a \supseteq b$, $b = \sigma(a)$ или $b = f(a)$. Таким образом, для логического отношения R справедливо $\succcurlyeq \subseteq R$. Другие свойства логического отношения вытекают из правил дедукции. В частности, отношение R на эквациональной решетке называется применимым, если для любой подстановки σ из $(a, b) \in R$ следует $(\sigma(a), \sigma(b)) \in R$.

Бинарное отношение на эквациональной решетке называется логическим, если оно содержит тавтологии, а также является применимым, дистрибутивным и транзитивным. Логическим замыканием произвольного бинарного отношения R называется наименьшее логическое отношение, содержащее R. Эквациональной LP-структурой называется эквациональная решетка с заданным на ней логическим отношением.

Два отношения R_1 и R_2, определенные на общей эквациональной решетке, называются эквивалентными, если их логические замыкания совпадают. Этот факт, как обычно, будем обозначать $R_1 \sim R_2$. Логической редукцией данного отношения R называется эквивалентное ему минимальное отношение R_0.

Необходимо отметить, что из приведенных выше формулировок непосредственно не следует существования логического замыкания или редукции для произвольного бинарного отношения на эквациональной решетке. Ответы на эти вопросы будут получены ниже.

Определение 1. Пусть задано некоторое отношение R на эквациональной решетке $\mathbb{F}$. Будем констатировать, что упорядоченная пара $a,b \in \mathbb{F}$ логически связана отношением R (обозначим этот факт $a \xrightarrow{R} b$), если выполнено одно из перечисленных далее условий

1) $(a,b) \in R$;
2) $a \succcurlyeq b$:
   2.1) $a \supseteq b$ или
   2.2) $b = \sigma(a)$ или
   2.3) $b = f(a)$;
3) существуют такие $a_1, b_1 \in \mathbb{F}$ и подстановка σ, для которых $a = \sigma(a_1), b = \sigma(b_1)$, причем $a_1 \xrightarrow{R} b_1$;
4) существуют такие $b_1, b_2 \in \mathbb{F}$, для которых $b_1 \bigcup b_2 = b$, причем $a \xrightarrow{R} b_1, a \xrightarrow{R} b_2$;
5) существует элемент $c \in \mathbb{F}$ такой, что $a \xrightarrow{R} c$ и $c \xrightarrow{R} b$.

Определение 1 по данному R задает новое бинарное отношение $\xrightarrow{R}$ на эквациональной решетке $\mathbb{F}$.

Как обычно, условия 1)–5) определения 1 будем также называть правилами вывода в данной LP-структуре. При выводе логической связи $a \xrightarrow{R} b$ шагом вывода будем называть применение ровно одного правила, возможно, одновременно к некоторому конечному множеству элементов решетки. Например:

- если $a_i \xrightarrow{R} b_i,\ i = 1,\dots,n$, то $\sigma(a_i) \xrightarrow{R} \sigma(b_i),\ i = 1,\dots,n$;
- если $a_i \xrightarrow{R} c_i,\ c_i \xrightarrow{R} b_i,\ i = 1,\dots,n$, то $a_i \xrightarrow{R} b_i,\ i = 1,\dots,n$.

Уровнем рекурсии в соотношении $a \xrightarrow{R} b$ будем называть количество шагов вывода, необходимое для получения этой связи. В данном случае учитываются лишь применения правил 3)–5) определения 1. Для связи по правилу 1) или 2) уровень рекурсии считается равным нулю.

В общем случае связь $a \xrightarrow{R} b$ может быть получена не единственным набором правил. Как следствие, будем лишь оценивать ее уровень рекурсии, не указывая его точного значения.

Лемма 1. Пусть R – логическое отношение на эквациональной решетке $\mathbb{F}$ и $a, b \in \mathbb{F}$. Тогда, если $a \xrightarrow{R} b$, то справедливо $(a,b) \in R$.

Доказательство. Проведем его с помощью индукции по m – оценке уровня рекурсии в соотношении $a \xrightarrow{R} b$. При $m = 0$ выполнено одно из условий 1)–2) определения 1. Случай 1) означает справедливость утверждения леммы. Если же выполнено 2), то и тогда $(a,b) \in R$, поскольку логическое отношение R содержит тавтологии.

Предположим, что утверждение леммы верно для некоторого $m \geq 0$, и докажем его при уровне рекурсии $m+1$. В этом случае новые для рассмотрения варианты могут дать правила 3)–5).

Если имеет место 3), то по предположению индукции уровень рекурсии в соответствующем соотношении $a_1 \xrightarrow{R} b_1$ ($a = \sigma(a_1), b = \sigma(b_1)$) не превосходит m, поэтому $(a_1, b_1) \in R$. Тогда, в силу свойства применимости логического отношения R, получим $(a,b) \in R$.

Рассмотрим вариант, когда соотношение $a \xrightarrow{R} b$ выведено из условия 4) определения 1. При этом по предположению индукции соотношения $a \xrightarrow{R} b_1, a \xrightarrow{R} b_2$ имеют уровень рекурсии $\leq m$, т.е. $(a,b_1),(a,b_2) \in R$. Тогда, в силу дистрибутивности R, получим $(a,b) \in R$.

Наконец, пусть справедливо 5). И в этом случае по предположению индукции базовые соотношения $a \xrightarrow{R} c$ и $c \xrightarrow{R} b$ имеют уровень рекурсии $\leq m$, то есть $(a,c),(c,b) \in R$. Отсюда, в силу транзитивности логического отношения R, имеем $(a,b) \in R$. □

Теорема 1. Для произвольного отношения R на эквациональной решетке логическое замыкание существует и совпадает с множеством $\xrightarrow{R}$ всех упорядоченных пар, логически связанных отношением R.

Доказательство. Покажем, что при произвольном R соответствующее отношение $\xrightarrow{R}$ является логическим. Действительно, в силу условия 2) определения 1 оно содержит тавтологии, из 3) следует его применимость, из 4) – дистрибутивность, а условие 5) означает транзитивность данного отношения. Далее, в силу условия 1) определения 1, отношение $\xrightarrow{R}$ содержит R. Для доказательства теоремы осталось показать, что это – наименьшее из логических отношений, содержащих R.

Пусть R' – любое другое логическое отношение, содержащее R. Тогда очевидно, что если $a \xrightarrow{R} b$, то $a \xrightarrow{R'} b$. Отсюда по лемме 1 имеем $(a,b) \in R'$. Следовательно, отношение $\xrightarrow{R}$ содержится в произвольно выбранном R' и, таким образом, является наименьшим логическим отношением, содержащим R. □

Следствие 1. Пусть R – бинарное отношение на эквациональной решетке и $a_t \xrightarrow{R} b_t$, $\forall t \in T$. Тогда отношение $R' = R \bigcup \{(a_t, b_t) \mid t \in T\}$ эквивалентно R.

Доказательство. Заметим вначале, что по определению любое логическое отношение является собственным логическим замыканием. Этот факт в силу теоремы 1 относится и к отношению $\xrightarrow{R}$. Далее, из определения 1 следует, что для любых $R_1 \subseteq R_2$ справедливо $\xrightarrow{R_1} \subseteq \xrightarrow{R_2}$. В рассматриваемом случае по построению отношения R' выполнены включения $R \subseteq R' \subseteq \xrightarrow{R}$. Переходя к логическим замыканиям и учитывая изложенное выше, получаем, что отношения R и R' имеют общее логическое замыкание $\xrightarrow{R}$. □

Пусть дано произвольное бинарное отношение R на эквациональной решетке. Его эквивалентным преобразованием называется такая замена всего множества упорядоченных пар R или его части, что полученное в результате новое отношение P логически эквивалентно R, то есть $P \sim R$.

Теорема 2. Пусть R_1, R_2, R_3, R_4 – отношения на общей эквациональной решетке. Если при этом $R_1 \sim R_2$ и $R_3 \sim R_4$, то $R_1 \bigcup R_3 \sim R_2 \bigcup R_4$.

Доказательство. Пусть $a \xrightarrow{R_1 \bigcup R_3} b$. Докажем, что при этом справедливо $a \xrightarrow{R_2 \bigcup R_4} b$. Для этого применим метод индукции по m – оценке уровня рекурсии в соотношении $a \xrightarrow{R_1 \bigcup R_3} b$.

При $m = 0$ имеет место одно из условий 1)–2) определения 1. Если справедливо 2), то тогда $a \xrightarrow{R_2 \bigcup R_4} b$, поскольку логическое отношение $\xrightarrow{R_2 \bigcup R_4}$ содержит и тавтологии. Если выполнено условие 1), то имеем $(a,b) \in R_1 \bigcup R_3$. Пусть, для определенности, $(a,b) \in R_1$ (вариант $(a,b) \in R_3$ аналогичен). В этом случае имеет место $a \xrightarrow{R_1} b$, откуда в силу условия эквивалентности $R_1 \sim R_2$ получим $a \xrightarrow{R_2} b$. Следовательно, справедливо и $a \xrightarrow{R_2 \bigcup R_4} b$. Таким образом, при $m = 0$ требуемое утверждение доказано.

Предположим далее, что оно верно для некоторого $m \geq 0$, и докажем его при уровне рекурсии $m + 1$. В этом случае новые для рассмотрения варианты для $a \xrightarrow{R_1 \bigcup R_3} b$ могут дать условия 3)–5) определения 1.

Если имеет место 3), то по предположению индукции уровень рекурсии в соответствующем соотношении $a_1 \xrightarrow{R_1 \bigcup R_3} b_1$

($a=\sigma(a_1), b=\sigma(b_1)$) не превосходит m, поэтому $a_1 \xrightarrow{R_2 \cup R_4} b_1$. В этом случае, в силу свойства применимости логического отношения $\xrightarrow{R_2 \cup R_4}$, получим $a \xrightarrow{R_2 \cup R_4} b$.

Рассмотрим вариант, когда соотношение $a \xrightarrow{R_1 \cup R_3} b$ получено из условия 4) определения 1. Тогда по предположению индукции соотношения $a \xrightarrow{R_1 \cup R_3} b_1, a \xrightarrow{R_1 \cup R_3} b_2$ имеют уровень рекурсии $\leq m$, поэтому $a \xrightarrow{R_2 \cup R_4} b_1, a \xrightarrow{R_2 \cup R_4} b_2$. Как следствие, в силу дистрибутивности $\xrightarrow{R_2 \cup R_4}$, получим $a \xrightarrow{R_2 \cup R_4} b$.

Наконец, пусть справедливо 5). И в этом случае базовые соотношения $a \xrightarrow{R_1 \cup R_3} c$ и $c \xrightarrow{R_1 \cup R_3} b$ имеют уровень рекурсии $\leq m$. Следовательно, по предположению индукции получаем $a \xrightarrow{R_2 \cup R_4} c$ и $c \xrightarrow{R_2 \cup R_4} b$. Отсюда, в силу транзитивности логического отношения $\xrightarrow{R_2 \cup R_4}$, имеем $a \xrightarrow{R_2 \cup R_4} b$. □

Следствие 2. Пусть R_1, R_2, R – отношения на общей эквациональной решетке. Если при этом $R_1 \sim R_2$, то $R_1 \cup R \sim R_2 \cup R$.

Следствия 1 и 2 обосновывают принцип локальности эквивалентных преобразований логических отношений на эквациональной решетке.

4.3.3. Структура логических связей

Для выяснения вопроса о возможности в общем процессе построения логического замыкания выделить этап транзитивного замыкания исследуем ряд свойств продукционно-логических связей на эквациональной решетке.

Лемма 1. Пусть R – отношение на эквациональной решетке и $a_i \xrightarrow{R} b_i$, $i=1,\ldots,n$. Тогда справедлива логическая связь $\bigcup_{i=1,\ldots,n} a_i \xrightarrow{R} \bigcup_{i=1,\ldots,n} b_i$.

Доказательство аналогично доказательству леммы 4.1.2.1[1]. □

Следствие 1. На основе леммы 1 можно ввести обобщенное правило вывода, соответствующее правилу 4) определения 4.3.2.1[2]. Оно состоит в том, что упорядоченная пара $a, b \in \mathbb{F}$ логически связана отношением R ($a \xrightarrow{R} b$), если выполнено следующее условие:

4′) существуют такие $a_i, b_i \in \mathbb{F}$, $i=1,\ldots,n$, что $\bigcup_{i=1,\ldots,n} a_i = a$, $\bigcup_{i=1,\ldots,n} b_i = b$, причем $a_i \xrightarrow{R} b_i$, $i=1,\ldots,n$.

Очевидно, что правило 4) определения 4.3.2.1 является частным случаем правила 4′).

[1] Лемма 1 из п. 4.1.2.

[2] Определение 1 из п. 4.3.2.

Лемма 2. Пусть R – отношение на эквациональной решетке $\mathbb{F}$. Тогда при выводе каждой логической связи $a \xrightarrow{R} b$ все применения правила 4′) могут быть произведены без участия правила 2.1) определения 4.3.2.1 со строгим неравенством. Роль последнего можно свести его к участию лишь в качестве компонента для транзитивного правила 5).

Доказательство аналогично доказательству леммы 4.1.2.2. □

Лемма 3. Пусть R – отношение на эквациональной решетке $\mathbb{F}$. Тогда при выводе логической связи $a \xrightarrow{R} b$ любое применение правила 3) определения 4.3.2.4 можно исключить либо поменять местами с предшествующим ему применением каждого из правил 2.1), 4′), 5), получив при этом ту же логическую связь.

Доказательство. Предположим, что вывод $a \xrightarrow{R} b$ получен из некоторого соотношения $a' \xrightarrow{R} b'$ применением правила 3). Тогда существует такая подстановка σ, что $a = \sigma(a'), b = \sigma(b')$. Рассмотрим возможные здесь случаи. Если имело место $a' \supseteq b'$ (правило 2.1), то выполнено $\sigma(a') \supseteq \sigma(b')$ и, соответственно, необходимость в применении правила 3) не возникает.

Рассмотрим случай, когда при получении вывода $a' \xrightarrow{R} b'$ непосредственно перед правилом 3) использовалось условие 4′). В этой ситуации существуют такие $a_i, b_i \in \mathbb{F},\ i = 1,\dots,n$, что $\bigcup\limits_{i=1,\dots,n} a_i = a',\ \bigcup\limits_{i=1,\dots,n} b_i = b'$, причем $a_i \xrightarrow{R} b_i,\ i = 1,\dots,n$. Вместо указанного последовательного применения правил 4′) и 3) поступим по-другому, а именно – сразу применим к имеющимся соотношениям $a_i \xrightarrow{R} b_i,\ i = 1,\dots,n$ правило 3). Тогда получим, что справедливо $\sigma(a_i) \xrightarrow{R} \sigma(b_i),\ i = 1,\dots,n$. Применяя далее правило 4′), приходим к соотношению $\bigcup\limits_{i=1,\dots,n} \sigma(a_i) \xrightarrow{R} \bigcup\limits_{i=1,\dots,n} \sigma(b_i)$, то есть $\sigma(\bigcup\limits_{i=1,\dots,n} a_i) \xrightarrow{R} \sigma(\bigcup\limits_{i=1,\dots,n} b_i)$. Таким образом, в данном случае путем применения правила 3) удается переместиться на шаг назад, получив при этом аналогичный результат $a \xrightarrow{R} b$.

Наконец, остается случай, когда при получении $a \xrightarrow{R} b$ перед правилом 3) было применено транзитивное правило 5). В этом случае существует элемент $c \in \mathbb{F}$ такой, что $a' \xrightarrow{R} c$ и $c \xrightarrow{R} b'$. Как и выше, к этим базовым соотношениям сразу применим правило 3). В результате получим, что $\sigma(a') \xrightarrow{R} \sigma(c)$, $\sigma(c) \xrightarrow{R} \sigma(b')$. Отсюда по правилу 5) приходим к соотношению $a \xrightarrow{R} b$. □

Лемма 4. Пусть R – отношение на эквациональной решетке. Тогда при выводе любой логической связи $a \xrightarrow{R} b$ все применения транзитивного правила 5) могут быть исключены либо перенесены в заключительную стадию данного процесса.

Доказательство. Докажем сформулированное утверждение с помощью индукции по m – оценке уровня рекурсии в соотношении $a \xrightarrow{R} b$. При $m = 0$ для a, b имеет место одно из условий 1)–2) определения 4.3.2.1. В этом случае вывод $a \xrightarrow{R} b$ вообще не содержит транзитивных связей. Как следствие, утверждение леммы выполнено.

Предположим далее, что оно верно для некоторого $m \geq 0$, и докажем это утверждение при уровне рекурсии $m+1$. По предположению индукции, первые m шагов вывода можно организовать так, что транзитивное правило 5) будет применяться лишь в конце процесса. В этом случае все зависит от того, какое правило вывода применено на последнем ($m+1$)-м шаге – 3), 4′) или 5). По лемме 3 все применения правила 3) можно считать реализованными до использования транзитивности. Следовательно, если оно было использовано на ($m+1$)-м шаге, то при выводе необходимости в использовании транзитивных связей не возникает вообще. Если последним применено правило 5), то утверждение леммы сразу оказывается выполненным, так как все транзитивные связи использованы в заключительной стадии вывода. Таким образом, остается исследовать случай, когда на ($m+1$)-м шаге применено правило вывода 4′).

Пусть связь $a \xrightarrow{R} b$ в конечном счете получена из условия 4′). Каждое базовое соотношение $a_i \xrightarrow{R} b_i$, $i = 1,\ldots,n$ при этом имеет уровень рекурсии $\leq m$ и, по предположению индукции, все свои транзитивные связи использует лишь в конце вывода. Другими словами, при каждом $i = 1,\ldots,n$ существует цепочка элементов $a_i = c_i^0, c_i^1, \ldots, c_i^{n_i} = b_i$ такая, что выполнены соотношения $c_i^{k-1} \xrightarrow{R} c_i^k$, $k = 1,\ldots,n_i$, при выводе которых не используется транзитивное правило 5). По причине конечности n величины n_i ограничены в совокупности, что означает $n_i \leq N$, $i = 1,\ldots,n$. В связи с этим фактом будем считать все указанные выше цепочки элементов равными по длине N, дополнив каждую из них в случае необходимости справа повторяющимся элементом $c_i^{n_i} = b_i$. Тогда для цепочек $a_i = c_i^0, c_i^1, \ldots, c_i^N = b_i$, $i = 1,\ldots,n$ построим элементы $\tilde{c}_k = \bigcup\limits_{i=1,\ldots,n} c_i^k$, $k = 0,\ldots,N$. Очевидно, что $\tilde{c}_0 = a$ и $\tilde{c}_N = b$. Кроме того, поскольку $c_i^{k-1} \xrightarrow{R} c_i^k$, $i = 1,\ldots,n$, то к этим парам можно применить правило 4′), то есть получаем $\tilde{c}_{k-1} \xrightarrow{R} \tilde{c}_k$, $k = 1,\ldots,N$.

Таким образом, удалось показать, что в рассматриваемом случае все применения транзитивного правила 5) в базовых соотношениях $a_i \xrightarrow{R} b_i$, $i = 1,\ldots,n$ можно заменить его применением на заключительной стадии вывода к построенным элементам $\tilde{c}_k$, $k = 0,\ldots,N$. □

4.3.4. Логическая редукция

Для произвольного отношения R на эквациональной решетке рассмотрим отношение $\tilde{R}$, построенное по R последовательным выполнением следующих действий (шагов):

1) добавить к R все пары (a,b), для которых $b = \sigma(a)$ либо $b = f(a)$, и обозначить новое отношение R_1;

2) добавить к R_1 всевозможные пары вида $(\sigma(a),\sigma(b))$, для которых $(a,b) \in R_1$, и обозначить новое отношение R_2;

3) добавить к R_2 всевозможные пары вида $(a_1 \cup a_2 \cup \ldots \cup a_m,$ $b_1 \cup b_2 \cup \ldots \cup b_m)$, где $(a_i,b_i) \in R_2$, и обозначить новое отношение R_3;

4) объединить R_3 с отношением $\supset$.

Заметим, что в силу бесконечности множества $\mathbb{F}$ описанный процесс построения $\tilde{R}$ носит теоретический характер. В приложениях в качестве решетки $\mathbb{F}$ можно взять конечное подмножество эквациональной решетки, построенное при ограничении максимального уровня вложенности подстановок термов.

Замечание 1. Применяя лемму 4.3.3.1[1] и следствие 4.3.2.1[2], нетрудно убедиться в том, что отношение $\tilde{R}$ логически эквивалентно R.

Лемма 1. Пусть R – отношение на эквациональной решетке. Тогда, если $a \xrightarrow{R} b$, и это соотношение может быть получено без использования транзитивного правила 5) определения 4.3.2.1, то $(a,b) \in \tilde{R}$.

Доказательство. Пусть имеет место $a \xrightarrow{R} b$, полученное без правила 5). Если указанное соотношение произошло непосредственно из условия 1) или 2) определения 4.3.2.1, то сразу имеем $(a,b) \in \tilde{R}$.

Остается рассмотреть нетривиальный случай применения цепочки правил 3), 4′). Все необходимые для этого тавтологии могут быть подготовлены в самом начале процесса вывода. При этом в силу лемм 4.3.3.2 и 4.3.3.3 пары вида $a \supset b$ не потребуются. Также по лемме 4.3.3.3 все применения правила 3) могут быть реализованы на начальном этапе процесса вывода. Следовательно, правило 4′) может лишь завершать этот процесс.

Если сопоставить указанные этапы вывода с последовательностью построения отношения $\tilde{R}$, то окажется, что вывод $a \xrightarrow{R} b$

[1] Лемма 1 из п. 4.3.3.

[2] Следствие 1 из п. 4.3.2.

соответствует построению некоторого подмножества $\tilde{R}$, что и доказывает включение $(a,b)\in\tilde{R}$. □

Теорема 1. Логическое замыкание отношения R на эквациональной решетке совпадает с транзитивным замыканием $\tilde{R}^*$ соответствующего отношения $\tilde{R}$.

Доказательство. Как было отмечено выше, отношение $\tilde{R}$ эквивалентно R. Следовательно, по определению логического замыкания, имеем $\tilde{R}\subseteq\xrightarrow{R}$. Отсюда, поскольку отношение $\xrightarrow{R}$ транзитивно, получаем $\tilde{R}^*\subseteq\xrightarrow{R}$.

Докажем обратное включение. Предположим, что $a\xrightarrow{R}b$. По лемме 4.3.3.4, при получении этого вывода все применения транзитивного правила 5) определения 4.3.2.1 могут быть перенесены в заключительную стадию процесса. Данное утверждение означает, что существует цепочка элементов $a=c_0,c_1,\dots,c_N=b$ такая, что выполнены соотношения $c_{k-1}\xrightarrow{R}c_k$, $k=1,\dots,N$, при выводе которых правило 5) не используется. Тогда по лемме 1 имеем $(c_{k-1},c_k)\in\tilde{R}$, откуда сразу получаем $(a,b)\in\tilde{R}^*$. Итак, $\xrightarrow{R}\subseteq\tilde{R}^*$. □

Переходим к непосредственному рассмотрению вопроса о существовании и построении логической редукции бинарных отношений на эквациональной решетке. Для произвольного отношения R введем новое отношение $\underset{\sim}{R}$, построенное из R последовательным выполнением действий (шагов), обратных построению $\tilde{R}$:

1) исключить из R все пары (a,b), для которых $a\supset b$, и обозначить новое отношение R_{-1}.

2) исключить из R_{-1} все пары (a,b) вида $(a_1\cup a_2\cup\ldots\cup a_m$, $b_1\cup b_2\cup\ldots\cup b_m)$, где $(a_i,b_i)\in R_{-1}$, причем (a,b) не совпадает ни с одной парой (a_i,b_i), и обозначить новое отношение R_{-2};

3) исключить из R_{-2} всевозможные пары вида $(\sigma(a),\sigma(b))$, для которых $(a,b)\in R_{-2}$, причем (a,b) не совпадает с парой $(\sigma(a),\sigma(b))$, и обозначить новое отношение R_{-3};

4) исключить из R_{-3} все пары (a,b), для которых $b=\sigma(a)$ либо $b=f(a)$.

Замечание 2. С помощью леммы 4.3.3.1 и следствия 4.3.2.1 можно убедиться в том, что отношение $\underset{\sim}{R}$ логически эквивалентно R.

Лемма 2. Пусть R – бинарное отношение на эквациональной решетке. Для того чтобы R являлось логической редукцией, необходимо и достаточно, чтобы R не содержало ни одной такой пары (a,b), что выполнено соотношение $a\xrightarrow{R\setminus\{(a,b)\}}b$.

Доказательство. Пусть отношение R представляет собой логическую редукцию, то есть является минимальным логически эквивалентным себе отношением. Если бы существовала пара $(a,b) \in R$, логически связанная отношением $R \setminus \{(a,b)\}$, то в силу следствия 4.3.2.1 ее можно было бы исключить из R, получив при этом меньшее эквивалентное отношение. Таким образом, при наличии указанной пары отношение R не может быть логической редукцией.

Для доказательства обратного утверждения предположим, что не существует ни одной пары $(a,b) \in R$, для которой справедливо $a \xrightarrow{R\setminus\{(a,b)\}} b$. Необходимо доказать, что в этом случае R есть логическая редукция. Предположим противное – пусть существует отношение $R_0 \subset R$, эквивалентное R, причем $(a,b) \in R \setminus R_0$. Тогда, поскольку $(a,b) \in R$, в силу эквивалентности рассматриваемых отношений справедливо $a \xrightarrow{R_0} b$. Так как отношение R_0 не содержит пару (a,b), то $R_0 \subseteq R \setminus \{(a,b)\}$, и логическая связь $a \xrightarrow{R_0} b$ противоречит сделанному предположению – таких пар (a,b) в R нет. Полученное противоречие доказывает требуемое утверждение. □

Следующая теорема указывает достаточное условие существования и способ построения логической редукции данного отношения.

Теорема 2. Пусть для отношения R на эквациональной решетке $\mathbb{F}$ построено соответствующее отношение $\tilde{R}$. Тогда, если для $\tilde{R}$ существует транзитивная редукция R^0, то соответствующее ей отношение $\underset{\sim}{R}^0$ представляет собой логическую редукцию исходного отношения $\tilde{R}$.

Доказательство. Из замечаний 1–2 следует, что указанное в теореме отношение $\underset{\sim}{R}^0$ логически эквивалентно R. Осталось показать, что $\underset{\sim}{R}^0$ является логической редукцией. Для этого достаточно проверить выполнение для $\underset{\sim}{R}^0$ условия леммы 2.

Пусть (a,b) – произвольная пара отношения $\underset{\sim}{R}^0$. Необходимо показать, что логическая связь $a \xrightarrow{\underset{\sim}{R}^0\setminus\{(a,b)\}} b$ невозможна. Предположим противное, а именно, что эта связь существует. Тогда в силу следствия 4.3.2.1 отношение $\underset{\sim}{R}^0 \setminus \{(a,b)\}$ эквивалентно $\underset{\sim}{R}^0$. Заметим, что применение правила 1) определения 4.3.2.1 для вывода $a \xrightarrow{\underset{\sim}{R}^0\setminus\{(a,b)\}} b$ невозможно. Причина в том, что пара (a,b) не содержится в множестве $\underset{\sim}{R}^0 \setminus \{(a,b)\}$.

По лемме 4.3.3.4 любая логическая связь может быть построена таким образом, что все применения транзитивного правила будут требоваться лишь в завершающей стадии ее вывода. Этот факт означает, что существует цепочка элементов $a = c_0, c_1, \dots, c_N = b$ такая, что выполнены соотношения $c_{k-1} \xrightarrow{\underset{\sim}{R}^0\setminus\{(a,b)\}} c_k$, $k = 1, \dots, N$, при выводе каждого из которых правило 5) не применяется. Отсюда по лемме 1 имеем $(c_{k-1}, c_k) \in \tilde{R}$. Таким образом, при $N > 1$ пара (a,b)

оказывается транзитивной в $\tilde{R}$. Следовательно, она не может содержаться в $\underset{\sim}{R}^0$, являющемся подмножеством транзитивной редукции отношения $\tilde{R}$. В результате получено противоречие исходному предположению $(a,b) \in \underset{\sim}{R}^0$.

Остается исследовать случай $N = 1$. В этой ситуации по лемме 4.3.3.3 существует вывод логической связи $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$, который может содержать применения правила 4′) лишь в заключительной стадии. Полученная таким образом пара (a,b) описывается шагом 3) процесса построения отношения $\tilde{R}$. Соответственно при нахождении $\underset{\sim}{R}^0$ (обратный процесс) она будет исключена. Таким образом, снова приходим к противоречию исходному предположению $(a,b) \in \underset{\sim}{R}^0$.

При выводе $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$ не могло использоваться и правило 3) определения 4.3.2.1. Причина состоит в том, что в этом случае при $(a',b') \in \underset{\sim}{R}^0$ соответствующая пара $(a,b) = (\sigma(a'), \sigma(b'))$ должна быть исключена на шаге 3) построения $\underset{\sim}{R}^0$. Применение правила 2) при $a \supset b$ исключается леммой 4.3.3.2, а остальные варианты правила 2) невозможны ввиду выполнения шага 4) процесса построения отношения $\underset{\sim}{R}^0$.

Таким образом, удалось исследовать потенциально возможные ситуации для предполагаемого логического вывода $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$. В результате установлено, что в каждом таком случае наличие связи $a \xrightarrow{\underset{\sim}{R}^0 \setminus \{(a,b)\}} b$ противоречит факту $(a,b) \in \underset{\sim}{R}^0$. Следовательно, отношение $\underset{\sim}{R}^0$ представляет собой логическую редукцию. □

4.3.5. Некоторые итоги

Как и в предыдущих главах, требование существования транзитивной редукции отношения $\tilde{R}$ на эквациональной решетке является избыточным для существования логической редукции исходного отношения R. Поскольку эквациональная решетка $\mathbb{F}$ по определению бесконечна, при объединении R с отношением $\supseteq$ есть опасность получения не имеющего транзитивной редукции отношения $\tilde{R}$. Таким образом, для усиления теоремы 4.3.4.2 предлагается использование всюду в изложенных выше построениях вместо $\supseteq$ отношения вида $\supseteq_R$, содержащего лишь необходимое для получения логической редукции R подмножество $\supseteq$. Кроме того, при практической реализации можно вместо эквациональной решетки рассмотреть ее конечное подмножество, построенное на основе ограничения уровня вложенности подстановок термов.

Кратко проиллюстрируем процесс минимизации множества правил на модифицированном примере из п. 1.6:

1) $x + y = z : s(x) + y = s(z)$;
2) $h(x + y) = h(z) : g(x) + y = g(z)$;
3) $s(x) + y = s(z), g(x) + y = g(z) : f(x) = f(z)$;
4) $x + y = z : f(x) = f(z)$.

Введем элементы решетки $\mathbb{F}$:

$$A \sim \{x + y = z\};\ B \sim \{s(x) + y = s(z)\};\ C \sim \{h(x + y) = h(z)\};$$
$$D \sim \{g(x) + y = g(z)\};\ E \sim \{f(x) = f(z)\}.$$

Они обозначены большими буквами, чтобы не было дублирования с символами термов. Исходное бинарное отношение R состоит из пар $(A,B),(C,D),(B \cup D,E),(A,E)$, соответствующих условным правилам 1)–4). Не будем описывать здесь полный процесс построения отношения $\tilde{R}$. Для краткости сформируем лишь его подмножество, которого будет достаточно для решения задачи. Итак, в силу очевидного равенства $C = h(A)$, имеем тавтологию (A,C), которая при построении $\tilde{R}$ должна быть добавлена на шаге 1). На этом же шаге добавим еще одну тавтологию – (B,B). Далее, согласно шагу 3) процесса построения $\tilde{R}$, в это множество необходимо включить пару $(A, B \cup C)$ (на основе имеющихся пар $(A,B),(A,C)$) и пару $(B \cup C, B \cup D)$ (на основе $(B,B),(C,D)$). Таким образом, имеем

$$\left\{\begin{matrix}(A,B),(C,D),(B \cup D,E),(A,E),(A,C),\\ (B,B),(A,B \cup C),(B \cup C,B \cup D)\end{matrix}\right\} \le \tilde{R}.$$

Заметим, что на практике при построении $\tilde{R}$ не обязательно физически добавлять тавтологии. Достаточно реализовать эффективный механизм проверки, является ли заданная пара тавтологией.

Производя транзитивную редукцию полученного множества, заметим наличие в нем цепочки пар $(A,B \cup C),(B \cup C,B \cup D),(B \cup D,E)$. Отсюда следует транзитивность в $\tilde{R}$ пары (A,E). Таким образом, пара (A,E) будет исключена при нахождении транзитивной редукции R^0 отношения $\tilde{R}$. Дальнейшие действия соответствуют построению для R^0 отношения $\underset{\sim}{R}^0$. На этом этапе будут исключены добавленные ранее пары $(B \cup C, B \cup D)$, $(A, B \cup C)$ и тавтологии (B,B), (A,C).

В результате получим отношение $\underset{\sim}{R}^0 = \{(A,B),(C,D),(B \cup D,E)\}$, которое является логической редукцией исходного отношения R и соответствует первым трем условным эквациональным правилам примера.

Полученные в настоящем разделе результаты обосновывают локально-эквивалентные преобразования множества условных правил, а также способ его оптимизации путем получения эквивалентной системы с минимальным набором правил.

Дальнейшие исследования на рассматриваемом направлении могут быть связаны с более глубоким анализом структуры условных правил. В п. 1.6 указывалось, что, заменив термы независимыми элементами некоторого множества, можно оптимизировать систему правил на основе методов главы 2. Этот уровень исследований можно считать первым. В настоящем разделе предложена модель второго уровня – с учетом связей между равенствами на основе функций и подстановок. Более глубокий третий уровень мог бы учитывать в равенствах структуру отдельных термов. Рассмотрим пример:

1) $x+y=z : s(x)+y=s(z)$;
2) $x+y=z : s(z)=g(x+z)$;
3) $x+y=z : s(x)+y=g(x+z)$.

Здесь условное правило 3) является «лишним»: при $x+y=z$ по правилам 1)–2) из $s(x)+y=s(z)$ и $s(z)=g(x+z)$ автоматически следует $s(x)+y=g(x+z)$. Однако этот факт можно заметить лишь на третьем уровне исследования.

На этом же уровне лежат и подходы к решению других задач, зависящих от структуры равенств и термов. К таковым, например, относится вопрос о влиянии внешних переменных [65] на возможности логической редукции (такие переменные имеются в правилах 1)–3)). Результаты исследований, представленные в данной работе, от внешних переменных не зависят.

В дальнейшем можно исследовать более общие алгебраические модели условных эквациональных теорий, если в качестве основы LP-структур вместо $\lambda(E)$ брать другие решетки. Например, при использовании решетки Линденбаума–Тарского [187] моделируемые условные правила смогут в качестве предпосылок и заключений содержать формулы пропозиционального исчисления. Таким образом можно будет рассматривать расширенные модели систем переписывания термов (например, [86, 99]). Общие методы исследования при этом останутся прежними.

Интересным представляется переход к модели ориентированных правил и выяснение вопроса о том, как логически эквивалентные преобразования множества правил влияют на основные свойства исходной условной СПТ (нетеровость, конфлюэнтность). Можно предположить, что в силу эквивалентности преобразований основные свойства СПТ сохранятся, однако этот вопрос нуждается в формальных исследованиях.

В настоящей главе были рассмотрены классы LP-структур, адекватные расширенным моделям продукционных систем. Их логика, являясь более сложной по сравнению со стандартными системами (главы 2–3), тем не менее сохраняет важное характеризующее ее свойство – монотонность. Однако существует ряд важных практических задач искусственного интеллекта, которые могут эффективно решаться на основе моделей продукционных систем *с немонотонным выводом*. Построению соответствующих им LP-структур посвящена следующая глава настоящей работы.

Глава 5
НЕМОНОТОННЫЕ LP-СТРУКТУРЫ

Применение рассмотренных выше моделей LP-структур ограничено системами с *монотонным* выводом, то есть такими системами продукционного типа, логический вывод в которых соответствует монотонному накоплению информации. Однако существуют практические задачи искусственного интеллекта, которые предполагают не только накопление, но и модификацию получаемых знаний.

В данной главе рассматриваются две новые задачи, формализация и исследование которых могут быть выполнены на основе LP-структур с немонотонным выводом.

5.1. LP-структуры на решетках типов

В этом разделе вводится и исследуется немонотонная LP-структура, содержащая семантику продукционно-логического вывода на иерархии типов в объектно-ориентированной системе с дополнительным отношением. Исследуется стандартный набор свойств таких структур – замкнутость, возможность эквивалентных преобразований, существование логической редукции. Показано, как изложенная теория может быть применена для верификации и автоматической оптимизации иерархий типов, в частности, при рефакторинге – модернизации устаревшего кода. Одним из важных направлений в этом плане является устранение дублирования кода путем «подъема» общих атрибутов по иерархии типов. Такая задача решается автоматически при построении логической редукции LP-структуры.

Заключительный подраздел посвящен оценкам сложности построения замыкания и редукции LP-структур на решетках типов. Показано, что эти задачи в рассматриваемой модели решаются за полиномиальное время.

5.1.1. Логическое замыкание и эквивалентные преобразования

В разделе 1.7 были изложены общие идеи формализованного представления иерархии типов на основе LP-структуры. Далее будет дано определение LP-структуры с учетом общих соображений раздела 1.7.

Вначале сформулируем некоторые понятия и условия, связанные с ограничением свойства $\vee$-дистрибутивности продукционно-логических отношений. Как отмечалось в п. 1.7, данное свойство описывает «поднятие» общих атрибутов по решетке типов. В этом случае трудности возникают при попытке ввести статическое формальное условие для описания динамического процесса. Более точно – условие $\vee$-дистрибутивности отношения R на любых подходящих парах (b_1, a), $(b_2, a) \in R$ должно быть выполнено как до «подъема» атрибута типа a, так и после него. В дальнейшем при ссылках на примеры для краткости будут использоваться фразы «подъем атрибута a», или просто «подъем a», если это не вызовет противоречий. Такое тем более возможно потому, что формальные определения данного раздела связаны не с типами, а с абстрактными элементами решетки.

Пара (b, a) элементов ограниченной решетки $\mathbb{F}$ называется простой, если $a \wedge b = O$ – нижняя грань $\mathbb{F}$.

Определение 1. Пусть R – отношение на ограниченной решетке $\mathbb{F}$. Две пары вида (b_1, a), $(b_2, a) \in R$ называются $\vee$-совместимыми в R, если существуют такие $c_1, c_2 \in \mathbb{F}$, что $b_1 \vee b_2 \leq c_1 \vee c_2$, причем пара $(c_1 \vee c_2, a)$ простая, а пары (c_1, a), $(c_2, a) \in R$ нетранзитивны в $R \bigcup \leq$. При этом набор элементов $C = (b_1, b_2, c_1, c_2, a)$ будем называть $\vee$-дистрибутивным кортежем, а $T = (c_1, c_2, a)$ – $\vee$-дистрибутивной тройкой (в R).

Поясним это определение на примере. Предположим, что типы b_1 и b_2 содержат атрибут a и пара $(b_1 \vee b_2, a)$ является простой. Тогда, в соответствии с рассуждениями, представленными в п.1.7, общий атрибут может быть «поднят», то есть перемещен в тип $b_1 \vee b_2$. До этого действия имеем $c_1 = b_1, c_2 = b_2$. После «подъема» атрибута условие определения 1 остается выполненным, однако с другими промежуточными элементами, а именно $c_1 = c_2 = b_1 \vee b_2$.

Указанная в определении 1 нетранзитивность пар (c_1, a), (c_2, a) обеспечивает «подъем» непосредственного атрибута типов b_1 и b_2, а не косвенного, связанного с ним цепочкой наследований и агрегаций. Конечно, «поднимаемый» атрибут неявно перемещается вместе со своими «возможностями», но именно он сам должен удовлетворять условиям определения 1.

Следующее понятие формально описывает возможные варианты конфликтов между кортежами (точнее – их тройками), претендующими на свойство $\vee$-дистрибутивности (см. также о конфликте пар в п.1.7).

Определение 2. Пусть имеются $\vee$-дистрибутивная тройка $T = (c_1, c_2, a)$, а также тройка $T' = (c_1', c_2', a')$, проверяемая на аналогичное свойство. Тройку T будем называть нейтрализующей для T' (обозначим $T' \prec T$), если выполнено одно из следующих условий:

1) $a = a'$, $c_1 \vee c_2 \neq c_1' \vee c_2'$ и справедливо хотя бы одно из неравенств $c_i' < c_1 \vee c_2$;
2) $a < a'$, $c_1 \vee c_2 \neq c_1' \vee c_2'$ и выполнено хотя бы одно из неравенств $c_i' \leq c_1 \vee c_2$;
3) $a < a'$, $c_1 \vee c_2 = c_1' \vee c_2'$.

Интерпретируемые с неформальных позиций условия определения 2 описывают ситуации, когда наличие нейтрализующей тройки T «угрожает» свойству $\vee$-дистрибутивности тройки T'. Рассмотрим в этом плане перечисленные выше условия 1)–3).

Пусть имеет место условие 1). Тогда, после возможного «подъема» a из c_1 и c_2 в $c_1 \vee c_2$, в силу соотношений $c_i' < c_1 \vee c_2$, $(c_1 \vee c_2, a) \in R$, $a = a'$ пара (c_i', a') окажется транзитивной в $R \bigcup \leq$, что сделает невозможной $\vee$-дистрибутивность тройки T'. Условие 1), в частности, содержит вариант конфликта, который отмечался в примере раздела 1.7. Для этого примера справедливы оба соотношения $T' \prec T$ и $T \prec T'$.

Если для T, T' выполнен вариант 2), то после «подъема» a в $c_1 \vee c_2$ получим соотношения $c_i' \leq c_1 \vee c_2$, $(c_1 \vee c_2, a) \in R$, $a < a'$, что вновь означает транзитивность в $R \bigcup \leq$ пары (c_i', a'). В случае 3) после «подъема» атрибута a в $c_1 \vee c_2 = c_1' \vee c_2'$ получим, что обе пары (c_i', a') транзитивны в $R \bigcup \leq$: $c_i' \leq c_1' \vee c_2'$, $(c_1' \vee c_2', a) \in R$, $a < a'$.

Тройку $T' = (c_1', c_2', a')$ будем называть неконфликтной, если для нее не существует ни одной нейтрализующей $\vee$-дистрибутивной тройки.

Понятия, введенные выше для троек вида $T = (c_1, c_2, a)$ и $T' = (c_1', c_2', a')$, автоматически распространяются и на связанные с ними кортежи $C = (b_1, b_2, c_1, c_2, a)$ и $C' = (b_1', b_2', c_1', c_2', a')$.

Две $\vee$-совместимые пары (b_1, a), (b_2, a) называются неконфликтно $\vee$-совместимыми, если для них существует неконфликтный $\vee$-дистрибутивный кортеж (b_1, b_2, c_1, c_2, a).

Отношение R на решетке называется ограниченно $\vee$-дистрибутивным, если для любых неконфликтно $\vee$-совместимых в R пар (b_1, a), (b_2, a) справедливо $(b_1 \vee b_2, a) \in R$.

Определение 3. Отношение R на ограниченной решетке называется логическим с ограничением объединений (в данном разделе – просто логическим), если оно содержит $\le$, транзитивно и ограниченно $\vee$-дистрибутивно. Логическим замыканием отношения R называется наименьшее логическое отношение, содержащее R и его множество неконфликтно совместимых пар.

Два отношения R_1 и R_2, определенные на общей решетке, называются эквивалентными ($R_1 \sim R_2$), если их логические замыкания совпадают. Логической редукцией отношения R называется эквивалентное ему минимальное (в смысле теории множеств) отношение R_0.

Заметим, что при этом не требуется выполнения вложения $R_0 \subseteq R$. В принципе можно говорить о некотором отношении R_0, как логической редукции вообще, не относя этот факт к какому-либо другому отношению R. Это означает, что при изъятии любой пары меньшее отношение не будет эквивалентно R_0.

Для выяснения вопроса о существовании логического замыкания и его редукции введем следующее определение логической связи.

Определение 4. Пусть задано произвольное отношение R на решетке $\mathbb{F}$. Будем констатировать, что упорядоченная пара $b, a \in \mathbb{F}$ логически связана отношением R ($b \xleftarrow{R} a$), если выполнено одно из следующих условий:

1) $(b,a) \in R$;
2) $b \le a$;
3) существуют такие $b_1, b_2 \in \mathbb{F}$, что $b_1 \vee b_2 = b$, причем $b_1 \xleftarrow{R} a, b_2 \xleftarrow{R} a$ и пары $(b_1,a),(b_2,a)$ неконфликтно $\vee$-совместимы;
4) существует элемент $c \in \mathbb{F}$ такой, что $b \xleftarrow{R} c$ и $c \xleftarrow{R} a$.

Условия 1)–4) определения 4 будем также именовать правилами вывода. При получении некоторой логической связи на основе определения 4 шагом вывода будем называть применение ровно одного правила вывода из этого определения, возможно, одновременно к некоторому конечному множеству элементов решетки. Например, если $b_t \xleftarrow{R} c_t, c_t \xleftarrow{R} a_t$, $t \in T$, то $b_t \xleftarrow{R} a_t$, $t \in T$.

Уровнем рекурсии в соотношении $b \xleftarrow{R} a$ будем называть количество шагов вывода, необходимое для получения этой связи. При этом учитываются лишь применения правил 3)–4). Для связи, основанной только на правиле 1) или 2), уровень рекурсии считается равным нулю. Поскольку в общем случае связь $b \xleftarrow{R} a$ может быть получена не единственным набором правил, будем, как правило, лишь оценивать ее уровень рекурсии, не указывая его точного значения.

Применительно к шагам вывода соотношения $b \xleftarrow{R} a$ можно употреблять слова «начальный», «последний», «предыдущий», «следующий» и так далее. При этом имеется в виду продвижение в направлении прямого логического вывода, то есть от пар исходного отношения ($R \bigcup \le$) к рассматриваемой паре отношения $\xleftarrow{R}$. Заметим, что рекурсивное определение 4 сформулировано в соответствии с принципом обратного вывода.

5.1.2. Свойства дистрибутивных троек и совместимых пар

Для дальнейших исследований необходимо установить ряд свойств $\vee$-дистрибутивных троек и $\vee$-совместимых пар.

Лемма 1. Пусть $T = (c_1, c_2, a)$ – неконфликтная $\vee$-дистрибутивная тройка и элемент $c_3 \in \mathbb{F}$ таков, что пары $(c_1 \vee c_3, a), (c_2 \vee c_3, a)$ простые, а пара $(c_3, a) \in R$ нетранзитивна в $R \bigcup \le$. Тогда $c_1 \vee c_2 \vee c_3 = c_1 \vee c_2 = c_2 \vee c_3 = c_1 \vee c_3$.

Доказательство. Сразу заметим, что интерес представляет лишь случай, когда $c_3 \ne c_1$, $c_3 \ne c_2$, иначе утверждение леммы тривиально. Рассмотрим тройку $T' = (c_2, c_3, a)$. В силу условия леммы она $\vee$-дистрибутивна. Кроме того, поскольку $c_2 \le c_2 \vee c_3$ и пары (c_2, a), (c_3, a) нетранзитивны в $R \bigcup \le$, то $c_2 < c_2 \vee c_3$. Последнее неравенство означает, что тройка T' может оказаться нейтрализующей для $T = (c_1, c_2, a)$. Чтобы этого не случилось (ведь по условию T неконфликтна!) в силу условия 1) определения 2 необходимо выполнение равенства $c_1 \vee c_2 = c_2 \vee c_3$, из которого сразу следуют соотношения $c_1 \vee c_2 \vee c_3 = c_1 \vee c_2 = c_2 \vee c_3$. Недостающее равенство $c_1 \vee c_2 \vee c_3 = c_1 \vee c_3$ можно получить, если вместо T' аналогично рассмотреть тройку (c_1, c_3, a). □

Лемма 2. Пусть R – произвольное бинарное отношение на $\mathbb{F}$ и $T = (c_1, c_2, a)$ – неконфликтная $\vee$-дистрибутивная тройка. Тогда множество неконфликтно $\vee$-совместимых пар отношения $R \bigcup \{(c_1 \vee c_2, a)\}$ совпадает с множеством неконфликтно $\vee$-совместимых пар исходного отношения R.

Доказательство. Вначале докажем, что в результате указанной операции, а именно – добавления $(c_1 \vee c_2, a)$ к R, исходное множество не сужается. Из определения 5.1.1.1[1] следует, что множество $\vee$-совместимых пар полностью определяется множеством нетранзитивных пар отношения $R \bigcup \le$. Таким образом, при доказательстве достаточно рассмотреть те пары нового отношения $R \bigcup \{(c_1 \vee c_2, a)\}$,

[1] Определение 1 из п. 5.1.1.

нетранзитивность которых может нарушиться в результате присоединения к R пары $(c_1 \vee c_2, a)$ (например, (c', a') при $c' \le c_1 \vee c_2$ и $a \le a'$).

Покажем, что любые неконфликтно $\vee$-совместимые пары $(d_1, a'), (d_2, a')$ останутся таковыми после добавления к R указанной в лемме пары $(c_1 \vee c_2, a)$. Соответствующий парам $(d_1, a'), (d_2, a')$ неконфликтный $\vee$-дистрибутивный кортеж обозначим $C' = (d_1, d_2, c_1', c_2', a')$. Согласно определению 5.1.1.1, справедливо $d_1 \vee d_2 \le c_1' \vee c_2'$, пара $(c_1' \vee c_2', a)$ простая и пары $(c_1', a'), (c_2', a') \in R$ нетранзитивны в $R \bigcup \le$. Их нетранзитивность может нарушиться в результате добавления пары $(c_1 \vee c_2, a)$ к отношению R. Покажем, как в этом случае заменить соответствующие элементы c_1', c_2' на другие $\tilde{c}_1, \tilde{c}_2$, чтобы для пар $(d_1, a'), (d_2, a')$ вновь выполнялись все требования определения 5.1.1.1.

Рассмотрим вначале случай $c_1 \vee c_2 = c_1' \vee c_2'$. Поскольку обе тройки $T = (c_1, c_2, a)$ и $T' = (c_1', c_2', a')$ неконфликтны, то согласно условию 3) определения 5.1.1.2 имеем $a = a'$. Пусть при этом $c_i' < c_1 \vee c_2$ ($i = 1$ или $i = 2$). Тогда после добавления $(c_1 \vee c_2, a)$ пара (c_i', a') становится транзитивной в $R \bigcup \le$. Обозначив $\tilde{c}_i = c_1 \vee c_2$, заметим, что пара $(\tilde{c}_i, a')$ нетранзитивна в $R \bigcup \le$. Если же неравенство $c_i' < c_1 \vee c_2$ не выполнено, то пара (c_i', a') остается нетранзитивной, и можно выбрать $\tilde{c}_i = c_i'$. Проверим, что выбранные элементы $\tilde{c}_1, \tilde{c}_2$ наряду с нетранзитивностью пар $(\tilde{c}_1, a'), (\tilde{c}_2, a') \in R$ обеспечивают и остальные требования $\vee$-совместимости пар $(d_1, a'), (d_2, a')$.

Во-первых, по построению $\tilde{c}_i$ справедливо $c_i' \le \tilde{c}_i$, отсюда имеем $c_1' \vee c_2' \le \tilde{c}_1 \vee \tilde{c}_2$. Из этого неравенства и соотношения $d_1 \vee d_2 \le c_1' \vee c_2'$ следует $d_1 \vee d_2 \le \tilde{c}_1 \vee \tilde{c}_2$, что и требуется в определении 5.1.1.1. Покажем далее, что $\tilde{c}_1 \vee \tilde{c}_2 = c_1' \vee c_2'$. Поскольку рассматривается вариант $c_1 \vee c_2 = c_1' \vee c_2'$, то из построения $\tilde{c}_i$ ($\tilde{c}_i = c_i'$ либо $\tilde{c}_i = c_1 \vee c_2$) следует, что $\tilde{c}_i \le c_1' \vee c_2'$, $i = 1, 2$. Отсюда вытекает соотношение $\tilde{c}_1 \vee \tilde{c}_2 \le c_1' \vee c_2'$. С другой стороны, как было замечено в начале абзаца, $c_1' \vee c_2' \le \tilde{c}_1 \vee \tilde{c}_2$. В итоге имеем $\tilde{c}_1 \vee \tilde{c}_2 = c_1' \vee c_2'$. Это равенство означает, что пара $(\tilde{c}_1 \vee \tilde{c}_2, a')$ является простой, так как пара $(c_1' \vee c_2', a')$ простая по определению. Таким образом, для элементов $\tilde{c}_1, \tilde{c}_2$ выполнены все требования определения 5.1.1.1.

Рассмотрим теперь вариант $c_1 \vee c_2 \ne c_1' \vee c_2'$ и в нем потенциально опасный случай $a \le a'$. В нем из-за неконфликтности тройки T' в силу условий 1)–2) определения 5.1.1.2 невозможно выполнение неравенств вида $c_i' \le c_1 \vee c_2$. Рассматривая условия 1)–2) отдельно, замечаем, что в каждом из вариантов сохраняется нетранзитивность пар (c_i', a'), $i = 1, 2$, а вместе с ней и $\vee$-совместимость пар $(d_1,a'),(d_2,a')$. Таким образом, при исходном неравенстве $c_1 \vee c_2 \neq c_1' \vee c_2'$ кортеж C' не меняет своих свойств, то есть можно выбрать $\tilde{c}_i = c_i'$, $i=1,2$.

Показано, что после добавления $(c_1 \vee c_2, a)$ к R пары $(d_1,a'),(d_2,a')$ сохраняют свойство $\vee$-совместимости. Для доказательства необходимого вложения множеств осталось проверить, что сохраняется и их свойство неконфликтности. Достаточно показать, что для тройки $\tilde{T}=(\tilde{c}_1,\tilde{c}_2,a')$ нет ни одной нейтрализующей $\vee$-дистрибутивной тройки $N=(n_1,n_2,p)$. С этой целью предположим противное, а именно – что она существует ($\tilde{T} \prec N$), и при этом проанализируем условия 1)–3) определения 5.1.1.2. Учитывая полученные выше соотношения $c_i' \leq \tilde{c}_i$ и $\tilde{c}_1 \vee \tilde{c}_2 = c_1' \vee c_2'$, нетрудно заметить, что для $T'=(c_1',c_2',a')$ также выполнено $T' \prec N$. Если при этом тройка N (вместе со своими свойствами) существовала до добавления пары $(c_1 \vee c_2, a)$ к R, то имеем противоречие, так как тройка T' изначально была неконфликтной. Рассмотрим ситуацию, когда сама тройка (n_1,n_2,p) или ее свойства (пара $(n_1 \vee n_2, p)$ простая, а пары $(n_1,p),(n_2,p) \in R$ нетранзитивны в $R \bigcup \leq$) появились в результате включения $(c_1 \vee c_2, a)$ в отношение R. Очевидно, что свойство нетранзитивности любой пары не может возникнуть в результате добавления к отношению другой пары, равно как и свойство простоты фиксированной пары элементов решетки. Таким образом, единственно возможным вариантом остается лишь появление самой тройки (n_1,n_2,p), когда $p=a$ и $n_j = c_1 \vee c_2$ ($j=1$ либо $j=2$). Положим для определенности $j=1$, то есть $N=(c_1 \vee c_2, n_2, a)$ (альтернативный случай симметричен). Далее заметим, что поскольку пара $(c_1 \vee c_2 \vee n_2, a)$ простая, то для тройки $T=(c_1,c_2,a)$ и пары (n_2,a) выполнены условия леммы 1. В силу этой леммы $c_1 \vee c_2 \vee n_2 = c_1 \vee c_2$. С учетом последнего равенства рассмотрим применительно к $\tilde{T} \prec N$ каждое из условий 1)–3) определения 5.1.1.2 в отдельности.

Пусть $a=a'$, $c_1 \vee c_2 \vee n_2 \neq \tilde{c}_1 \vee \tilde{c}_2$ и справедливо одно из неравенств $\tilde{c}_i < c_1 \vee c_2 \vee n_2$, то есть имеет место условие 1). Тогда выполнены и соотношения $c_i' < c_1 \vee c_2 \vee n_2 = c_1 \vee c_2$. Кроме того, в рассматриваемом варианте $c_1 \vee c_2 \neq \tilde{c}_1 \vee \tilde{c}_2 = c_i' \vee c_2'$. Получено, что изначально тройка $T=(c_1,c_2,a)$ была нейтрализующей для тройки $T'=(c_1',c_2',a')$, что противоречит условию неконфликтности T'.

Рассмотрим следующий вариант, когда тройка $N=(c_1 \vee c_2, n_2, a)$ является нейтрализующей для $\tilde{T}$ в силу условия 2). Согласно этому условию, $a<a'$, $c_1 \vee c_2 \vee n_2 \neq \tilde{c}_1 \vee \tilde{c}_2$ и выполнено хотя бы одно из неравенств $\tilde{c}_i \leq c_1 \vee c_2 \vee n_2$. Тогда, как и в предыдущем случае,

справедливо $c_i' \le c_1 \vee c_2$ и $c_1 \vee c_2 \ne c_1' \vee c_2'$, то есть $T' \prec T$ и имеем аналогичное противоречие.

Наконец, если $a < a'$, $c_1 \vee c_2 \vee n_2 = \tilde{c}_1 \vee \tilde{c}_2$, то и $c_1 \vee c_2 = c_1' \vee c_2'$. И в данном случае оказывается, что $T' \prec T$ было выполнено до добавления $(c_1 \vee c_2, a)$ к R, чего не могло быть по условию леммы.

Полученные противоречия показывают, что в результате добавления $(c_1 \vee c_2, a)$ к R произвольные $\vee$-совместимые пары $(d_1, a'),(d_2, a')$ сохраняют свое свойство неконфликтности. Таким образом, доказано, что в результате данной операции исходное множество неконфликтно $\vee$-совместимых пар не сужается. Для полного доказательства леммы 2 осталось установить, что оно и не расширяется. С этой целью вначале покажем, что расширение было бы возможным лишь за счет единственной новой $\vee$-дистрибутивной тройки $T_\vee = (c_1 \vee c_2, c_1 \vee c_2, a)$. Предположим противное, то есть что после добавления $(c_1 \vee c_2, a)$ в R появились новые неконфликтно $\vee$-совместимые пары $(d_1, a'),(d_2, a')$ с соответствующей им тройкой $T' = (c_1', c_2', a')$, не совпадающей с $T_\vee$. Она не могла существовать до добавления $(c_1 \vee c_2, a)$ к R. При этом, как и выше, единственно возможным вариантом является лишь появление самой тройки (c_1', c_2', a'), когда $a' = a$ и $c_j' = c_1 \vee c_2$ ($j = 1$ или $j = 2$). Снова не ограничивая общности можно считать, что $j = 1$, то есть $T' = (c_1 \vee c_2, c_2', a)$. Кроме того, в силу сделанного предположения имеем $c_1 \vee c_2 \ne c_2'$. Так как пара $(c_1 \vee c_2 \vee c_2', a)$ простая, то для тройки $T = (c_1, c_2, a)$ и пары (c_2', a) выполнены условия леммы 1, по которой имеем $c_1 \vee c_2 \vee c_2' = c_1 \vee c_2$, то есть $c_2' < c_1 \vee c_2$. Последнее неравенство приводит к противоречию, так как из него следует транзитивность пары (c_2', a) в $R \cup \le$, чего не может быть по определению дистрибутивной тройки T'.

Таким образом, в результате описанной в лемме операции появилась единственная новая $\vee$-дистрибутивная тройка $T_\vee = (c_1 \vee c_2, c_1 \vee c_2, a)$. Однако, ее появление не порождает новых дистрибутивно совместимых пар $(b_1, a),(b_2, a)$, поскольку неравенство $b_1 \vee b_2 \le (c_1 \vee c_2) \vee (c_1 \vee c_2) = c_1 \vee c_2$ было бы выполнено и ранее для тройки $T = (c_1, c_2, a)$.

Итак, лемма 2 полностью доказана. □

Замечание 1. Пользуясь данными выше определениями 1–3, легко показать, что множество $\vee$-совместимых пар (с их конфликтами) определяется множеством нетранзитивных пар отношения $R \cup \le$. Отсюда следует инвариантность множества неконфликтно $\vee$-совместимых пар отношения R относительно еще одной операции – расширения R за счет транзитивных пар отношения $R \cup \le$.

Лемма 3. Множество неконфликтно $\vee$-совместимых пар отношения $\xleftarrow{R}$ совпадает с таковым множеством базового отношения R.

Доказательство. Достаточно установить, что при построении отношения $\xleftarrow{R}$ из отношения R за счет правил 2)–4) определения 5.1.1.4 указанное в лемме множество сохраняется. Для правил 2) и 4) подлежащее доказательству утверждение сразу следует из замечания 1. Покажем, что аналогичное верно и для правила 3).

Пусть пары $(b_1,a),(b_2,a)$ неконфликтно $\vee$-совместимы. Рассмотрим соответствующий им неконфликтный кортеж $C=(b_1,b_2,c_1,c_2,a)$. Как отмечалось выше, пары $(c_1,a),(c_2,a)$ также неконфликтно $\vee$-совместимы, поэтому отношение $\xleftarrow{R}$ должно содержать и пару $(c_1 \vee c_2,a)$. При этом, по лемме 2, операция добавления к R пары $(c_1 \vee c_2,a)$ сохраняет его множество неконфликтно совместимых пар.

Рассматривая далее пары $(b_1 \vee b_2, c_1 \vee c_2),(c_1 \vee c_2,a)$, замечаем, что пара $(b_1 \vee b_2,a)$ транзитивна в $R \bigcup \leq$ при новом (расширенном) отношении R. Эта пара также должна быть добавлена к R по правилам 2) и 4). При этом, как указано в начале доказательства, исходное множество неконфликтно совместимых пар расширяемого отношения R не изменится. □

5.1.3. Логическое замыкание и эквивалентные преобразования

В настоящем разделе исследуются вопросы о существовании и архитектуре логического замыкания рассматриваемой LP-структуры, а также ее эквивалентных преобразованиях.

Лемма 1. При произвольном R отношение $\xleftarrow{R}$ является логическим.

Доказательство. Необходимо проверить, что данное отношение содержит $\leq$, транзитивно и ограниченно $\vee$-дистрибутивно. Первые два указанных свойства непосредственно следуют из п. 2) и п. 4) определения 5.1.1.4[1]. Свойство ограниченной $\vee$-дистрибутивности требует более подробного рассмотрения.

Пусть $b_1,b_2,a \in \mathbb{F}$, $b_1 \xleftarrow{R} a, b_2 \xleftarrow{R} a$ и пары $(b_1,a),(b_2,a)$ неконфликтно $\vee$-совместимы в $\xleftarrow{R}$. Необходимо доказать, что тогда $b_1 \vee b_2 \xleftarrow{R} a$. С этой целью покажем, что при сделанных предположениях пары $(b_1,a),(b_2,a)$ неконфликтно $\vee$-совместимы в R. Тогда подлежащее доказательству утверждение сразу будет следовать из п. 3) определения 5.1.1.4.

[1] Определение 4 из п. 5.1.1.

Итак, согласно определению 5.1.1.1 имеем: существуют такие $c_1, c_2 \in \mathbb{F}$, что $b_1 \vee b_2 \le c_1 \vee c_2$, пара $(c_1 \vee c_2, a)$ простая, $c_1 \xleftarrow{R} a$, $c_2 \xleftarrow{R} a$ и пары $(c_1, a), (c_2, a)$ нетранзитивны в $\xleftarrow{R}$ (поскольку $(R \bigcup \le) \subseteq \xleftarrow{R}$, то и в $R \bigcup \le$). Последнее означает, что соотношения $c_1 \xleftarrow{R} a, c_2 \xleftarrow{R} a$ были получены без применения п. 4) определения 5.1.1.4. Простота пары $(c_1 \vee c_2, a)$ означает также, что не использовался и п.2). Таким образом, из п. 1) и п. 3) определения 5.1.1.4 следует, что для пар $(c_1, a), (c_2, a)$ остается лишь такой вариант, когда $c_1 = c_1^1 \vee ... \vee c_1^{m_1}$, $c_2 = c_2^1 \vee ... \vee c_2^{m_2}$, где $(c_j^{k_j}, a) \in R$, $k_j = 1, ..., m_j$, $j = 1, 2$, причем все пары $(c_j^{k_j}, a)$ нетранзитивны в $R \bigcup \le$.

Рассмотрим вначале случай $m_1 = m_2 = 1$. Тогда $(c_1, a), (c_2, a) \in R$, и лемма уже доказана (неконфликтность тройки $T = (c_1, c_2, a)$ в R следует из ее неконфликтности в большем множестве $\xleftarrow{R}$).

Предположим теперь, что $m_1 > 1$. В этом случае дистрибутивная тройка $T' = (c_1^1, c_1^2, a)$ является неконфликтной в R. Поскольку пара $(c_1 \vee c_2, a)$ простая, то для тройки T и любой оставшейся пары вида $(c_j^{k_j}, a)$ выполнены условия леммы 5.1.2.1. По ее утверждению справедливо равенство $c_1^1 \vee c_1^2 \vee c_j^{k_j} = c_1^1 \vee c_1^2$. Складывая (в смысле операции объединения $\vee$) такие равенства по всем $k_j = 1, ..., m_j$, $j = 1, 2$, получим $c_1 \vee c_2 = c_1^1 \vee c_1^2$. В итоге приходим к следующему факту: существуют такие $c_1^1, c_1^2 \in \mathbb{F}$, что $b_1 \vee b_2 \le c_1^1 \vee c_1^2$, пара $(c_1^1 \vee c_1^2, a)$ простая, пары $(c_1^1, a), (c_1^2, a) \in R$ нетранзитивны в $R \bigcup \le$, тройка $T' = (c_1^1, c_1^2, a)$ неконфликтна в R. Он означает, что пары $(b_1, a), (b_2, a)$ неконфликтно $\vee$-совместимы в R. Отсюда в силу п. 3) определения 5.1.1.4 следует $b_1 \vee b_2 \xleftarrow{R} a$, то есть отношение $\xleftarrow{R}$ ограниченно $\vee$-дистрибутивно. □

Лемма 2. Пусть R – некоторое логическое отношение на решетке $\mathbb{F}$ и $a, b \in \mathbb{F}$. Тогда, если справедливо $b \xleftarrow{R} a$, то $(b, a) \in R$.

Доказательство. Проведем его с помощью индукции по m – оценке уровня рекурсии в соотношении $b \xleftarrow{R} a$. При $m = 0$ имеет место одно из условий 1)–2) определения 5.1.1.4. Случай 1) непосредственно означает справедливость доказываемого утверждения. Если же справедливо 2), то и тогда $(b, a) \in R$, поскольку логическое отношение R содержит и $\le$.

Предположим далее, что лемма верна для некоторого $m \ge 0$ и докажем ее утверждение при уровне рекурсии $m + 1$. В этом случае новые для рассмотрения варианты могут дать условия 3)-4) определения 5.1.1.4.

Рассмотрим вариант, когда соотношение $b \xleftarrow{R} a$ происходит из условия 3) определения 5.1.1.4. При этом по предположению индукции соотношения $b_1 \xleftarrow{R} a, b_2 \xleftarrow{R} a$ имеют уровень рекурсии $\le m$, то есть $(b_1, a), (b_2, a) \in R$. Тогда, в силу ограниченной дистрибутивности R, получим $(b, a) \in R$.

Наконец, пусть справедливо 4). И в этом случае по предположению индукции базовые соотношения $b \xleftarrow{R} c$ и $c \xleftarrow{R} a$ имеют уровень рекурсии $\le m$. Следовательно, $(b, c), (c, a) \in R$. Отсюда, в силу транзитивности логического отношения R, имеем $(b, a) \in R$. □

Теорема 1. Для произвольного отношения R на решетке $\mathbb{F}$ логическое замыкание существует и совпадает с множеством $\xleftarrow{R}$ всех упорядоченных пар, логически связанных отношением R.

Доказательство. В силу п.1) определения 5.1.1.4 отношение $\xleftarrow{R}$ содержит R. По лемме 5.1.2.3 оно также содержит и множество неконфликтно $\vee$-совместимых в R пар. Из леммы 1 следует, что $\xleftarrow{R}$ является логическим. Осталось показать, что это – наименьшее из таковых отношений.

Пусть R' – другое логическое отношение, содержащее R и его множество неконфликтно совместимых пар. Тогда очевидно, что если $b \xleftarrow{R} a$, то $b \xleftarrow{R'} a$. Отсюда по лемме 2 имеем $(b, a) \in R'$. Следовательно, отношение $\xleftarrow{R}$ содержится в произвольно выбранном R', и, таким образом, является наименьшим логическим отношением, содержащим R и его множество неконфликтно совместимых пар. □

Далее рассмотрим вопросы, связанные с эквивалентными преобразованиями логических структур на решетках типов. Пусть дано отношение R на решетке $\mathbb{F}$. Его эквивалентным преобразованием называется такая замена множества упорядоченных пар R, в результате которой полученное новое отношение P эквивалентно R.

В главах 2–4 для логических отношений с монотонным выводом показано, что локально-эквивалентное преобразование части исходного множества R приводит к логически эквивалентному общему отношению P. Для рассматриваемых в настоящем разделе отношений это в общем неверно. Например, если при $T' \prec T$ преобразовать (эквивалентно) отдельный набор пар из T', общий результат окажется неэквивалентным. Однако имеет место эквивалентность ряда преобразований.

Теорема 2. Пусть R – отношение на решетке $\mathbb{F}$. Тогда каждая из следующих операций над R приводит к эквивалентному отношению:

• добавление или исключение пары (b, a), если $b \le a$;

• добавление пары $(c_1 \vee c_2, a)$, если тройка $T = (c_1, c_2, a)$ $\vee$-дистрибутивна и неконфликтна в R;

• добавление или исключение пары (c,a) при наличии пар $(c,b),(b,a) \in (R \cup \leq)$, $c \neq b$, $b \neq a$.

Доказательство следует непосредственно из определения 5.1.1.4 и доказанных выше лемм.

5.1.4. Структура логических связей и редукция

В главах 2–4 для каждой структуры с монотонным выводом показано, что логическое замыкание произвольного бинарного отношения R совпадает с транзитивным замыканием другого отношения $\tilde{R} \supseteq R$, построенного в виде некоторого «дистрибутивного многообразия» R. В рассматриваемом случае, основанном на ограниченной $\vee$-дистрибутивности, также удается разделить процесс построения логического замыкания на этапы дистрибутивного и транзитивного замыканий.

Для отношения R на решетке $\mathbb{F}$ рассмотрим отношение $\tilde{R}$, в результате последовательного выполнения следующих двух шагов:

• для каждой неконфликтной $\vee$-дистрибутивной тройки (c_1, c_2, a) $(c_1 \neq c_2)$ добавить к исходному отношению пару $(c_1 \vee c_2, a)$;

• к полученному отношению добавить отношение $\leq$.

Заметим, что по теореме 5.1.3.2 отношение $\tilde{R}$ эквивалентно R.

Теорема 1. Логическое замыкание отношения R совпадает с транзитивным замыканием $\tilde{R}^*$ соответствующего отношения $\tilde{R}$.

Доказательство. Пусть $b \xleftarrow{R} a$. Покажем, что при этом $(b,a) \in \tilde{R}^*$, что в свою очередь означает вложение $\xleftarrow{R} \subseteq \tilde{R}^*$. Воспользуемся индукцией по m – уровню рекурсии в соотношении $b \xleftarrow{R} a$. При $m = 0$ имеет место одно из условий 1)–2) определения 5.1.1.4. Очевидно, что $\tilde{R}^*$ содержит отношения R и $\leq$.

Предположим теперь, что доказываемое вложение верно для некоторого $m \geq 0$. Докажем его справедливость при уровне рекурсии $m+1$. В этом случае новые для рассмотрения варианты могут появиться из правил 3)–4) определения 5.1.1.4[1].

Рассмотрим ситуацию, когда соотношение $b \xleftarrow{R} a$ получено из условия 3). В этом случае существуют неконфликтно $\vee$-совместимые в R пары $(b_1, a),(b_2, a)$ с кортежем (b_1, b_2, c_1, c_2, a), причем $b = b_1 \vee b_2$. Тогда при построении $\tilde{R}$ на первом шаге к исходному отношению должна быть добавлена пара $(c_1 \vee c_2, a)$, на втором – пара $(b_1 \vee b_2, c_1 \vee c_2)$. Наконец, при нахождении транзитивного замыкания $\tilde{R}^*$ появляется также пара $(b_1 \vee b_2, a)$. Таким образом, $(b,a) \in \tilde{R}^*$.

Предположим теперь, что $b \xleftarrow{R} a$ произошло из правила 4). В этом случае по предположению индукции базовые связи $b \xleftarrow{R} c$ и $c \xleftarrow{R} a$ имеют уровень рекурсии $\leq m$, то есть $(b,c),(c,a) \in \tilde{R}^*$.

[1] Определение 4 из п. 5.1.1.

Поскольку $\tilde{R}^*$ – транзитивное замыкание, то оно содержит все свои транзитивные пары, то есть $(b,a) \in \tilde{R}^*$.

Таким образом, доказана справедливость вложения $\xleftarrow{R} \subseteq \tilde{R}^*$. Покажем обратное. Пусть $(b,a) \in \tilde{R}^*$. Тогда, в силу свойств транзитивного замыкания, существует такой набор элементов $d_0, d_1, \ldots, d_n$, что $(d_{j-1}, d_j) \in \tilde{R}, j = 1, \ldots, n$, причем $d_0 = b, d_n = a$. Если при этом $(d_{j-1}, d_j) \in R$ либо $d_{j-1} \le d_j$, то по условиям 1)–2) определения 5.1.1.4 имеем $d_{j-1} \xleftarrow{R} d_j$. Рассмотрим пару (d_{j-1}, d_j), для которой свойства 1)–2) не выполнены. Тогда возможен лишь случай, когда пара $(d_{j-1}, d_j) \in \tilde{R}$ имеет вид $(c_1 \vee c_2, a)$, соответствующий некоторой неконфликтной дистрибутивной тройке $T = (c_1, c_2, a)$ отношения R (см. построение $\tilde{R}$). Как следствие, по теореме 5.1.3.2 и в этом случае $d_{j-1} \xleftarrow{R} d_j$. Таким образом, установлено, что для пары $(b,a) \in \tilde{R}^*$ существует набор $d_0, d_1, \ldots, d_n (d_0 = b, d_n = a)$, для которого справедливо $d_{j-1} \xleftarrow{R} d_j, j = 1, \ldots, n$. Применяя к его элементам последовательно (например, слева направо) n раз правило 4) определения 5.1.1.4, получим $b \xleftarrow{R} a$. □

Далее докажем утверждения, которые необходимы для изучения вопросов о существовании и построении логической редукции отношений на ограниченной решетке.

Лемма 1. Пусть R – бинарное отношение на решетке $\mathbb{F}$ и $b_t \xleftarrow{R} a_t \ \forall t \in T$. Тогда отношение $R' = R \bigcup \{(b_t, a_t) \mid t \in T\}$ эквивалентно R.

Доказательство. Заметим вначале, что по определению любое логическое отношение является собственным логическим замыканием. В силу теоремы 5.1.3.1 это относится и к отношению $\xleftarrow{R}$. Далее, из определения 5.1.1.4 очевидным образом следует, что для любых $R_1 \subseteq R_2$ справедливо $\xleftarrow{R_1} \subseteq \xleftarrow{R_2}$. В рассматриваемом случае по построению отношения R' выполнены включения $R \subseteq R' \subseteq \xleftarrow{R}$. Переходя к логическим замыканиям и учитывая изложенное выше, получаем, что отношения R и R' имеют общее логическое замыкание $\xleftarrow{R}$. □

Лемма 2. Пусть R – бинарное отношение на решетке $\mathbb{F}$. Для того, чтобы R являлось логической редукцией, необходимо и достаточно, чтобы R не содержало ни одной такой пары (b,a), для которой выполнено соотношение $b \xleftarrow{R \setminus \{(b,a)\}} a$.

Доказательство. Пусть отношение R представляет собой логическую редукцию, то есть является минимальным логически эквивалентным себе отношением. Предположим противное, а именно, что существует пара $(b,a) \in R$, логически связанная отношением $R \setminus \{(b,a)\}$. Если это так, то в силу леммы 1 пару (b,a) можно исключить из R, получив при этом меньшее эквивалентное отношение. Таким образом, при сделанном предположении отношение R не может быть логической редукцией.

Для доказательства обратного утверждения предположим, что не существует ни одной пары $(b,a) \in R$, для которой справедливо $b \xleftarrow{R\setminus\{(b,a)\}} a$. Необходимо доказать, что в этом случае R есть логическая редукция. Вновь предположим противное – пусть существует отношение $R_0 \subset R$, эквивалентное R, и $(b,a) \in R \setminus R_0$. Тогда, поскольку $(b,a) \in R$, в силу эквивалентности рассматриваемых отношений справедливо $b \xleftarrow{R_0} a$. Так как отношение R_0 не содержит пару (b,a), то $R_0 \subseteq R \setminus \{(b,a)\}$, и логическая связь $b \xleftarrow{R_0} a$ противоречит сделанному предположению – таких пар (b,a) в R нет. Полученное противоречие доказывает требуемое утверждение. □

Рассмотрим непосредственно вопрос о существовании и построении логической редукции бинарных отношений.

Теорема 2. Пусть для отношения R построено соответствующее отношение $\tilde{R}$. Тогда, если для $\tilde{R}$ существует транзитивная редукция R^0, то отношение $\underset{\sim}{R}^0$, полученное исключением из R^0 всех пар вида $b \le a$, представляет собой логическую редукцию исходного отношения R.

Доказательство. Из леммы 1 следует, что указанное в теореме отношение $\underset{\sim}{R}^0$ логически эквивалентно R. Осталось показать, что $\underset{\sim}{R}^0$ является логической редукцией вообще. Для этого достаточно проверить выполнение для $\underset{\sim}{R}^0$ условия леммы 2.

Пусть (b,a) – произвольная пара отношения $\underset{\sim}{R}^0$. Необходимо показать, что логическая связь $b \xleftarrow{\underset{\sim}{R}^0\setminus\{(b,a)\}} a$ невозможна. Предположим противное, а именно, что эта связь существует. Тогда в силу леммы 1 отношение $\underset{\sim}{R}^0 \setminus \{(b,a)\}$ эквивалентно $\underset{\sim}{R}^0$. Сразу заметим, что применение правила 1) определения 5.1.1.4 для вывода $b \xleftarrow{\underset{\sim}{R}^0\setminus\{(b,a)\}} a$ невозможно, поскольку пара (b,a) не содержится в множестве $\underset{\sim}{R}^0 \setminus \{(b,a)\}$.

По лемме 5.1.2.3 множество неконфликтно $\vee$-совместимых пар произвольного бинарного отношения инвариантно относительно применения правил вывода 2)–4) определения 5.1.1.4. Следовательно, любая логическая связь может быть получена таким образом, что все необходимые для этого применения $\vee$-дистрибутивного правила 3) будут произведены на начальных шагах вывода, а все применения транзитивного правила 4) – лишь в завершающей стадии этого вывода. Таким образом, для связи $b \xleftarrow{\underset{\sim}{R}^0\setminus\{(b,a)\}} a$ существует цепочка элементов $b = c_0, c_1, \dots, c_n = a$ такая, что выполнены соотношения $c_i \xleftarrow{\underset{\sim}{R}^0\setminus\{(b,a)\}} c_{i-1}$, $i = 1, \dots, n$, при выводе каждого из которых правило 4) не используется. Отсюда следует, что $(c_i, c_{i-1}) \in \tilde{R}$. Таким образом, получили, что при $n > 1$ пара (b,a) оказывается транзитивной в $\tilde{R}$. Следовательно, она не может содержаться в $\underset{\sim}{R}^0$, являющемся подмножеством транзитивной редукции отношения $\tilde{R}$. Это противоречит исходному предположению $(b,a) \in \underset{\sim}{R}^0$.

Остается рассмотреть случай $n = 1$. В этой ситуации имеющаяся логическая связь $b \xleftarrow{\underset{\sim}{R}^0 \setminus \{(b,a)\}} a$ означает, что $(b,a) \in \tilde{R}$. В силу вышеуказанной инвариантности множества неконфликтно $\vee$-совместимых пар для (b,a) остается лишь одна из двух возможностей: применено правило 2) или 3). Если это 2), то при получении $\underset{\sim}{R}^0$ такая пара (b,a) должна быть исключена.

Покажем, наконец, что в случае применения правила 3) данная пара $(b,a) \in \tilde{R}$ будет исключена на этапе вычисления транзитивной редукции $\tilde{R}$. Пусть существовали такие $b_1, b_2 \in \mathbb{F}$, что $b_1 \vee b_2 = b$, причем $b_1 \xleftarrow{R} a, b_2 \xleftarrow{R} a$ и пары $(b_1, a), (b_2, a)$ неконфликтно $\vee$-совместимы. Тогда при построении $\tilde{R}$ к нему была добавлена соответствующая пара $(c_1 \vee c_2, a)$. Поскольку (см. определение 5.1.1.1) $b_1 \vee b_2 \le c_1 \vee c_2$, то пара (b,a) оказывается транзитивной в $\tilde{R}$. В результате снова приходим к противоречию исходному предположению $(b,a) \in \underset{\sim}{R}^0$.

Таким образом, рассмотрены возможные варианты предполагаемого логического вывода $b \xleftarrow{\underset{\sim}{R}^0 \setminus \{(b,a)\}} a$. В результате установлено, что в каждом случае наличие связи $b \xleftarrow{\underset{\sim}{R}^0 \setminus \{(b,a)\}} a$ противоречит факту $(b,a) \in \underset{\sim}{R}^0$. Следовательно, по лемме 2 отношение $\underset{\sim}{R}^0$ представляет собой логическую редукцию. □

5.1.5. Алгоритмические вопросы

В данном разделе кратко рассматриваются некоторые вопросы практической реализации LP-структур на решетках типов и, соответственно, вычислительной сложности построения их логического замыкания и редукции. Настоящая работа в основном посвящена теоретическому обоснованию решения подобных задач. Однако краткое обсуждение алгоритмических вопросов поможет составить общее представление о практической применимости теории LP-структур на решетках типов.

Как уже отмечалось ранее в главе 2, разработка и получение оценок алгоритмов на решетках в общем случае представляется «неблагодарным делом». Например, булеан при количестве точек n состоит из 2^n различных элементов, со всеми вытекающими «алгоритмическими последствиями». Решетка типов обычно не является дистрибутивной, и число ее элементов не так велико. По этой причине в данном случае все же допустимо в качестве основы анализа сложности выбрать величину N – общее число элементов решетки, а также хранить саму решетку и бинарные отношения на ней в виде матриц смежности размера $N \times N$.

Таким образом, в данном разделе можно абстрагироваться от низкоуровневых проблем представления решеток, которые рассматриваются далее в главе 6 настоящей работы.

Будем обращаться к быстрым алгоритмам построения транзитивного замыкания и транзитивной редукции бинарных отношений. Классический алгоритм Уоршолла [83] для графа с N вершинами строит транзитивное замыкание за $O(N^3)$ шагов. Имеется также ряд улучшений данного алгоритма ([61, 70, 82, 87, 201] и другие). Например, в [81, 201] рассматриваются алгоритмы с оценкой сложности $O(N^{\log_2 7})$. Согласно [2], транзитивная редукция вычисляется со сложностью транзитивного замыкания.

Итак, пусть имеется решетка $\mathbb{F}$, представленная матрицей смежности $N \times N$. Построим для нее матрицу достижимости, для чего потребуется не более $O(N^3)$ шагов. С помощью этой матрицы операция $a \le b$ $(a, b \in \mathbb{F})$ вычисляется за время $O(1)$. Операции $a \wedge b$ и $a \vee b$ сводятся к нахождению соответственно максимума и минимума при однократном просмотре решетки, то есть занимают $O(N)$ шагов каждая. Вспомнив, что результаты этих операций также принадлежат решетке, построим еще две матрицы $N \times N$. В первой из них для каждой пары a, b (число пар составляет $O(N^2)$) будет храниться индекс элемента $a \wedge b$, во второй – $a \vee b$. Построение каждой такой матрицы обойдется в $O(N^3)$ шагов, и в дальнейшем с их помощью любую решеточную операцию можно выполнить за время $O(1)$.

Пусть также задано отношение R на решетке $\mathbb{F}$. Как показано в пп. 5.1.3–5.1.4, задачи нахождения логического замыкания и редукции данной LP-структуры связаны с рассмотрением $\vee$-дистрибутивных троек $T = (c_1, c_2, a)$, где пара $(c_1 \vee c_2, a)$ простая, а пары $(c_1, a), (c_2, a) \in R$ нетранзитивны в $R \bigcup \le$. Для избавления от транзитивных пар сразу построим транзитивную редукцию отношения $R \bigcup \le$. Затем исключим из полученной матрицы R все подчиненные пары. Эта процедура займет не более $O(N^3)$ шагов.

На следующем этапе сформируем все $\vee$-дистрибутивные тройки отношения R . С указанной целью будем просматривать решетку (N шагов) и для каждого элемента a по соответствующему столбцу матрицы R находить всевозможные подходящие пары c_1, c_2 ($O(N^2)$ действий). Таким образом, полный набор $\vee$-дистрибутивных троек $\mathbb{T}$ получится за время $O(N^3)$.

Очередной этап алгоритма (построения логического замыкания или редукции) состоит в том, чтобы в полученном множестве $\mathbb{T}$ оставить лишь неконфликтные тройки. Для этого достаточно попарно их сопоставить. Согласно определению 5.1.1.2, сопоставление двух троек осуществляется за время $O(1)$, а количество сопоставляемых пар квадратично зависит от числа троек $|\mathbb{T}|$. Общее число троек, составленных из произвольных элементов решетки, равно $C_N^3 = O(N^3)$ (число сочетаний из N по 3). Может появиться идея об улучшении этой оценки с использованием свойства $\vee$-дистрибутивной тройки $T = (c_1, c_2, a)$. К сожалению, следующий далее пример опровергает такие предположения, и в результате сложность построения множества неконфликтных $\vee$-дистрибутивных троек $\mathbb{T}_0$ составит $O(N^6)$. Полученные здесь оценки также оправдывают хранение множества троек в виде булевского трехмерного массива $N \times N \times N$, который обеспечит, в частности, выполнение любой операции доступа к множеству за время $O(1)$.

Приведем частный случай отношения R на решетке $\mathbb{F}$, при котором количество $\vee$-дистрибутивных троек составит $O(N^3)$. Разобьем решетку на две равночисленные группы элементов $\{a_i, i = 1, \ldots, N/2\}$ и $\{c_j, j = 1, \ldots, N/2\}$. Сформируем отношение $R = \{(c_j, a_i), i = 1, \ldots, N/2; j = 1, \ldots, N/2\}$. Оно ациклично и при соответствующей конфигурации решетки состоит лишь из нетранзитивных пар. Тогда для каждого элемента a_i можно составить $\vee$-дистрибутивные тройки вида $T = (c_{j_1}, c_{j_2}, a_i)$ $(j_1 \neq j_2)$ в количестве $O(C_{N/2}^2)$. Здесь также предполагается, что в рассматриваемой решетке число различных элементов вида $c_{j_1} \vee c_{j_2}$ (они не участвуют в построении троек) является константным. Таким образом, получим $|\mathbb{T}| = \dfrac{N}{2} O(C_{N/2}^2) = O(N^3)$. Потенциально возможным является случай, когда и $|\mathbb{T}_0| = O(N^3)$.

Далее по схеме п. 5.1.3 необходимо построить дистрибутивное замыкание $\tilde{R}$ исходного отношения R. Перебрав с этой целью все множество $\mathbb{T}_0$, потратим соответственно времени $O(N^3)$. Если в качестве исходной стояла задача нахождения логического замыкания, то по теореме 5.1.4.1 остается выполнить транзитивное замыкание отношения $\tilde{R}$, что займет не более $O(N^3)$ шагов. Если же изначально решалась задача нахождения логической редукции, то согласно теореме 5.1.4.2 для $\tilde{R}$ построим транзитивную редукцию R^0 ($O(N^3)$). Отсюда с использованием матрицы $\mathbb{F}$ за время $O(N^2)$ найдем искомый результат.

Таким образом, доказана следующая теорема.

Теорема 1. Задачи нахождения логического замыкания и логической редукции LP-структуры на решетке типов решаются за полиномиальное время, не превышающее $O(N^6)$, где N – общее число элементов решетки.

5.2. LP-структуры на некоммутативных решетках

В данном разделе вводятся *некоммутативные решетки* как обобщение классических решеток. Операции на некоммутативных решетках формализуют не только накопление, но и замещение знаний. Основным результатом здесь является теорема об ассоциативности операции некоммутативного объединения. Далее вводятся и анализируются действующие на этих решетках логические бинарные отношения. Эти отношения могут быть использованы для моделирования немонотонного логического вывода. Доказана основная теорема о существовании логического замыкания отношений на некоммутативных решетках.

Рассмотрен пример приложения теории некоммутативных LP-структур для исследования логических свойств императивных алгоритмов. В частности, на основе LP-структур сопоставляются возможности императивного и логического программирования.

5.2.1. Некоммутативные решетки

Основой данного понятия являются решетки множеств. В этой связи возвращаемся к соответствующей системе обозначений для операций на LP-структурах ($\subseteq$, $\supseteq$, $\bigcap$, $\bigcup$).

Если решетка $\mathbb{F}$ моделирует некоторую область знаний, то для элементов $A, B \in \mathbb{F}$ знание $A \bigcup B \in \mathbb{F}$ может рассматриваться как объединение знаний A и B (точнее – добавление знаний B к знаниям A). При этом по определению решетки $A \bigcup B = B \bigcup A$. Однако в некоторых ситуациях часть знаний B может противоречить части знаний A. Предлагаемое решение состоит в том, чтобы вместо $\bigcup$ ввести на $\mathbb{F}$ другую операцию. Эта операция часть содержимого A будет замещать противоречащей ей частью содержимого B, а остальные части – объединять. Тогда, очевидно, коммутативность операции будет нарушена. При этом знания B (второй операнд) предполагаются более актуальными, чем A. Аналогично можно ввести и такую некоммутативную операцию объединения, при которой более актуальным будет первый операнд.

Пусть $\mathbb{F}$ – полная решетка с относительными дополнениями, O – ее нижняя грань. Зафиксируем некоторый элемент $X \subseteq \mathbb{F}$ и рассмотрим множество всех элементов решетки $\mathbb{F}$, каждый из которых содержит лишь одну точку элемента X или не содержит ни одной. Обозначим это множество $\mathbb{F}_X$. Точки элемента X будем называть несовместимыми точками, а сам X – элементом несовместимых точек. Необходимость введения этих терминов обусловлена способом построения множества $\mathbb{F}_X$.

Если $A \in \mathbb{F}_X$ содержит точку x_A элемента X, то в силу свойств решетки $\mathbb{F}$ справедливо

$$A = A' \bigcup x_A, \tag{1}$$

где $A' \in \mathbb{F}_X$ – относительное дополнение элемента x_A решетки $\mathbb{F}$ в ее замкнутом интервале $[O, A]$ (см. [93]). Очевидно, что A' не содержит точек элемента X. Вообще, для данного элемента $A \in \mathbb{F}_X$ символом A' будем обозначать соответствующий элемент в представлении (1). Всюду в дальнейшем для элементов множества $\mathbb{F}_X$ будем рассматривать представление (1) в обобщенном смысле, полагая $A' = A$, $x_A = O$, если A не содержит точек X.

Элементам множества $\mathbb{F}_X$ будем приписывать определенные *состояния*, связанные с точками X. Если для $A \in \mathbb{F}_X$ справедливо $A \neq A'$, то A всегда имеет состояние x_A (см. (1)). Если же $A' = A$, то такой элемент A в зависимости от ситуации способен находиться в различных состояниях, однако в каждый фиксированный момент – не более чем в одном. Факт нахождения элемента $A \in \mathbb{F}_X$ в состоянии x_A будем обозначать $[A] = x_A$. Если состояние A пусто, то будем обозначать этот факт $[A] = O$. Состояние самого элемента O всегда считается пустым. Для $C = A \bigcap B$, $A, B \in \mathbb{F}_X$ положим $[C] = [A] \bigcap [B]$.

Поскольку $\mathbb{F}$ – решетка с относительными дополнениями, то для любых $A, B \in \mathbb{F}$ можно определить операцию разности: $A \setminus B$ – относительное дополнение элемента $A \bigcap B$ в замкнутом интервале $[O, A]$. Что касается состояния элемента $A \setminus B$ (при $A, B \in \mathbb{F}_X$), то в случае $[A] = [B]$ будем считать, что $[A \setminus B] = O$, иначе – $[A \setminus B] = [A]$.

Далее на множестве $\mathbb{F}_X$ введем операцию $\bigcup_{X,R}$ следующим образом. Пусть $A, B \in \mathbb{F}_X$. Если $A \neq A'$ и $[B] = x_B \neq O$, то $A \bigcup_{X,R} B = A' \bigcup B \bigcup x_B$, в противном случае положим $A \bigcup_{X,R} B = A \bigcup B$. Что касается состояния элемента $C = A \bigcup_{X,R} B$, то по определению будем считать, что оно всегда совпадает с состоянием элемента B, за исключением ситуации, когда состояние B не определено. В этом случае оно переходит к C от A.

Соответственно введем операцию $\bigcup_{X,L}$: $A \bigcup_{X,L} B = B \bigcup_{X,R} A$.

Легко заметить, что множество $\mathbb{F}_X$ инвариантно относительно операций $\bigcup_{X,R}$, $\bigcup_{X,L}$, а также и $\bigcap, \setminus$. Это множество напоминает фактор-пространство в $\mathbb{F}$ с классом эквивалентности X. Важное отличие состоит в том, что точки множества X не вполне эквивалентны. Они замещают друг друга при выполнении операций, однако не отождествляются.

Очевидно, введенные операции некоммутативны. Далее будет показано, что каждая из операций $\bigcup_{X,R}$ и $\bigcup_{X,L}$ является ассоциативной.

Введем оператор $\boldsymbol{\delta}_{X,R}$, определенный на упорядоченных наборах состояний вида $x_1,\dots,x_m$, где $m \geq 0$ и каждый элемент x_i – это точка в X (при этом $[x_i] = x_i$) либо $x_i = O$ (тогда $[x_i] = O$). Итак, $\boldsymbol{\delta}_{X,R}(x_1,\dots,x_m) = x_0$, где x_0 – первый справа непустой элемент набора $x_1,\dots,x_m$ либо O. Аналогично определяется $\boldsymbol{\delta}_{X,L}$, выбирающий первый непустой элемент слева.

Лемма 1. При любом ассоциативном порядке выполнения операций $\bigcup_{X,R}$ справедливо соотношение $x_1 \bigcup_{X,R} x_2 \bigcup_{X,R} \dots \bigcup_{X,R} x_m = \boldsymbol{\delta}_{X,R}(x_1,\dots,x_m)$.

Доказательство. Докажем это утверждение с помощью индукции по $m \geq 0$. При $m \leq 2$ оно следует непосредственно из определений $\bigcup_{X,R}$ и $\boldsymbol{\delta}_{X,R}$. Предположим, что оно верно при любом количестве операндов, не превосходящем некоторого m, и рассмотрим случай $m+1$.

В этой ситуации объединим в произвольные группы наборы смежных операндов исходного выражения так, чтобы каждый операнд попал ровно в одну группу. Для нетривиальной ассоциативности также необходимо, чтобы групп было больше одной, и хотя бы одна из них содержала более одного операнда. Пусть число этих групп окажется равным p. По построению $p \leq m$ и количество операндов в каждой группе также не превосходит m. По этой причине для каждой группы выполнено утверждение леммы, и, соответственно, в группах могут быть получены однозначные результаты вычислений. Обозначим их $y_1,\dots,y_p$, а вместо $x_1 \bigcup_{X,R} \dots \bigcup_{X,R} x_{m+1}$ рассмотрим выражение $y_1 \bigcup_{X,R} \dots \bigcup_{X,R} y_p$, где $p \leq m$. Для него также выполнено предположение индукции, то есть $y_1 \bigcup_{X,R} \dots \bigcup_{X,R} y_p = \boldsymbol{\delta}_{X,R}(y_1,\dots,y_p)$.

Для доказательства леммы осталось проверить, что при любом описанном способе группировки операндов справедливо $\boldsymbol{\delta}_{X,R}(y_1,\dots,y_p) = \boldsymbol{\delta}_{X,R}(x_1,\dots,x_{m+1})$. Если при этом все состояния $x_1,\dots,x_{m+1}$ пусты, то очевидно, что $\boldsymbol{\delta}_{X,R}(y_1,\dots,y_p) = \boldsymbol{\delta}_{X,R}(x_1,\dots,x_{m+1}) = O$. Далее рассмотрим нетривиальный случай. По определению имеем $\boldsymbol{\delta}_{X,R}(x_1,\dots,x_{m+1}) = x_0$, где x_0 – первый справа непустой элемент набора $x_1,\dots,x_{m+1}$. Рассмотрим группу G_0, в которую вошел элемент x_0. Очевидно, что все группы G_j, расположенные правее G_0, состоят лишь из пустых элементов. Как следствие – соответствующие им значения y_j также пусты. Кроме того, и в самой группе G_0 элемент x_0 – *первый* справа непустой. Применяя для вычислений в G_0 предположение индукции, получим, что результат в этой группе равен x_0. Таким образом, первый справа непустой элемент в наборе $y_1,\dots,y_p$ равен x_0. Этот факт означает, что $\boldsymbol{\delta}_{X,R}(y_1,\dots,y_p) = x_0$. □

Аналогично доказывается, что при любом порядке выполнения операций $\bigcup_{X,L}$ справедливо $x_1 \bigcup_{X,L} x_2 \bigcup_{X,L} \dots \bigcup_{X,L} x_m = \boldsymbol{\delta}_{X,L}(x_1,\dots,x_m)$.

Лемма 2. Пусть $A_1,\dots,A_m \in \mathbb{F}_X$, $m \geq 0$. Рассмотрим выражение $C = A_1 \bigcup_{X,R} A_2 \bigcup_{X,R} \dots \bigcup_{X,R} A_m$. Введем обозначения $a_i = [A_i]$ и $a_0 = \boldsymbol{\delta}_{X,R}(a_1,\dots,a_m)$. Тогда при любом ассоциативном порядке выполнения операций $\bigcup_{X,R}$ справедливо: если существует такое j, что $A'_j \neq A_j$, то $C = A'_1 \cup A'_2 \cup \dots \cup A'_m \cup a_0$, иначе $C = A'_1 \cup A'_2 \cup \dots \cup A'_m$. Кроме того, в любом случае $[C] = a_0$.

Доказательство. Вновь воспользуется индукцией по $m \geq 0$. При $m \leq 2$ утверждение леммы также непосредственно вытекает из определений $\bigcup_{X,R}$ и $\boldsymbol{\delta}_{X,R}$. Предположим, что оно справедливо при любом количестве операндов, не превосходящем некоторого m. Рассмотрим случай $m+1$.

Заметим вначале, что если все $A'_j = A_j$, $j = 1,\dots,m+1$, то по определению операции $\bigcup_{X,R}$ при любом порядке вычислений будем иметь $C = A'_1 \cup \dots \cup A'_{m+1}$. Осталось рассмотреть альтернативный вариант. В этом варианте хотя бы один элемент набора $A_1,\dots,A_{m+1}$ имеет непустое состояние. Обозначим его A_0. Очевидно, что $[A_0] = a_0$, где a_0 определен в условии леммы.

Объединим в группы наборы смежных операндов соответствующего выражения C таким образом, чтобы каждый операнд находился ровно в одной группе. Для исключения тривиальных случаев потребуем, чтобы групп было больше одной, и хотя бы одна из них содержала более чем один операнд. Предположим, что количество групп оказалось равным p. По их построению $p \leq m$ и количество операндов в группе не превосходит m. Следовательно, для каждой группы выполнено утверждение леммы, и в группах могут быть получены однозначные результаты вычислений. Обозначим их $Y_1,\dots,Y_p$. Тогда, по предположению индукции, если некоторая группа G_k состоит из смежных элементов $A_{k1},\dots,A_{kl}$, то $Y'_k = A'_{k1} \cup \dots \cup A'_{kl}$ и $[Y'_k] = \boldsymbol{\delta}_{X,R}([A'_{k1}],\dots,[A'_{kl}])$.

Напомним, что рассматривается такой вариант, когда для некоторого j справедливо $A_j \neq A'_j$. Как следствие, для некоторой группы G_{k0} (содержащей элемент A_j) по предположению индукции выполнено $Y_{k0} \neq Y'_{k0}$. Этим фактором ниже будет определяться вид окончательного результата вычислений над $Y_1,\dots,Y_p$.

Итак, наряду с выражением C рассмотрим выражение $Y_1 \bigcup_{X,R} \dots \bigcup_{X,R} Y_p$, где $p \leq m$. Для него также выполнено предположение индукции, то есть $Y_1 \bigcup_{X,R} \dots \bigcup_{X,R} Y_p = Y'_1 \bigcup \dots \bigcup Y'_p \bigcup y_0$, где $y_0 = \delta_{X,R}([Y_1],\dots,[Y_p])$. Подставляя вместо Y_k их выражения через исходные операнды, получим

$$Y_1 \bigcup_{X,R} \dots \bigcup_{X,R} Y_p = A'_1 \bigcup \dots \bigcup A'_{m+1} \bigcup y_0 .$$

Чтобы доказать требуемое представление для C, осталось установить равенство $y_0 = a_0$. Для этого рассмотрим группу G_0, в которую вошел элемент A_0 (см. начало доказательства). Результат вычислений в группе G_0 обозначим Y_0. Очевидно, что все группы G_j, расположенные правее G_0, состоят лишь из элементов с пустыми состояниями. По этой причине соответствующие им значения $[Y_j]$ также пусты. Кроме того, и в самой группе G_0 элемент A_0 – это *первый* справа элемент с непустым состоянием. Применяя для вычислений в G_0 предположение индукции, получим, что $[Y_0] = a_0$. Таким образом, первый справа непустой элемент в наборе $[Y_1],\dots,[Y_p]$ равен a_0. Этот факт означает, что $y_0 = a_0$.

Рассмотрим, наконец, вопрос о состоянии элемента C, который вычисляется в соответствии с принятым ранее разбиением операндов на группы. Состояние C также может быть получено с помощью предположения индукции, которое сначала применяется к каждой группе, а затем к результатам объединения групп, то есть $[C] = y_0$. Если состояния не всех элементов $A_1,\dots,A_{m+1}$ пусты, то $a_0 \neq O$ и, как показано выше, $y_0 = a_0$. В противном случае для каждого Y_k справедливо $[Y_k] = O$. Соответственно $y_0 = \delta_{X,R}([Y_1],\dots,[Y_p]) = O = a_0$. □

Аналогичная лемма справедлива и для операции $\bigcup_{X,L}$.

Из леммы 2 непосредственно вытекает следующая теорема.

Теорема 1. Операции $\bigcup_{X,R}$ и $\bigcup_{X,L}$ ассоциативны. □

Операцию $\bigcup_{X,R}$ будем называть некоммутативным объединением с правым приоритетом, операцию $\bigcup_{X,L}$ – некоммутативным объединением с левым приоритетом.

Обозначим $\supseteq_X$ отношение, состоящее из всех таких упорядоченных пар (A,B) элементов $A,B \in \mathbb{F}_X$, что $A \supseteq B$ и $[A] = [B]$.

Определение 1. Множество $\mathbb{F}_X$ с введенными на нем некоммутативными операциями $\bigcup_{X,R}$, $\bigcup_{X,L}$, а также с исходной операцией пересечения $\bigcap$, назовем верхне-некоммутативной (в контексте настоящего раздела – просто некоммутативной) решеткой, порожденной данным элементом X решетки $\mathbb{F}$.

В дальнейшем будем рассматривать в основном одну из двух новых операций – $\bigcup_{X,R}$, обозначая ее для краткости $\bigcup_X$ и используя полные обозначения лишь при необходимости. Аналогично вместо $\delta_{X,R}$ будем использовать обозначение δ_X.

Нетрудно заметить, что при $X = O$ решетка $\mathbb{F}_X$ совпадает с $\mathbb{F}$. Таким образом, понятие некоммутативной решетки шире понятия обычной решетки. Рассмотрим некоторые детали связи некоммутативного объединения $\bigcup_X$ с обычной операцией $\bigcup$.

Выражение (некоммутативное объединение) $A \bigcup_X B$ будем называть нормализованным, если из двух элементов A, B точку элемента X содержит не более чем один элемент.

Замечание 1. Очевидно, что для нормализованного объединения справедливо $A \bigcup_X B = A \bigcup B = B \bigcup A$.

Из определения операции $\bigcup_X$ следует, что если для $A, B \in \mathbb{F}_X$ выполнено $A' \neq A$ и $B' \neq B$, то имеет место равенство $A \bigcup_X B = A' \bigcup_X B$. Если же $A' = A$ либо $B' = B$, то объединение $A \bigcup_X B$ уже нормализовано. Этот факт позволяет сделать вывод о том, что выражение вида $A \bigcup_X B$ всегда может быть заменено равнозначным ему нормализованным.

Возможно обобщение понятия нормализации на случай m операндов. Объединение $A_1 \bigcup_X \ldots \bigcup_X A_m$, в котором не более чем один элемент содержит точку элемента X, называется нормализованным. Справедлива следующая лемма.

Лемма 3. Любое выражение вида $A_1 \bigcup_X \ldots \bigcup_X A_m$ может быть заменено равнозначным ему нормализованным. Новое выражение получается из исходного таким образом, что каждый операнд A_j со свойством $A_j \neq A_j'$ заменяется на A_j', за исключением крайнего правого из таковых, а остальные операнды сохраняются.

Доказательство. Если операндов с указанным свойством в выражении нет или он только один, то оно само является нормализованным. В противном случае в силу леммы 2 имеем $A_1 \bigcup_X \ldots \bigcup_X A_m = A_1' \bigcup A_2' \bigcup \ldots \bigcup A_m' \bigcup a_0$. Заметим, что по определению элемент a_0 представляет собой состояние такого A_i, который в наборе $A_1, \ldots, A_m$ расположен не левее чем крайний справа элемент A_j со свойством $A_j \neq A_j'$. В предельном случае возможно лишь равенство $[A_j] = a_0$. Описанное же в лемме преобразование затрагивает лишь те операнды исходного выражения, которые расположены левее чем A_j. Следовательно, оно сохраняет как способ вычисления выражения, так и саму величину $A_1' \bigcup A_2' \bigcup \ldots \bigcup A_m' \bigcup a_0$. □

В следующем подразделе определяются некоммутативные решетки, порожденные не единственным элементом X, а их конечным либо счетным множеством $\{X^p\}$. В этом случае состояния элементов определяются векторами, компоненты которых составлены из точек множеств $\{X^p\}$. Именно такие некоммутативные решетки адекватно моделируют работу императивных алгоритмов.

5.2.2. Некоммутативные решетки с расширенным множеством X

Введем вместо одного элемента несовместимых точек их конечное либо счетное множество $\bar{X} = \{X^p\}$, где $X^p \in \mathbb{F}$, $p \in P$. Рассмотрим совокупность всех элементов полной решетки $\mathbb{F}$, любой из которых содержит не более одной точки каждого элемента X^p. По-прежнему обозначим такое множество $\mathbb{F}_X$.

Будем рассматривать *векторы состояний* вида $\bar{x} = \{x^p\}$, $p \in P$, где каждая компонента x^p является точкой элемента X^p либо равна O. На парах векторов состояний определим покомпонентную операцию $\bigcap$, которая выполняется над соответствующими составляющими векторов как элементами решетки $\mathbb{F}$. Введем также операцию $\bigcup_{X,R} = \bigcup_X$ такую, что $\bar{x} \bigcup_{X,R} \bar{y} = \bar{z} = \{z^p\}$, $p \in P$, где

$$z^p = \begin{cases} y^p, y^p \neq 0 \\ x^p, y^p = 0 \end{cases}.$$

Соответственно определяется операция $\bigcup_{X,L}$: $\bar{x} \bigcup_{X,L} \bar{y} = \bar{y} \bigcup_{X,R} \bar{x}$.

Лемма 1. Операции $\bigcup_{X,R}$ и $\bigcup_{X,L}$ ассоциативны.

Доказательство. Поскольку указанные в условии леммы операции выполняются над векторами покомпонентно, то их исследование сводится к рассмотрению каждой компоненты в отдельности, то есть – случая $p = 1$, результаты для которого были приведены в п. 5.2.1. □

Для состояния $\bar{x} = \{x^p\}$ обозначим $x = \bigcup_p x^p$, где операция объединения берется в смысле решетки $\mathbb{F}$. В силу предполагаемой полноты $\mathbb{F}$ имеем $x \in \mathbb{F}_X$. Аналогично п. 5.2.1, для любого $A \in \mathbb{F}_X$ справедливо представление

$$A = A' \bigcup x_A, \tag{1}$$

где x_A – элемент указанного выше вида (объединение компонент вектора состояния $\bar{x}_A$), а A' не содержит ни одной точки элементов X^p. Всюду для данного $A \in \mathbb{F}_X$ символом A' будем обозначать соответствующий элемент в представлении (1).

Элементам множества $\mathbb{F}_X$ будем приписывать состояния вида $\overline{x}_A = \{x_A^p\}$, соответствующие представлению (1). Если для $A \in \mathbb{F}_X$ выполнено $A \neq A'$, то A всегда имеет состояние $\overline{x}_A$. Если же $A' = A$, то такой элемент A в зависимости от ситуации способен находиться в различных состояниях, однако в каждый фиксированный момент – не более чем в одном. Факт нахождения элемента $A \in \mathbb{F}_X$ в состоянии $\overline{x}_A$ будем обозначать $[A] = \overline{x}_A$. Состояние нижней грани O решетки $\mathbb{F}$ всегда считается «пустым», то есть все его компоненты равны O. Если все компоненты вектора состояния произвольного элемента A равны O, то будем обозначать этот факт как $[A] = [O]$. Для $C = A \bigcap B$, $A, B \in \mathbb{F}_X$ положим $[C] = [A] \bigcap [B]$, где операция $\bigcap$ выполняется над $[A]$ и $[B]$ покомпонентно (см. выше).

На множестве $\mathbb{F}_X$ введем операцию $\bigcup_{X,R}$ следующим образом. Пусть $A, B \in \mathbb{F}_X$. Тогда $A = A' \bigcup x_A$, $B = B' \bigcup x_B$. Положим $C = A \bigcup_{X,R} B = A' \bigcup B' \bigcup x_C$, где $\overline{x}_C = \overline{x}_A \bigcup_{X,R} \overline{x}_B$, а операция $\bigcup_{X,R}$ над состояниями $\overline{x}_A, \overline{x}_B$ определена выше. При этом $[C] = [A] \bigcup_{X,R} [B]$. Последнее равенство существенно, если $A = A'$ и $B = B'$.

Соответственно введем операцию $\bigcup_{X,L}$: $A \bigcup_{X,L} B = B \bigcup_{X,R} A$.

Теорема 1. Каждая из операций $\bigcup_{X,R}$ и $\bigcup_{X,L}$ является ассоциативной.

Доказательство. Пусть $A_1, \ldots, A_m \in \mathbb{F}_X$, $m \geq 0$. Рассмотрим выражение $C = A_1 \bigcup_{X,R} A_2 \bigcup_{X,R} \ldots \bigcup_{X,R} A_m$. Вычисляя его, например, последовательно слева направо, по определению операции $\bigcup_{X,R}$ получим $C = (A_1' \bigcup A_2' \bigcup \ldots \bigcup A_m') \bigcup x_C$, где $\overline{x}_C = \overline{x}_{A_1} \bigcup_{X,R} \overline{x}_{A_2} \bigcup_{X,R} \ldots \bigcup_{X,R} \overline{x}_{A_p}$, причем $[C] = [A_1] \bigcup_{X,R} [A_2] \bigcup_{X,R} \ldots \bigcup_{X,R} [A_m]$. Отсюда нетрудно заметить, что ассоциативность $\bigcup_{X,R}$ следует из ассоциативности обычной операции $\bigcup$ на решетке $\mathbb{F}$, а также доказанной в лемме 1 ассоциативности операции $\bigcup_{X,R}$ на векторах состояний. Операция $\bigcup_{X,L}$ рассматривается аналогично. □

Операцию $\bigcup_{X,R}$ будем называть некоммутативным объединением с правым приоритетом, операцию $\bigcup_{X,L}$ – некоммутативным объединением с левым приоритетом.

Замечание 1. Пусть $A, B \in \mathbb{F}_X$. Очевидно, что $A \bigcup_{X,R} B \subseteq A \bigcup B$, имея в виду рассматриваемую решетку $\mathbb{F}$. Если же $[A] = [B]$, то $A \bigcup_{X,R} B = A \bigcup B$.

Аналогичное замечание справедливо и для операции $\bigcup_{X,L}$.

Замечание 2. Из определения этих операций очевидным образом следуют также соотношения $B \subseteq A \bigcup_{X,R} B$ и $A \subseteq A \bigcup_{X,L} B$.

Обозначим $\supseteq_X$ отношение, состоящее из всех таких упорядоченных пар (A, B) элементов $A, B \in \mathbb{F}_X$, что $A \supseteq B$ и $[A] = [B]$.

Определение 1. Множество $\mathbb{F}_X$ с введенными на нем некоммутативными операциями $\bigcup_{X,R}$, $\bigcup_{X,L}$, а также с исходной операцией пересечения $\bigcap$, назовем верхне-некоммутативной (в контексте настоящего раздела – просто некоммутативной) решеткой, порожденной данным множеством $X = \{X^p\}$ элементов решетки $\mathbb{F}$.

Аналогично п. 5.2.1, при $X = O$ решетка $\mathbb{F}_X$ совпадает с $\mathbb{F}$, то есть и в данном случае понятие некоммутативной решетки шире понятия обычной решетки.

Из двух введенных ранее операций некоммутативного объединения будем в основном рассматривать одну – $\bigcup_{X,R}$, обозначая ее для краткости $\bigcup_X$ и используя полные обозначения лишь при необходимости.

Замечание 3. Пусть $A, B \in \mathbb{F}_X$. Нетрудно заметить, что $A \bigcup_X B \subseteq A \bigcup B$, имея в виду решетку $\mathbb{F}$. Если же $[A] = [B]$, то $A \bigcup_X B = A \bigcup B$.

Замечание 4. Из определения операции U_X следует, если $A \supseteq_X B$, то справедливо $A \bigcup_X B = A$.

Замечание 5. Из определения этой же операции очевидным образом следует также соотношения $B \subseteq A \bigcup_X B$.

Рассмотрим связь операции некоммутативного объединения $\bigcup_X$ с обычной операцией $\bigcup$.

Выражение $A \bigcup_X B$ называется нормализованным, если несовместимые точки содержатся не более чем в одном из двух элементов A, B.

Важным свойством нормализованного объединения $A \bigcup_X B$ является очевидное соотношение $A \bigcup_X B = A \bigcup B = B \bigcup A$. Из равенств $A \bigcup_X B = A' \bigcup B' \bigcup x_C = A' \bigcup (B' \bigcup x_C)$ следует, что выражение вида $A \bigcup_X B$ всегда может быть заменено нормализованным с сохранением результата.

Понятие нормализации некоммутативного объединения легко обобщается на случай нескольких операндов. Объединение $A_1 \bigcup_X \ldots \bigcup_X A_m$, в котором несовместимые точки содержит не более чем один элемент, называется нормализованным. Из доказательства теоремы 1 легко выводится представление $A_1 \bigcup_X \ldots \bigcup_X A_m = A_1' \bigcup \ldots A_{m-1}' \bigcup (A_m' \bigcup x_C)$. Отсюда следует, что любое выражение вида $A_1 \bigcup_X \ldots \bigcup_X A_m$ может быть заменено равнозначным ему нормализованным объединением.

5.2.3. Немонотонные логические отношения

В разделе 2.3 было рассмотрено свойство монотонности отношений на решетках. Показано (лемма 2.3.1[1]), что для логического

[1] Лемма 1 из п. 2.3.

отношения условие дистрибутивности эквивалентно условию монотонности. В связи с этим обстоятельством, в данном разделе при определении логических отношений на некоммутативных решетках будем исходить из некоторого обобщения свойства монотонности.

Псевдомонотонным называется такое бинарное отношение R на решетке $\mathbb{F}_X$, для которого выполнено следующее условие *псевдомонотонности*:

$$\text{если } (A,B) \in R, \text{ то } (A, A \bigcup_X B) \in R \quad (\forall A,B \in \mathbb{F}_X). \tag{1}$$

Отношение R на $\mathbb{F}_X$ называется продукционно-логическим (в данном разделе – просто логическим), если оно содержит $\supseteq_X$, псевдомонотонно и транзитивно. Логическим замыканием отношения R, заданного на $\mathbb{F}_X$, называется наименьшее логическое отношение, содержащее R. В этом случае LP-структурой называется некоммутативная решетка с заданным на ней логическим отношением.

Нетрудно заметить, что отношение $\supseteq_X$ само является логическим.

Следующая лемма «локально» связывает свойство псевдомонотонности со свойством дистрибутивности логического отношения на некоммутативной решетке.

Лемма 1. Пусть R – логическое отношение на $\mathbb{F}_X$ и $A,B_1,B_2 \in \mathbb{F}_X$, причем $[A]=[B_1]=[B_2]$. Тогда из $(A,B_1),(A,B_2) \in R$ следует $(A, B_1 \bigcup_X B_2) \in R$.

Доказательство. Согласно замечанию 5.2.2.3[1], в условиях настоящей леммы имеем $B_1 \bigcup_X B_2 = B_1 \bigcup B_2$. Поскольку также $A \bigcup_X B_1 = A \bigcup B_1$ и $A \bigcup_X B_2 = A \bigcup B_2$, то для рассматриваемых элементов A,B_1,B_2 условие *псевдомонотонности* (1) превращается в условие *монотонности* вида (2.3.1). В этой связи, повторяя вторую часть доказательства леммы 5.2.1, получим $(A, B_1 \bigcup B_2) \in R$. Однако, поскольку в рассматриваемом случае $B_1 \bigcup_X B_2 = B_1 \bigcup B_2$, то лемма доказана. □

Для выяснения структуры немонотонного логического отношения введем понятие логической связи. Пусть R – отношение на некоммутативной решетке $\mathbb{F}_X$. Будем констатировать, что элементы $A,B \in \mathbb{F}_X$ дистрибутивно связаны отношением R, если существует упорядоченная пара $(A_1,B_1) \in R$ либо $A_1 = B_1$, такая, что $A \supseteq_X A_1$ и $B = A \bigcup_X B_1$.

Замечание 1. Очевидно, что при $A \supseteq_X B$ пара (A,B) дистрибутивно связана отношением R, в этом случае $(A_1,B_1)=(B,B)$.

---

[1] Замечание 3 из п. 5.2.2.

Замечание 2. Легко также видеть, если $(A,B)\in R$, то пара $(A, A\bigcup_X B)$ дистрибутивно связана отношением R. Для нее $(A_1,B_1)=(A,B)$.

Лемма 2. Если R – логическое отношение на $\mathbb{F}_X$ и пара (A,B) дистрибутивно связана отношением R, то $(A,B)\in R$.

Доказательство. Поскольку отношение R содержит $\supseteq_X$, то в условиях леммы $(A,A_1)\in R$. Далее, так как $(A_1,B_1)\in R$ и отношение R удовлетворяет (1), имеем $(A_1, A_1\bigcup_X B_1)\in R$. Отсюда, пользуясь транзитивностью R, получим $(A, A_1\bigcup_X B_1)\in R$. Применяя к последнему соотношению условие вида (1), находим, что $(A, A\bigcup_X A_1\bigcup_X B_1)\in R$. Наконец, поскольку $A\bigcup_X A_1 = A$ (см. замечание 5.2.2.4), то в силу ассоциативности операции $\bigcup_X$ (теорема 5.2.2.1), имеем $A\bigcup_X A_1\bigcup_X B_1 = A\bigcup_X B_1$. Таким образом, $(A, A\bigcup_X B_1)\in R$, что и требовалось доказать. □

Определение 1. Пусть R – произвольное отношение на $\mathbb{F}_X$ и даны два элемента $A,B\in\mathbb{F}_X$. Пусть также существует упорядоченный набор элементов $\vec{r}_{AB}=(B_1,\dots,B_m)$ ($B_1,\dots,B_m\in\mathbb{F}_X$, $0\le m<\infty$), такой, что в последовательности $(B_0,B_1),(B_1,B_2),\dots,(B_m,B_{m+1})$, где $B_0=A$, $B_{m+1}=B$, каждая пара дистрибутивно связана R (в случае $m=0$ это сама пара (A,B)). Тогда указанный набор $\vec{r}_{AB}$ называется логической цепочкой (длины m), соединяющей A и B посредством R. Пара (A,B) при этом называется логически связанной отношением R и этот факт обозначается $A\xrightarrow[X]{R}B$.

Замечание 3. Из леммы 2 следует: если R – логическое отношение на $\mathbb{F}_X$ и пара (A,B) логически связана отношением R, то $(A,B)\in R$.

Рассмотрим операции над логическими цепочками и их свойства.

Пусть существуют логические цепочки $\vec{r}_{AB}$ и $\vec{r}_{BC}$, соединяющие соответственно (A,B) и (B,C) отношением R. Цепочка $\vec{r}_{AC}=\vec{r}_{AB}\vec{r}_{BC}$, составленная последовательно из $\vec{r}_{AB},B,\vec{r}_{BC}$, называется транзитивным объединением цепочек $\vec{r}_{AB}$ и $\vec{r}_{BC}$.

Замечание 4. Очевидно, $\vec{r}_{AC}=\vec{r}_{AB}\vec{r}_{BC}$ является логической цепочкой, соединяющей (A,C) отношением R. Таким образом, если (A,B) и (B,C) логически связаны R, то и пара (A,C) логически связана R.

В разделе 2.3 при исследовании *монотонных* логических отношений наряду с операцией транзитивного объединения логических цепочек введено также и понятие суммы цепочек. Свойства суммы логических цепочек во многом обусловлены дистрибутивностью логических отношений. Поскольку рассматриваемые в данном разделе *немонотонные* отношения в общем случае не обладают свойством дистрибутивности, то и аналоги сумм цепочек «работают» лишь в локальном смысле (см. лемму 1), когда все элементы цепочек имеют одно и то же состояние.

Пусть задано произвольное отношение R на $\mathbb{F}_X$ и выбраны элементы $A, B, C, D \in \mathbb{F}_X$, такие, что существуют логические цепочки $\vec{r}_{AB} = (B_1, \dots, B_m)$ и $\vec{r}_{CD} = (D_1, \dots, D_p)$, соединяющие соответственно (A, B) и (C, D) отношением R, причем $[A] = [B_1] = \dots = [B_m] = [B] = [C] = [D_1] = \dots = [D_p] = [D]$. Пусть для определенности $m \ge p$. Суммой цепочек $\vec{r}_{AB}$ и $\vec{r}_{CD}$ называется цепочка $\vec{r}_{AB} + \vec{r}_{CD} = (B_1 \bigcup_X D_1, \dots, B_p \bigcup_X D_p, B_{p+1} \bigcup_X D_p, \dots, B_m \bigcup_X D_p)$.

Замечание 5. Аналогично разделу 2.3 можно показать, что $\vec{r}_{AB} + \vec{r}_{CD}$ является логической цепочкой, соединяющей пару $(A \bigcup_X B, C \bigcup_X D)$ отношением R.

Пусть даны логические цепочки $\vec{r}_{AB} = (B_1, \dots, B_m)$ и $\vec{r}_{AC} = (C_1, \dots, C_p)$, соединяющие соответственно (A, B) и (A, C), причем $[A] = [B_1] = \dots = [B_m] = [B] = [C_1] = \dots = [C_p] = [C]$. Дистрибутивным объединением цепочек $\vec{r}_{AB}$ и $\vec{r}_{AC}$ называется цепочка $\vec{r}_{A(B+C)} = \vec{r}_{AB} + \vec{r}_{AC}$.

Замечание 6. Поскольку дистрибутивное объединение – это частный случай суммы, то $\vec{r}_{A(B+C)}$ является логической цепочкой, соединяющей $(A, B \bigcup_X C)$ отношением R.

Введенные над логическими цепочками операции легко могут быть распространены на конечное число операндов с сохранением указанных свойств.

Перейдем к доказательству существования логического замыкания отношения на некоммутативной решетке.

Теорема 1. Для произвольного отношения R на некоммутативной решетке $\mathbb{F}_X$ логическое замыкание существует и представляет собой множество $\xrightarrow[X]{R}$ всех упорядоченных пар $A, B \in \mathbb{F}_X$, логически связанных отношением R.

Доказательство. В силу замечания 1 определенное в условии теоремы отношение $\xrightarrow[X]{R}$ содержит $\supseteq_X$. Докажем, что отношение $\xrightarrow[X]{R}$ транзитивно. Пусть $A \xrightarrow[X]{R} B, B \xrightarrow[X]{R} C$. Тогда существуют логические цепочки $\vec{r}_{AB}$ и $\vec{r}_{BC}$, соединяющие соответственно (A, B) и (B, C) отношением R. Рассмотрим цепочку $\vec{r}_{AC} = \vec{r}_{AB}\vec{r}_{BC}$ – транзитивное объединение исходных цепочек. Согласно замечанию 4, $\vec{r}_{AC}$ является логической цепочкой, соединяющей A и C в R. Следовательно, $A \xrightarrow[X]{R} C$, то есть определенное в теореме отношение транзитивно. Наконец, из замечания 2 следует, что $\xrightarrow[X]{R}$ удовлетворяет условию (1).

Таким образом, отношение $\xrightarrow[X]{R}$ – логическое. Покажем теперь, что $\xrightarrow[X]{R} \supseteq R$. Пусть $(A, B) \in R$. Тогда в силу замечания

2 пара $(A, A \bigcup_X B)$ дистрибутивно связана R и соответственно $A \xrightarrow[X]{R} A \bigcup_X B$. Поскольку $A \bigcup_X B \supseteq B$ (см. замечание 5.2.2.5) и $\xrightarrow[X]{R}$ содержит $\supseteq_X$, то $A \bigcup_X B \xrightarrow[X]{R} B$. Наконец, в силу доказанной выше транзитивности $\xrightarrow[X]{R}$, получим $A \xrightarrow[X]{R} B$.

Итак, логическое отношение $\xrightarrow[X]{R}$ содержит R. Осталось показать, что оно наименьшее из таковых. Пусть R_1 – любое другое логическое отношение, содержащее R. Пусть также $A \xrightarrow[X]{R} B$. Тогда, поскольку существует соединяющая (A, B) посредством $R \subseteq R_1$ логическая цепочка $\vec{r}_{AB}$, то в силу замечания 3 справедливо $(A, B) \in R_1$. Таким образом, $\xrightarrow[X]{R} \subseteq R_1$. □

В дальнейшем понадобятся еще несколько новых понятий.

Логическая цепочка $\vec{r}_{AB} = (B_1, \ldots, B_m)$ называется *полной*, если при добавлении к ней в любую позицию любого элемента $B_{m+1} \in \mathbb{F}_X$, не совпадающего с соседними элементами, она перестает быть цепочкой, логически связывающей пару (A, B).

Логическая цепочка $\vec{r}_{AB}$ называется *терминальной*, если она не может быть продолжена. Этот факт означает, что не существует такого элемента $C \in \mathbb{F}_X$, что пара (A, C) логически связана в R, $C \neq B$ и цепочка $\vec{r}_{AB}$ содержится в начале цепочки $\vec{r}_{AC}$.

Логическая цепочка $\vec{r}_{AB} = (B_1, \ldots, B_m)$ называется монотонной, если из каждого элемента несовместимых точек X^p ($p \in P$) в элементах $(A, B_1, \ldots, B_m, B)$ в совокупности содержится не более одной точки. Это означает, что *монотонная* цепочка осуществляет *монотонный* логический вывод.

5.2.4. О продукционной модели императивных алгоритмов

Как известно, во время работы императивной программы информация (подобно немонотонной логике) не только накапливается, но и замещается. В разделе 1.8 представлен фрагмент программы на языке Паскаль, вычисляющий максимальный элемент целочисленного массива. Затем для этой программы была составлена эквивалентная логическая программа – исходный набор фактов и совокупность правил продукционного вывода. Возникает вопрос: любую ли императивную программу можно заменить эквивалентной ей логической? В настоящем подразделе в качестве приложения немонотонных LP-структур приводится обоснование положительного ответа на этот вопрос. Данное исследование можно считать частным примером, который показывает применимость немонотонных LP-структур для исследования свойств и оптимизации императивных программ.

Введем необходимые для дальнейшего изложения базовые определения.

Определение 1. Логической программой на некоммутативной решетке $\mathbb{F}_X$ называется тройка $P_R = (F_0, F_{\mathrm{Res}}, R)$, где F_0, F_{Res} – два фиксированных подмножества решетки $\mathbb{F}_X$, которые называются соответственно множеством *начальных элементов* и множеством *результатов*, R – некоторое бинарное отношение на $\mathbb{F}_X$. *Выполнением* логической программы P_R с данным *начальным элементом* $A_0 \in F_0$ назовем нахождение совокупности всех таких максимальных элементов $A_t \in F_{\mathrm{Res}}$, что пара (A_0, A_t) логически связана полной терминальной цепочкой в R. Полученную совокупность элементов $\{A_t\}$ будем называть *результатами* программы P_R, соответствующими данному начальному элементу A_0 $(t \in T)$.

Здесь в качестве T фигурирует некоторый произвольный контейнер индексов, как это принято в общей теории множеств [130]. Если T состоит из единственного элемента, то результат логической программы называется *однозначным*.

В настоящем разделе не ставится задача общего исследования свойств немонотонных логических программ. Основная цель – выявление их связи с императивными алгоритмами. Для этого рассмотрим важный подкласс таких программ – детерминированные логические программы на решетках. Особенность детерминированной программы состоит в том, что при некотором фиксированном $A_M \in \mathbb{F}_X$, который перед выполнением добавляется к каждому начальному элементу и называется *начальным управляющим элементом*, ее результат однозначен, а процесс выполнения полностью предсказуем.

Логическая программа $P_R = (F_0, F_{\mathrm{Res}}, R)$ на $\mathbb{F}_X$ называется *детерминированной* при данном $A_M \in \mathbb{F}_X$, если для любого $A_0 \in F_0$ при начальном элементе $A_0 \bigcup_X A_M$ ее соответствующий результат $A_{\mathrm{Res}} \in F_{\mathrm{Res}}$ однозначен и достигается единственной (с точностью до порядка элементов в монотонных подцепочках) логической цепочкой в R.

Для таких программ можно выделить еще одно подмножество решетки $\mathbb{F}_X$ – множество *управляющих элементов* F_M, которое не пересекается с F_0 и F_{Res}. Роль управляющих элементов состоит исключительно в обеспечении детерминированного характера действий. Таковым является указанный в определении 2 управляющий начальный элемент $A_M \in F_M$. В связи с этим обстоятельством для детерминированных программ будем также использовать обозначение $P_{R,M} = (F_M, F_0, F_{\mathrm{Res}}, R)$.

Проиллюстрируем эти понятия на упомянутом ранее примере из раздела 1.8. Итак, в качестве некоммутативной решетки $\mathbb{F}_X$ в данном случае можно рассматривать булеан (множество подмножеств) всевозможных пар вида *переменная = значение* для всех переменных, используемых в логической программе. Множество значений каждой отдельной переменной x_i образует совокупность несовместимых точек X_i решетки $\mathbb{F}_X$ (переменная не может иметь одновременно несколько значений своего типа). Отношение R образовано такими упорядоченными парами, что левая часть пары (как элемент решетки) порождает предпосылку некоторого правила логической программы, а правая часть пары получается в результате применения того же правила.

Управляющее множество F_M в данном примере состоит из следующих пар несовместимых точек: {*начало=да*, *нет*}, {*цикл=да*, *нет*}. Множество F_0 составляют всевозможные наборы значений исходного целочисленного массива A, множество $F_{\mathrm{Re}\,s}$ – целочисленные значения переменной *Max*. Очевидно, данная логическая программа является детерминированной при управляющем начальном элементе A_M: *начало=да*. Результат выполнения программы – это элемент решетки $\mathbb{F}_X$, образованный ее точкой $Max = A_{\max}$, где $A_{\max}$ – фактический максимальный элемент массива.

Для дальнейшего изложения потребуется некоторая формализация понятия императивного алгоритма в терминах некоммутативных решеток. Будем здесь опираться на известную структурную теорему Дейкстры, согласно которой любой императивный алгоритм может быть преобразован в эквивалентный, содержащий лишь действия, управляемые структурами трех видов: *цепочкой*, *альтернативой* и *циклом while* (с предусловием). К сожалению, литературные источники с доказательством этой теоремы труднодоступны. Однако подобная теорема (правда, без ссылки на Э. Дейкстру), сформулирована и доказана в [134].

В качестве (императивных) *элементарных действий* на некоммутативной решетке $\mathbb{F}_X$ будем рассматривать применения операции некоммутативного объединения. В частности, результатом действия над некоторым элементом $A \in \mathbb{F}_X$ можно считать $C = A \bigcup_X B$, где $B \in \mathbb{F}_X$.

Для выполнения императивной программы (как и логической) необходимо наличие исходных данных – некоторой совокупности возможных начальных элементов F_0 решетки $\mathbb{F}_X$. При выполнении действий (операций $\bigcup_X$) начальный элемент $A_0 \in F_0$ последовательно трансформируется в некоторый новый. На заключительном этапе получается результат программы – результирующий элемент $A_{\mathrm{Res}} \in F_{\mathrm{Res}} \subset \mathbb{F}_X$. Обозначим A_{Curr} переменную со значениями текущего элемента, который имеется на каждом шаге после выполнения всех предыдущих операций программы. Таким образом, до выполнения императивной программы $A_{Curr} = A_0$, после него – $A_{Curr} = A_{\mathrm{Res}}$.

Далее необходимо формализовать понятия управляющих структур.

Императивной цепочкой длины m будем называть произвольный упорядоченный набор $I = (B_C^1, \dots, B_C^m)$ элементов решетки $\mathbb{F}_X$. Индекс «C» означает возможную зависимость каждого элемента цепочки от текущего значения A_{Curr}. Результатом последовательного применения императивной цепочки $I = (B_C^1, \dots, B_C^m)$ к исходному значению A_{Curr} (результатом ее выполнения) назовем элемент $A_{Curr} \bigcup_X B_C^1 \bigcup_X \dots \bigcup_X B_C^m$, который становится новым значением переменной A_{Curr}.

Для определения альтернативы и цикла необходимо понятие *условия*. Будем предполагать, что вычисление условий не вызывает побочных эффектов. Любое логическое выражение в программе имеет множество истинности, которое может быть представлено в виде наборов значений переменных. Например, неравенство для вещественных переменных $x < y$ может быть реализовано множеством взаимоисключающих пар вида $\{x = a_1, y = a_2 \mid a_1, a_2 \in R_1,\ a_1 < a_2\}$, конечным в практических реализациях. Таким образом, в рассматриваемой модели каждое логическое условие эквивалентно принадлежности одного из заданных элементов решетки $\{A_{Cond}^t \mid t \in T\}$ текущему элементу A_{Curr}. Будем также обозначать $\{\overline{A}_{Cond}^t \mid t \in \overline{T}\}$ совокупность элементов, соответствующую отрицанию условия с элементами $\{A_{Cond}^t \mid t \in T\}$.

Альтернативой называется выражение вида

$$\text{if } \{A_{Cond}^t\} \text{ then } B_C^1 \text{ else } B_C^2. \tag{1}$$

Результат выполнения альтернативы – новое значение A_{Curr}, вычисляемое по формуле $A_{Curr} \bigcup_X B_C^1$, если при некотором $t \in T$ справедливо $A_{Cond}^t \subseteq A_{Curr}$, и $A_{Curr} \bigcup_X B_C^2$ – в противном случае.

Наконец, *циклом* называется конструкция

$$\text{while } \{A_{Cond}^t\} \text{ do } B_C. \tag{2}$$

Ее выполнение заключается в рекуррентном вычислении нового значения $A_{Curr} := A_{Curr} \bigcup_X B_C$, пока существует такое $t \in T$, что $A_{Cond}^t \subseteq A_{Curr}$.

Наконец, расширим введенные определения, предположив, что описанные выше конструкции могут быть вложенными. Этот факт означает, что в качестве их содержимого ($B_C^1, B_C^2, \ldots, B_C^m, B_C$) могут фигурировать не только отдельные элементы решетки, но и, в свою очередь, другие цепочка, альтернатива или цикл. Определения структур нетрудно формализовать, воспользовавшись некоторым аналогом формы Бэкуса–Наура (например, [89]). Однако в целях экономии места такое определение здесь не приводится. Для каждой из перечисленных конструкций будем использовать общее название – *императивная структура*.

Отметим, что используемая здесь грамматика императивных программ носит схематично-обобщенный характер. Ее задача – проиллюстрировать не конкретные вычисления, а возможности управления последовательностью действий.

Определение 2. *Императивной программой* на некоммутативной решетке $\mathbb{F}_X$ называется тройка $P_I = (F_0, F_{\mathrm{Res}}, I)$, где F_0, F_{Res} – два фиксированных подмножества элементов решетки $\mathbb{F}_X$ (множества *начальных элементов* и *результатов*), а I – любая императивная структура на $\mathbb{F}_X$. *Выполнением* императивной программы P_I с данным *начальным* элементом $A_0 \in F_0$ назовем применение конструкции I к элементу $A_{Curr} = A_0$. Получаемый после этого элемент $A_{Curr} = A_{\mathrm{Res}} \in F_{\mathrm{Res}}$ будем называть *результатом* программы P_I, соответствующим данному начальному A_0.

Очевидно, любая императивная программа детерминирована. В связи с этим обстоятельством императивные (как и детерминированные логические) программы могут также называться алгоритмами.

Замечание 1. Сравнивая определения логических и императивных программ, можно отметить, что каждый из этих видов программ по исходному элементу решетки вычисляет некоторый результирующий. Таким образом, правомерна постановка вопроса о сравнении их выразительных возможностей. Кроме того, можно рассматривать некоторый их гибрид, например, когда некоторые содержащиеся в императивных структурах элементы представляют собой логические программы с текущими исходным и результирующим элементами решетки.

Очевидно, структуру императивной программы можно изобразить в виде дерева, отражающего вложенность ее управляющих конструкций. Корень этого дерева представляет собой конструкцию верхнего уровня, а листьями являются элементы решетки $\mathbb{F}_X$. *Глубиной вложенности* (или просто *вложенностью*) императивной программы будем называть высоту этого дерева. В частности, любая управляющая структура императивной программы, содержащая лишь элементы решетки $\mathbb{F}_X$, имеет вложенность $= 1$.

Лемма 1. Для любой императивной программы $P_I = (F_0, F_{\mathrm{Re}\,s}, I)$, имеющей единичную вложенность, существует детерминированная логическая программа $P_{R,M} = (F_M, F_0, F_{\mathrm{Re}\,s}, R)$ с теми же $F_0, F_{\mathrm{Re}\,s}$, которая при любом начальном элементе $A_0 \in F_0$ и результате $A_{\mathrm{Re}\,s} \in F_{\mathrm{Re}\,s}$ программы P_I выдает для своего начального элемента вида $A_0 \bigcup_X A_M$ результат, совпадающий с $A_{\mathrm{Re}\,s}$.

Доказательство. В силу того, что глубина вложенности программы равна 1, достаточно рассмотреть лишь управляющие структуры в отдельности, а именно – цепочку, альтернативу и цикл с элементами решетки в качестве содержимого. В каждом случае для императивной структуры построим соответствующее отношение R, реализующее эквивалентную (в смысле результата) логическую программу. При этом будем использовать дополнительные управляющие элементы решетки $a_j,\ j = 0,\dots,n$ (несовместимые точки), которые обеспечивают детерминированность действий. Элемент a_0 служит стартовой управляющей точкой структуры, элемент a_n – завершающей. Не ограничивая общности, в целях упрощения формулировок можно считать, что решетка $\mathbb{F}_X$ содержит необходимые управляющие элементы изначально.

В случае императивной цепочки $I = (B_C^1,\dots,B_C^m)$ соответствующее отношение R составим из всех пар вида $(A_C^j \bigcup_X a_{j-1}, B_C^j \bigcup_X a_j)$, $j = 1,\dots,m$, где A_C^j – объединение всех элементов решетки, от которых зависит B_C^j. Тогда для любого начального элемента A_0 результат обеих программ будет вычисляться по одной и той же формуле $A_{\mathrm{Re}\,s} = A_0 \bigcup_X B_C^1 \bigcup_X \dots \bigcup_X B_C^m$.

Пусть теперь I – альтернатива вида (1). В этом случае отношение R может быть образовано парами $(A_{Cond}^t \bigcup_X A_C^1 \bigcup_X a_0, B_C^1 \bigcup_X a_1)$, $t \in T$ и $(\bar{A}_{Cond}^t \bigcup_X A_C^2 \bigcup_X a_0, B_C^2 \bigcup_X a_1)$, $t \in \bar{T}$, где множество $\{\bar{A}_{Cond}^t \mid t \in \bar{T}\}$ соответствуют отрицанию условия альтернативы. Данное отношение R порождает детерминированную логическую программу, результат которой для любого $A_0 \in F_0$ совпадает с результатом альтернативы I: если для некоторого $t \in T$ $A_0 \supseteq A_{Cond}^t$, то $A_{\mathrm{Re}\,s} = A_0 \bigcup_X B_C^1$, в противном случае – $A_{\mathrm{Re}\,s} = A_0 \bigcup_X B_C^2$.

Рассмотрим, наконец, последнюю управляющую структуру – цикл (2). Логическую программу построим на основе отношения R, состоящего из всех пар $(A_{Cond}^t \bigcup_X A_C \bigcup_X a_0, B_C)$, $t \in T$ и $(\bar{A}_{Cond}^t \bigcup_X a_0, a_2)$, $t \in \bar{T}$. Пропущенный здесь элемент a_1 резервируется для использования в случаях, когда цикл будет содержать вложенную структуру. Легко заметить, что полученная логическая программа детерминирована, а результат ее при любом начальном $A_{Curr} \in F_0$ совпадает с результатом цикла, а именно – пока для некоторого $t \in T$ имеет место $A_{Curr} \supseteq A^t_{Cond}$, выполняется $A_{Curr} := A_{Curr} \bigcup_X B_C$. □

Замечание 2. Как следует из доказательства леммы, соответствующая логическая программа не только получает аналогичный результат, но также сохраняет порядок и содержание вычислений императивной программы. Дополнительные действия логической программы связаны лишь с управляющими элементами решетки, с которыми исходная императивная программа не работает.

Перейдем теперь к основному утверждению настоящего подраздела. Оно распространяет результат леммы 1 на императивные программы произвольной вложенности.

Теорема 1. Для любой императивной программы $P_I = (F_0, F_{\mathrm{Re}s}, I)$ на некоммутативной решетке $\mathbb{F}_X$ можно построить детерминированную логическую программу $P_{R,M} = (F_M, F_0, F_{\mathrm{Re}s}, R)$, которая при любом начальном элементе $A_0 \in F_0$ и результате $A_{\mathrm{Re}s} \in F_{\mathrm{Re}s}$ программы P_I получает для своего начального элемента вида $A_0 \bigcup_X A_M$ тот же результат $A_{\mathrm{Re}s}$, при этом сохраняя порядок и содержание действий P_I.

Доказательство. Проведем его с помощью индукции по N – глубине вложенности программы P_I. При $N = 1$ утверждение теоремы доказано в лемме 1 (см. также замечание 2). Предположим далее, что оно верно для любого показателя вложенности, не превосходящего некоторого N. Покажем его справедливость для программы с вложенностью $N+1$.

Рассмотрим управляющую структуру верхнего уровня I_0 программы P_I. Ее содержимым являются элементы решетки либо другие императивные структуры. Для любой из этих вложенных структур можно применить предположение индукции, так как ее показатель не превосходит N. Таким образом, каждой непосредственно вложенной в I_0 императивной структуре I_l в формируемом отношении R будет соответствовать множество пар R_l с собственным набором управляющих точек $\{a^l_j \mid j = 0, \ldots, n_l\}$ (см. доказательство леммы 1). Подчеркнем, что несовместимыми между собой являются все точки a^l_j с общим верхним индексом.

Остается дополнить отношение R множеством пар R_0, которое будет реализовывать саму управляющую структуру I_0. Пусть I_0 – цепочка либо альтернатива. Тогда заменим в I_0 каждую непосредственно вложенную структуру I_l ее стартовой управляющей точкой a_0^l. Затем по лемме 1 построим для структуры I_0 соответствующую логическую программу, полагая при этом элементы, от которых зависит a_0^l, равными O.

Особым является случай, когда I_0 представляет собой цикл. В данной ситуации вложенной в I_0 структуре I_l сопоставим множество пар вида $(A_{Cond}^t \bigcup_X a_0^0, a_0^l \bigcup_X a_1^0)$, $t \in T$ для исключения преждевременных инициализаций I_l, а также пару $(a_e^l \bigcup_X a_1^0, a_0^0)$ для повторяемости цикла. Здесь a_e^l – завершающая управляющая точка структуры I_l, a_1^0 – зарезервированная ранее управляющая точка цикла I_0 (см. доказательство леммы 1).

Совокупность пар отношения полученной для I_0 логической программы примем в качестве искомого множества R_0, добавляемого к R. Поскольку порядок действий I_0 при этом сохраняется, а наличие каждой точки a_0^l дает старт выполнению алгоритма R_l, адекватно реализующего I_l, то найденное отношение R удовлетворяет требованиям теоремы. При этом $F_M = \bigcup_l \{a_j^l\}, A_M = a_0^0$. □

Замечание 3. Рассматриваемые в теореме 1 императивная и логическая программы выполняют одни и те же действия над общим для них подмножеством решетки $\mathbb{F}_X$. Этот факт позволяет сделать вывод о том, что в определенном смысле класс императивных программ является подклассом логических программ.

В настоящей главе были рассмотрены два специальных класса LP-структур, содержащих семантику немонотонного вывода. Таким образом, открываются возможности для применения описываемой теории в таких областях, которые ранее не относились к области продукционных систем. Изложенные подходы могут быть распространены и на другие задачи информатики. Следующая, заключительная, глава настоящей работы посвящена описанию компьютерной реализации общей теории LP-структур, а также ее практическим применениям к исследованию и оптимизации баз знаний продукционного типа.

Глава 6
КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ И ПРИМЕНЕНИЕ LP-СТРУКТУР

При разработке математических объектов исследования иногда должного внимания не уделяется их прикладным аспектам. В результате увеличивается вероятность того, что для применения таких моделей, по крайней мере, на текущем этапе, окажется недостаточно интеллектуальных или технических ресурсов. Основная цель настоящей главы – продемонстрировать возможность применения теории LP-структур на практике. С этой целью описывается созданная автором интегрированная среда разработки продукционных экспертных систем, а также анализируются реализация и применение LP-структуры для верификации и оптимизации созданных с ее помощью баз знаний.

Предварительно приводятся некоторые общие принципы реализации положений п.п. 2.3–2.6, не столь значимые для теоретического исследования, однако существенные на практике. Далее рассматриваются вопросы кодирования LP-структур, опирающиеся на известные методы представления решеток битовыми векторами [36, 77]. Анализируются особенности и интерфейс объектно-ориентированного класса LPStructure, разработанного автором на языке C++ с использованием библиотеки STL. Класс LPStructure инкапсулирует наиболее важные свойства и методы описанных в главе 2 стандартных LP-структур, включая нахождение логической редукции и решение продукционно-логических уравнений. В последующем разделе представлены структура и основные возможности пакета программ LPExpert (реализованного на Delphi), предназначенного для разработки и эксплуатации продукционных экспертных систем. Этот пакет первоначально создавался с использованием идей, изложенных в книге [79], с существенными их усовершенствованиями. Впоследствии к нему была добавлена динамическая библиотека – интерфейс класса LPStructure, что значительно расширило возможности пакета в плане эффективности создания с его помощью баз знаний. В заключительных разделах главы на тестовых примерах демонстрируются возможности LPExpert по выявлению избыточных правил, а также анализируются преимущества логического вывода на основе решения продукционно-логических уравнений в сравнении со стандартным обратным выводом.

Заметим, что рассматриваемые в настоящей главе алгоритмы показывают практическую применимость теории, изложенной в предыдущих главах книги. В некоторых случаях они могут быть улучшены.

Основными результатами главы 6 можно считать впервые созданную программную реализацию LP-структуры, а также применение теории LP-структур для решения практически значимых задач. Разработанные автором пакеты программ демонстрируют достаточно высокое быстродействие и, соответственно, выполняют возложенные на них функции. Создание более эффективных алгоритмов и получение соответствующих оценок могут составить предмет отдельных исследований, которые должны лишь подтвердить богатые потенциальные возможности стандартной теории LP-структур. Такие же выводы могут относиться и к реализации других видов LP-структур.

6.1. Общие принципы реализации

Одним из важных вопросов, возникающих при реализации LP-структур, является обоснование формата представления бинарных отношений на решетках. Сразу заметим, что, например, для булеана при количестве его точек n общее число элементов составит 2^n. Этот факт означает, что в общем случае способ хранения отношения на решетке в виде статической матрицы (смежности или инциденций) не подходит. Даже небольшой учебный пример из [79] содержит около 80 единичных значений объектов, что при моделировании LP-структурой трансформируется в соответствующее количество точек решетки. В данном случае использовались бы огромные разреженные матрицы, стандартное хранение которых в памяти компьютера было бы не только неэффективным, но и вряд ли возможным на практике. Таким образом, бинарное отношение на решетке будем представлять в виде (динамического) множества пар ее элементов. Последующие замечания также подчеркивают основную особенность реализации LP-структур, а именно – потребность снижения расхода памяти, иногда в ущерб быстродействию.

Далее с точки зрения практической реализации рассмотрим задачу нахождения логической редукции стандартной LP-структуры. В приложении к экспертным продукционным системам ее решение заключается в получении минимальной базы знаний, эквивалентной исходной.

Как показано в п. 2.5, для бинарного отношения R на решетке $\mathbb{F}$ логическая редукция $\underset{\sim}{R}^0$ может быть получена последовательным выполнением следующих шагов:

1) добавить к R все пары вида (A, A), где $A \in \mathbb{F}$ (рефлексивные пары), и обозначить новое отношение R_1;

2) добавить к R_1 всевозможные пары (A, B) с элементами вида $A = \bigcup_i A_i$, $B = \bigcup_i B_i$, где все (A_i, B_i) ($i = 1, \ldots, n$) принадлежат R_1;

3) объединить полученное отношение с отношением включения $\supseteq$ и обозначить новое отношение $\tilde{R}$;

4) построить транзитивную редукцию R^0 отношения $\tilde{R}$;

5) исключить из $\tilde{R}$ содержащиеся в нем пары вида $A \supset B$ и обозначить новое отношение R_{-1};

6) исключить из R_{-1} всевозможные пары (A, B) с элементами вида $A = \bigcup_i A_i$, $B = \bigcup_i B_i$, где все (A_i, B_i) ($i = 1, \ldots, n$) принадлежат R_{-1} и не совпадают с (A, B);

7) исключить из полученного отношения все рефлексивные пары.

В силу тех же соображений экономии памяти условимся, что физическое добавление к множеству новых пар следует производить лишь в случае необходимости. В частности, нет смысла хранить рефлексивные (шаг 1) или подчиненные (шаг 3) пары, особенно, если способ кодирования решетки будет допускать эффективное вычисление частичного порядка.

Непосредственная реализация шага 2 также потребовала бы чрезмерного расхода памяти, соизмеримого с построением булеана на универсуме R. Кроме того, в дальнейших вычислениях из огромного количества добавленных таким образом пар практически использовалась бы лишь незначительная часть. Как следствие, приходим к тезису о целесообразности динамического построения необходимых в дальнейшем пар множества $\tilde{R}$.

Обсудим также вопрос построения транзитивной редукции (шаг 4) отношения $\tilde{R}$. Как показано в [2], эта задача вычислительно эквивалентна задаче нахождения транзитивного замыкания, что, например, при применении известного алгоритма Уоршолла [83] составило бы $O(N^3)$ операций. Однако в нашем случае величина N заменяется выражением 2^N и, таким образом, нахождение транзитивной редукции посредством замыкания не представляется эффективным. В качестве практически реализуемого варианта остается исследование множества пар как такового на транзитивную избыточность. Таким образом, в предлагаемой реализации будем строить транзитивную редукцию отношения $\tilde{R}$ путем исключения пар, связанных транзитивными цепочками.

Теперь с точки зрения практической реализации рассмотрим вопросы решения продукционно-логических уравнений для стандартных LP-структур. Как уже отмечалось в главе 2, применение данного аппарата теоретически позволяет уменьшить количество медленных запросов, направляемых базе данных или интерактивному пользователю при обратном выводе.

Как было изложено в п. 2.6, для решения уравнения исходное каноническое отношение R представляется в виде слоев, каждый из которых порождает не более одного решения. Слой содержит максимально возможный набор пар исходного отношения с уникальными правыми частями. Два слоя различаются хотя бы одной парой. Непосредственная реализация слоев привела бы к избыточному хранению пар, так как слои могут иметь большие пересечения. Для решения этого вопроса поступим следующим образом.

Как и в п.2.6, вначале разобьем R на непересекающиеся подмножества, каждое из которых образовано парами вида (A, x_p) с одним и тем же точечным элементом x_p в качестве правой части. Обозначим эти подмножества R^p соответственно их элементу $x_p, p \in P$. При реализации канонического отношения R для каждого x_p достаточно хранить лишь совокупность левых частей пар с правой частью x_p. Каждому из подмножеств R^p сопоставим итератор – индексную переменную j_p, способную перебирать собственное подмножество, останавливаясь на каждой паре ровно один раз. Таким образом, любой слой R_t в отношении R получается соответствующим набором значений итераторов $\{j_p\}$. Нахождение решения в отдельном слое сводится к получению списка начальных вершин графа, из которых достижима данная вершина (она соответствует точке правой части уравнения).

Работа с различными слоями может быть организована независимо и даже параллельно. Такой подход в определенном смысле максимально экономит память, однако может приводить к многократному повторению вычислений, ведь, как уже отмечалось ранее, два слоя могут иметь много общих пар. Другой вариант – запоминать результаты работы в слое, чтобы ими, при необходимости, можно было воспользоваться и в другом слое, который пересекается с данным. Этот способ несколько противоречит принятой ранее генеральной линии на экономию памяти, однако он не приводит к неустранимым деструктивным последствиям. В классе LPStructure реализуются оба указанных подхода. Выбор одного из них является опциональным и может задаваться пользователем.

6.2. Кодирование LP-структур

Очевидно, что способ представления в памяти LP-структур непосредственно связан с методами кодирования решеток. Последнему вопросу посвящены работы [36, 77, 126], а также специальный раздел монографии [120]. Из замечаний предыдущего пункта следует, что предлагаемая автором реализация LP-структур должна максимально экономить память и предоставлять быстрые «решеточные» операции (объединение, пересечение, частичный порядок). Этому требованию во многом удовлетворяет кодирование решеток битовыми векторами.

Как указано, например, в [77], конечная булева решетка может быть представлена множеством битовых векторов (векторов с двоичными значениями компонент) одинаковой размерности N, где N – число точек решетки. Каждому элементу a булевой решетки соответствует единственный битовый вектор $BitVect(a)$, при этом каждой точке – вектор с одной единицей в соответствующей позиции и нулями – в остальных. Вычисление операций на решетке сводится к вычислению побитовых логических операций сложения ($|$) и умножения ($\&$) над компонентами векторов, то есть

$$a \vee b = BitVect(a) \mid BitVect(b)\,;$$

$$a \wedge b = BitVect(a) \,\&\, BitVect(b)\,;$$

$$a \le b \Leftrightarrow BitVect(a) \,\&\, BitVect(b) = BitVect(a)\,;$$

$$a \le b \Leftrightarrow BitVect(a) \mid BitVect(b) = BitVect(b)\,.$$

Как известно (например, [193]), побитовые логические операции относятся к низкоуровневым и, соответственно, наиболее быстрым двуместным операциям в современных компьютерах. Конечно, теоретически с точки зрения анализа сложности алгоритмов нельзя утверждать, что операция над двумя векторами будет выполняться за константное время, поскольку разрядность компьютера (например, 32 или 64) вполне может оказаться недостаточной для представления битового вектора единственным словом памяти. Однако величина сложности операции $N / 32$ или $N / 64$, очевидно, вполне приемлема при общем количестве элементов решетки 2^N.

Поскольку представленная в данной главе реализация стандартных LP-структур использует конечные булевы решетки, то заявленные ранее потребности исчерпываются указанным выше способом кодирования решеток. Однако, если необходимо использование более общих видов решеток, можно обратиться, например, к исследованию [36], где описываются возможности кодирования битовыми векторами для произвольных решеток.

Еще один вопрос реализации LP-структур связан со способом хранения вторичных бинарных отношений на решетках. В соответствии с замечаниями п. 6.1 было принято представление бинарного отношения в виде множества пар. Следует уточнить, что эффективность операций доступа к множеству (поиск, включение и исключение элементов) на практике обеспечивается, например, возможностью его хранения в виде сбалансированного бинарного дерева (так реализуются множества в STL – стандартной библиотеке C++ [107]). Такой подход требует сопоставления элементам хранимого множества уникальных вполне упорядоченных ключей. Ключ по возможности должен помещаться в слове (или двойном слове) компьютера. Тогда при доступе к множеству операция сравнения ключей будет выполняться за константное время. Таким образом, приходим к идее хранения пары элементов решетки как пары ссылок на соответствующие им битовые векторы. Эти ссылки будут содержаться в смежных частях (старшей и младшей) двойного целого числа, и данное число может использоваться в качестве упомянутого выше ключа.

Для хранения канонических отношений на решетках при реализации ряда алгоритмов будем применять еще одно представление, а именно – в виде вектора множеств. Напомним, что каноническое отношение на точечной решетке (см. п. 2.4) состоит из пар «хорновского» типа. Правая часть такой пары представляет собой точку решетки, а левая – объединение точек. Для такого отношения можно построить массив, компонентами которого являются подмножества элементов решетки. Индексом этого массива служит номер точки решетки (например, номер единственной ненулевой компоненты в соответствующем точке битовом векторе), а содержимым компоненты массива – множество элементов решетки, составляющих с данной точкой пары в качестве их левых частей.

Указанное представление достаточно эффективно как с точки зрения использования памяти, так как хранятся лишь левые части пар, так и быстродействия, потому что доступ к ним осуществляется по индексу в массиве. Этот способ применяется для реализации в LP-структуре алгоритмов обратного вывода при неизменном множестве правил.

Представление отношения на решетке в виде указанного вектора множеств в определенной мере можно считать аналогом «константной» сети Rete [25]. Эта сеть используется для ускорения прямого (или смешанного) вывода в продукционных экспертных системах. При прямом выводе на основе содержимого рабочей памяти системы осуществляется «вход в продукции» со стороны их левых частей. По этой причине для элементов текущей рабочей памяти сеть Rete содержит ссылки на левые части продукций, удовлетворяемые этими элементами. Для каждой левой части продукции необходимо хранение и соответствующей ей правой части. В результате сеть Rete представляет собой довольно объемную надстройку над имеющимся множеством продукционных правил, хотя и эффективно организованную.

Предложенная автором конструкция (вектор множеств) предназначена для повышения эффективности исключительно обратного вывода. Она более экономична по памяти, поскольку не предусматривает дублирования информации. Если сеть Rete реализует дополнительную надстройку над множеством продукций, то вектор множеств *заменяет* множество продукций. Данный результат объясняется особенностью архитектуры стандартных продукционных систем, а именно – «неравноправием» левых и правых частей продукций. Согласно следствию 2.4.2[1], для отношения на точечной решетке всегда существует эквивалентное каноническое отношение (с элементарными правыми частями пар), в то время как левые части продукций в общем случае невозможно эквивалентным образом «урезать» до отдельных точек.

6.3. Класс LPStructure

Рассмотрим особенности построения и основные возможности класса LPStructure, реализующего стандартные LP-структуры, представленные в главе 2. Данная разработка выполнена на языке C++ с привлечением библиотеки STL – Standard Template Library в среде программирования MS Visual Studio.

Основной целью материала настоящего раздела является демонстрация работоспособности теории LP-структур на примере их основного вида. Созданный класс LPStructure может быть реорганизован в абстрактный класс и применен в качестве основы проектирования иерархии классов, охватывающих теорию LP-структур целиком. В общем случае задача реализации всех рассмотренных в настоящей работе видов LP-структур (и, тем более, не рассмотренных) слишком объемна. Она может составить предмет отдельного исследования.

[1] Следствие 2 из п. 2.4.

Как уже отмечалось в п.6.2, для хранения в памяти элементов решетки в контексте подходов, принятых в настоящей работе, применяются битовые векторы. В классе LPStructue используется гибкая структура битового вектора – его длина может настраиваться «клиентской» программой. Размерность такого вектора задается двумя параметрами – длиной в байтах одного кластера (*eItemSize*) и количеством этих кластеров (*eLengthItems*).

Величина *eItemSize* определяет размер кластера – части вектора, которая обрабатывается одной логической операцией языка C++. Эта величина хранится как статическое константное поле класса и может быть изменена с его перекомпиляцией. Выбором $eItemSize = 1$ можно структурировать битовый вектор в виде последовательности байтов, что приводит к экономии памяти и делает структуру вектора более прозрачной с точки зрения понимания алгоритмов. Если положить, например, $eItemSize = 4$, то можно повысить быстродействие операций над векторами, так как многие циклы в программе станут в 4 раза короче, а 32-разрядные операции окажутся более эффективными для ЭВМ с соответствующей длиной слова памяти.

Количество кластеров битового вектора хранится в целочисленном поле *eLengthItems.* Конструктор класса получает в качестве параметра соответствующую целую величину для настройки данного поля. Это обстоятельство предоставляет возможность «клиентской» программе при создании LP-структуры динамически настраивать длину битовых векторов, исходя из размера решетки и, соответственно, объема имеющейся базы знаний.

В классе определены типы *Elem* и *Pair* путем переименования соответственно типов *ULONG* и *ULONGLONG.* Тип *Elem* предполагает хранение адреса элемента решетки, а именно – соответствующего элементу битового вектора. Тип *Pair* содержит смежную пару величин *Elem*, то есть предназначен для хранения пары элементов (точнее, их адресов) из некоторого бинарного отношения на решетке. Конечно, типы *Elem* и *Pair* также можно было бы реализовать в виде классов с перегруженными операторами (|, &, <=), соответствующими операциям на решетке. Однако было решено использовать простейшие типы (*ULONG* и *ULONGLONG*), следуя принятой в п.6.1 стратегии экономии памяти.

Как уже отмечалось ранее, в данной разработке используется библиотека STL. В частности, широко применяются ее параметризованные контейнерные классы *vector* и *set*, а также их комбинации. Для хранения задаваемых на решетке бинарных отношений общего вида определен тип *PairContainer* как $set<Pair>$. Некоторыми алгоритмами в качестве промежуточных данных используются векторы пар – *PairVector* ($vector<Pair>$). Имеется также тип *ElemContainer* ($set<Elem>$) для представления множеств элементов решетки. Каноническое бинарное отношение в соответствии с изложением п. 6.2 представляется вектором множеств *ElemContainerVector* ($vector<$ $ElemContainer$ $*>$).

Таким образом, перечислены основные типы, используемые в классе LPStructure. Это сделано с целью дать представление о подходах, которые применялись при практической реализации LP-структур. Далее опишем основные функциональные возможности класса.

Класс LPStructure инкапсулирует перечисленные ниже основные методы.

Конструктор

```
LPStructure(const int aLen = 8);
```

в качестве параметра принимает значение целочисленного поля *eLengthItems* для настройки длины битовых векторов. Как следует из приведенного объявления, по умолчанию принята структура вектора в виде байтовых кластеров.

Ряд методов предназначен для обеспечения стандартных операций над множеством пар LP-структуры. К ним относятся следующие функции, назначение которых следует из их названий:

```
BOOL insert(const Pair pair);
BOOL insert(const Elem left, const Elem right);
Pair extract();
BOOL erase(const Pair pair);
BOOL erase(const Elem left, const Elem right);
void clear();
ULONGLONG count() const;
```

Несколько методов реализуют основную функциональность класса. Они имеют два общих параметра

```
const OnLPEvent onEvent = NULL,
const DWORD dwUser = 0.
```

Эти параметры обеспечивают возможность сообщения «клиенту» детальной информации о событиях, происходящих во время их работы. Параметр *onEvent* (если задан) содержит адрес функции обратного вызова – обработчика LP-событий. Параметр *dwUser*, как это принято во многих функциях WinAPI [205], содержит сопутствующую информацию «пользователя». Обработчик LP-событий обязан иметь следующий прототип:

```
typedef void (__cdecl *OnLPEvent)(const Pair aPair,
                                  const int nEventType,
                                  const DWORD dwUser);
```

Здесь параметр *aPair* содержит пару элементов решетки, связанную с произошедшим событием. Параметр *nEventType* передает код события, *dwUser* возвращает информацию «пользователя», которая может быть использована им по собственному усмотрению, например, для идентификации источников событий. Предусмотрены следующие виды событий (возможные значения параметра *nEventType*):

```
static const int etRedundant = 0; // Лишнее правило
static const int etInference = 1; // Элемент связи
static const int etPreImageCount = 2; // Кол-во прообразов
static const int etPreImage = 3; // Очередной прообраз
```

Следующие два метода предназначены для построения логической редукции LP-структуры на основе анализа логических связей пар элементов.

Функция

```
BOOL isLConnected(const Pair aPair,
                  const OnLPEvent onEvent = NULL,
                  const DWORD dwUser = 0);
```

определяет наличие логической связи для пары элементов, задаваемой параметром *aPair*. Остальные параметры описаны выше. Они позволяют «клиенту» получить структуру логической связи, если она будет обнаружена.

Функция

```
ULONGLONG lReductionLC (const OnLPEvent onEvent = NULL,
                        const DWORD dwUser = 0);
```

непосредственно строит логическую редукцию LP-структуры, обращаясь в процессе своей работы к функции *isLConnected*. Ее параметры позволяют при необходимости сообщить «клиенту» об обнаруженных «лишних» парах. В качестве результата она возвращает количество пар редуцированной LP-структуры.

Метод

```
void maxImage(const Elem eSource,
              const Elem eRes,
              const OnLPEvent onEvent = NULL,
              const DWORD dwUser = 0);
```

для заданного элемента решетки *eSource* вычисляет в LP-структуре его наибольший образ *eRes* (см. п.2.6 настоящей работы). Параметры *onEvent* и *dwUser* позволяют «клиенту» получать информацию о парах, которые вносят вклад в прямой логический вывод.

Метод

```
ULONG minPreImages(const Elem eDest,
                   const Elem aNegative = 0,
                   const INT aMaxCount = 0,
                   const OnLPEvent onEvent = NULL,
                   const DWORD dwUser = 0,
                   const int nOptimize = otMemory);
```

предназначен для нахождения решений продукционно-логических уравнений. Для заданного элемента решетки *eDest* он вычисляет минимальные прообразы, о которых сообщает вызывающей программе посредством параметров *onEvent* и *dwUser*. Параметр *nOptimize*, как отмечено в заключительной части п. 6.1, задает один из двух возможных режимов оптимизации работы – по памяти или по времени.

Два специальных параметра функции *minPreImages* носят более прикладной характер, чем остальные. Параметр *aNegative*, если задан, содержит объединение начальных точек решетки, которые должны быть принудительно исключены из процесса поиска прообразов. Указанные точки могут, например, соответствовать заведомо ложным фактам базы знаний. Учет их наличия позволяет сократить время работы алгоритма вычисления прообразов. Параметр *aMaxCount*, если не равен нулю, содержит целое значение, ограничивающее число искомых прообразов. Когда этот параметр положителен, при достижении указанного им количества найденных прообразов процесс прекращается, и найденные прообразы немедленно возвращаются «клиенту». Отрицательное значение параметра *aMaxCount*, точнее, его абсолютная величина, интерпретируется как период времени – количество миллисекунд, отведенных для вычисления прообразов. Параметры *aNegative* и *aMaxCount* позволяют пошагово решать продукционно-логическое уравнение, получая и обрабатывая решения ограниченными порциями. Данный способ используется алгоритмом кластерно-релевантного LP-вывода (см. ниже п. 6.4.4).

Итак, представлены основные публичные методы класса LP-Structure. Для их реализации используется также внутренняя функциональность, основное содержание которой описывается ниже.

Функция

```
void getLConnected(const Elem eDest, Elem eRes);
```

вычисляет объединение всех элементов решетки *eRes*, которые содержатся в прообразах заданного элемента *eDest*. При решении уравнения предварительный вызов данной функции позволяет существенно сократить множество обрабатываемых элементов в слоях исходного отношения.

Следующие два перегруженных[1] метода

```
void minPreImagesOne(const Elem eDest,
                     ElemContainer *ecRes);
void minPreImagesOne(const int nRItem,
                     const int nRBit,
                     ElemContainer *ecRes);
```

предназначены для нерекурсивного нахождения множества *ecRes* всех минимальных начальных прообразов заданной точки решетки в LP-структуре. В первом варианте функции исходная точка задается как элемент решетки *eDest*, во втором – номерами кластера *nRItem* и его разряда *nRBit* в соответствующем точке битовом векторе. Эти функции используются при решении продукционно-логических уравнений в режиме экономии памяти.

Функция

```
BOOL ecPursue(const int nItem, const int nBit);
```

рекурсивно находит множество всех минимальных начальных прообразов заданной точки решетки (*nRItem* и *nRBit*), а также всех промежуточных точек, участвующих в данном рекурсивном алгоритме. Результаты формируются в векторе множеств – соответстующем поле класса, имеющем тип *ElemContainerVector*.

Предыдущие функции в своей работе обращаются к двум методам, позволяющим соответственно суммировать прообразы (объединять их как элементы решетки) и пополнять множество прообразов, одновременно контролируя минимальность прообразов, то есть исключая из множества прообразы, содержащие другие прообразы (см. также п. 2.6).

```
void addPreImageMin(ElemContainer *ecSum,
                    ElemContainer *ecOne);
void insPreImageMin(ElemContainer *ecDst,
                    ElemContainer *ecSrc);
```

Ряд перечисленных далее несложных методов обеспечивает выполнение основных решеточных операций. Они на данном этапе описаны в секции *private*, однако могут быть перенесены в *public* при расширении функциональности класса LPStructure:

```
BOOL EQ(const Elem ls, const Elem rs); // (ls) == (rs)
BOOL LE(const Elem ls, const Elem rs); // (ls) <= (rs)
BOOL LT(const Elem ls, const Elem rs); // (ls) < (rs)
BOOL EZ(const Elem ls); // (ls) == 0
void lJoin(const Elem ls, const Elem rs); //(ls)|=(rs)
BOOL lDiff(const Elem ls, const Elem rs); //(ls)-=(rs)
```

[1] Перегруженными называются методы с одинаковыми именами, различающиеся списком параметров.

Поводя итог изложенному выше, отметим, что описана основная функциональность класса LPStructure, что также позволяет судить о принципах его построения. Отметим, что для обеспечения возможности использовать данный класс в других системах программирования его интерфейс был оформлен в виде динамической C-библиотеки LPStructure.dll. Названия и параметры ее функций соответствуют представленным выше публичным методам класса LPStructure, поэтому нет необходимости в их более подробном описании. Заметим лишь, что почти каждая из этих функций имеет дополнительный параметр

```
const HANDLE lps,
```

который фактически представляет адрес экземпляра LPStructure. Полный список функций библиотеки LPStructure.dll выглядит следующим образом:

```
typedef LPStructure::Pair Pair;
typedef LPStructure::OnLPEvent OnLPEvent;
typedef LPStructure::Elem Elem;

extern "C" // Чтобы имена в DLL были предсказуемыми
  {
  DLL_API HANDLE lpsCreate(const int nLen);
  DLL_API void lpsDelete(const HANDLE lps);

  DLL_API BOOL lpsInsert(const HANDLE lps, const Pair aPair);

  // Получить пару из множества (и исключить ее)
  DLL_API Pair lpsExtract(const HANDLE lps);

  DLL_API ULONGLONG lpsLReductionLC(const HANDLE lps,
                                    const OnLPEvent onEvent = NULL,
                                    const DWORD dwUser = 0);
  DLL_API BOOL lpsConnected(const HANDLE lps,
                            const Pair aPair,
                            const OnLPEvent onEvent = NULL,
                            const DWORD dwUser = 0);

  DLL_API void lpsMaxImage(const HANDLE lps,
                           const Elem eSource,
                           const Elem eRes,
                           const OnLPEvent onEvent = NULL,
                           const DWORD dwUser = 0);
```

```
DLL_API ULONG lpsMinPreImages(const HANDLE lps,
                              const Elem eDest,
                              const Elem aNegative = 0,
                              const INT aMaxCount = 0,
                              const OnLPEvent onEvent = NULL,
                              const DWORD dwUser = 0,
                              const int nOptimize = otMemory);

DLL_API void lpsFreeLPCode(const HANDLE lps,
                           const Elem eDest);
}
```

6.4. LPExpert – интегрированная среда логического программирования

В настоящем разделе кратко описываются архитектура и основные функциональные возможности разработанного автором пакета программ LPExpert. Он представляет собой интегрированную среду для создания и эксплуатации продукционных экспертных систем. В основу рассматриваемого пакета были положены концептуальные положения монографии [79]. Несмотря на то, что эта небольшая по объему книга носила в основном учебный характер, ее оригинальные идеи оказались востребованными. В дальнейшем они использовались и развивались в целом ряде работ по экспертным системам, в первую очередь – обучающим (например, [29, 84, 91]).

Пакет программ LPExpert реализован в системе программирования Delphi 5. Он оснащен многодокументным оконным интерфейсом пользователя и состоит из следующих основных блоков: редактор правил; компилятор правил; машина вывода. В последнее время к его функциональности была добавлена dll-библиотека – интерфейс описанного в п. 6.3 класса LPStructure, что значительно расширило возможности пакета в плане эффективности создания с его помощью баз знаний.

6.4.1. Структура базы знаний

Как уже отмечалось ранее, назначение пакета LPExpert – разработка и эксплуатация продукционных экспертных систем. Опишем поддерживаемую данным пакетом структуру базы знаний (БЗ). Она представляет собой существенное расширение структур БЗ, описанных в [79] и, впоследствии, в [91].

БЗ содержит знания эксперта в некоторой предметной области. Она включает известные факты, выраженные в виде объектов, атрибутов и условий. В пакете LPExpert принята запись факта в виде пары «*объект = значение*». Данное утверждение может быть верно лишь с некоторой вероятностью, понятие которой здесь заменено понятием *коэффициента доверия* – целым числом от 1 до 100. Коэффициент 100 соответствует достоверному утверждению. Некоторые объекты могут быть многозначными, то есть одновременно допускать несколько значений. С такими объектами в БЗ связаны списки значений.

С точки зрения экспертной системы объекты можно разделить на следующие три класса.

1) Объекты, значения которых невозможно вывести с помощью правил продукционной системы из значений других объектов. Их значения система может получить лишь непосредственно от пользователя или извлечь из имеющихся фактов. Назовем эти объекты *начальными.*

2) Объекты, значения которых выводятся с помощью правил, причем эти объекты имеют практический смысл для пользователя. Будем называть их *объектами экспертизы.*

3) Объекты, значения которых можно вывести, однако они недоступны пользователю и созданы разработчиком БЗ для улучшения ее структуры. Такие объекты называются *внутренними.*

Заметим, что теоретически в конкретных БЗ возможны «смешанные» объекты, некоторые значения которых выводимы из правил, а другие значения являются начальными. Наличие таких объектов можно рассматривать как один из признаков некорректности БЗ – потенциальную противоречивость. Действительно, при диалоговом выборе значения объекта пользователь может указать любое из предложенных, в том числе и такое, которое противоречит значению, выводимому из правила. Тем не менее, нетрудно преобразовать БЗ так, чтобы она не содержала «смешанных» объектов. Достаточно, например, для «начальных» значений $\{v_i, i = 1,\dots,n\}$ каждого «смешанного» объекта A ввести новый объект B с (начальными) значениями $\{u_i, i = 1,\dots n;'?'\}$, добавив также к БЗ совокупность правил $u_i \to v_i, i = 1,\dots n$. В результате объект A станет внутренним, а новый объект B окажется начальным. В силу данного обстоятельства, в описываемой модели БЗ начальными считаются лишь такие объекты, которые не имеют ни одного выводимого значения.

Во время считывания (компиляции) БЗ экспертная система автоматически производит классификацию объектов.

Для каждого начального объекта в БЗ обычно заготовлен вопрос, который в некоторый момент может быть задан пользователю с целью получения значения объекта. С такими объектами также связано понятие *разрешенных значений*, то есть фактически их областей определения. При ответе на заданный системой вопрос пользователь выбирает значение из предлагаемого списка, включающего набор разрешенных значений.

Важнейшей составной частью БЗ является совокупность правил (продукций), с помощью которых система делает выводы о значениях неначальных объектов.

Правило состоит из двух частей – предпосылки и заключения. Как предпосылка, так и заключение представляют собой подмножества фактов БЗ (указанного выше вида). Кроме того, каждый факт заключения может иметь коэффициент доверия, а при отсутствии заданного коэффициента факт по умолчанию считается достоверным.

Таким образом, БЗ экспертной системы содержит следующую информацию:

1) список объектов экспертизы;
2) список многозначных объектов;
3) список разрешенных значений для каждого начального объекта;
4) вопросы, соответствующие начальным объектам;
5) продукционные правила;
6) накопленные факты – *рабочее множество.*

Кроме того, элементы БЗ (объекты, значения, правила) могут содержать дополнительную информацию (комментарии, изображения) для улучшения дружественного характера пользовательского интерфейса.

Расширения возможностей пакета LPExpert основаны на представлении наборов фактов и правил базы знаний LP-структурой. Каждый единичный факт изображается точкой решетки – булеана. Предпосылка и заключение правила представляются двумя элементами решетки, а сами правила – упорядоченными парами бинарного отношения в LP-структуре. Очевидно, определяемое множеством правил бинарное отношение является рефлексивным и транзитивным. Дистрибутивность этого отношения легко следует из возможности вывода с его помощью совокупности фактов по частям.

6.4.2. Синтаксис базы знаний

Постоянное хранение баз знаний в системе организовано на внешнем запоминающем устройстве компьютера (как правило – жестком диске) в виде текстовых файлов (обычно – с расширением .exp). Для простоты будем называть их файлами правил.

База знаний некоторой предметной области (подобно «приложениям» в современных системах программирования) образует «проект» и может располагаться в нескольких файлах правил. Перечень имен файлов, относящихся к общему проекту, хранится в специальном файле проекта БЗ (обычное расширение .prj). В этом же файле содержатся комментарий к проекту и путь к базовой папке, в которой обычно располагаются все файлы правил данного проекта. Комментарием может также снабжаться каждый файл правил в отдельности. Фрагменты файлов правил БЗ приведены в Приложении А настоящей монографии. В полном варианте файлы доступны на личной web-странице автора http://expert.vrn.ru/SD/.

Файл правил – это текстовый файл, содержащий следующие виды строк.

`(эксп)<имя объекта>`

– задает объект экспертизы с именем <имя объекта>;

`(многозначный)<имя объекта>`

– задает многозначный объект;

`(зн)<имя объекта> = <знач 1>, <знач 2>, …, <знач N>`

– задает список разрешенных значений для начального объекта с именем <имя объекта>;

`(вопрос)<имя объекта> = <текст вопроса>`

– назначает вопрос начальному объекту;

`(карт)<имя объекта> = <имя файла>`

– назначает объекту изображение (картинку), содержащееся в заданном файле <имя файла>;

`(звук)<имя объекта> = <имя файла>`

– назначает объекту звуковой .wave-файл <имя файла>;

`(см)<имя объекта> = <текст комментария>`

– назначает комментарий («смысл») произвольному объекту.

Указанные строки могут располагаться в произвольном порядке. В строках допускаются незначащие пробелы. Все ключевые слова языка БЗ (*эксп*, *многозначный*, *зн* и другие) являются регистронезависимыми.

Некоторые из перечисленных возможностей введены для улучшения дружественного характера пользовательского интерфейса экспертной системы. При просмотре объектов и их значений конечному пользователю доступны комментарии и изображения объектов с возможным звуковым сопровождением. Такие возможности позволяют ему лучше ориентироваться в предметном содержании БЗ.

Комментарии могут занимать несколько строк, в этом случае их синтаксис представляется следующим образом:

```
(см)<имя объекта>
<строки комментария>
#
```

Возможен также «объектно-ориентированный» стиль описания объектов БЗ. В этом стиле объект можно описать представленным далее блоком строк:

```
(объект) <имя объекта>
<строки описания объекта>
кобъект
```

Здесь все <строки описания объекта> относятся к заданному общему объекту и могут быть следующими:

```
эксп
многозначный
зн = <знач 1>, <знач 2>, …, <знач N>
вопрос = <текст вопроса>
карт = <имя файла>
звук = <имя файла>
см = <текст комментария>
```

(или многострочный вариант).

При использовании объектно-ориентированного стиля возможны также комментарии к *значениям* объекта:

```
(см)<имя значения> = <текст комментария>.
```

Кроме описанных строк, файл правил содержит также собственно *правила,* каждое из которых имеет следующий вид:

```
прав(<любой текст 1>) <любой текст 2>
<предпосылка>
то
<заключение>.
```

Здесь <любой текст 1> – это идентификатор данного правила, обычно – числовой номер, после которого в качестве <любой текст 2> принято записывать «: если» (см. примеры в Приложении А).

Раздел <предпосылка> может содержать одну или несколько строк вида

```
<объект> = <значение>,
```

которые могут закачиваться незначащим для БЗ союзом 'и'.

Раздел <заключение> содержит одну или несколько строк вида

```
<объект> = <значение>, кд = <n>,
```

которые также могут закачиваться незначащим союзом 'и'. Здесь <n> – коэффициент доверия данного утверждения (от 1 до 100). Конструкция [, кд=<n>] может быть опущена, и в данном случае считается кд = 100.

Правило может содержать многострочный комментарий. В этом случае оно заканчивается не точкой, а двоеточием, и все последующие строки до ближайшего символа ‘#’ будут отнесены к комментарию. Комментарии к правилам опционально выдаются на экран по желанию пользователя и обычно содержат предметные пояснения (например, ссылки на литературу или законодательные акты).

6.4.3. Структура пакета LPExpert и принципы реализации

Пакет программ LPExpert состоит из следующих основных блоков:

- редактор правил;
- компилятор правил,
- машина вывода,
- модуль интерфейса класса LPStructure.

Значительную часть в его архитектуре занимает также оконный интерфейс пользователя.

Редактор предназначен для подготовки файлов правил в текстовом режиме. Он предоставляет все возможности стандартного текстового редактора, включая опции «вырезать», «скопировать», «вставить», а также поиск, автозамену текста, индикацию текущих координат курсора. Возможен предварительный просмотр, а также печатание документов и установки настроек принтера. Одна из существенных возможностей встроенного редактора – взаимодействие с компилятором правил и выделение текста, содержащего ошибку.

Другая группа возможностей редактора правил связана с просмотром хранящейся в памяти двоичной информации – откомпилированной БЗ в виде объектов, значений и правил. Такие возможности оказывают существенную помощь при отладке БЗ.

Компилятор содержит блоки лексического и синтаксического анализа.

При запуске компиляции проекта база знаний (один или несколько файлов правил) считывается в динамическую оперативную память, при этом БЗ переводится во внутреннее двоичное представление. При обнаружении во время компиляции ошибки открывается соответствующий исходный файл в редакторе. В нем выделяется содержащая ошибку строка и пользователю сообщается о характере ошибки.

Двоичное хранение БЗ в памяти реализуется путем широкого использования динамических структур, в основном – коротких строк и индексированных списков. Следует отметить, что применение таблиц имен и, соответственно, использование двубайтовых индексов вместо самих имен в объектах, значениях и правилах позволило значительно уменьшить объем занимаемой памяти по сравнению с реализацией, предложенной в [79]. Высокая скорость считывания множества правил при этом практически сохранена.

Ядром экспертной системы является ее *машина вывода*, которая способна, используя факты БЗ, интерпретировать правила для решения поставленной задачи – определения значений заданного объекта экспертизы. Этот процесс называется *экспертизой*. Она осуществляется на основе чистого обратного вывода – путем формулирования пробных гипотез и их рекурсивной проверки на соответствие указанной цели. При этом все факты, установленные в процессе вывода, автоматически заносятся в БЗ. Основной блок машины вывода реализован в виде рекурсивной процедуры.

Существенной особенностью реализованной в пакете LPExpert машины вывода является возможность для пользователя отказаться в очередном диалоге от выбора значения объекта. Вместо этого он имеет право сделать шаг назад к предыдущему диалогу, то есть отменить сделанный выбор значения предыдущего объекта. Во время вывода на экран также опционально выдается трассировка применяемых правил. Таким образом, можно констатировать, что имеются эффективные средства отладки разрабатываемых БЗ.

Заметим, что реализация указанной выше отмены выбора не является тривиальной. После сделанного пользователем выбора значения объекта вполне могли быть получены новые факты применением многочисленных продукций. Указанная возможность достигается путем построения дополнительных структур, содержащих хронологию логического вывода. В целях экономии памяти данная возможность является опциональной.

Модуль интерфейса *класса LPStructure* обеспечивает расширенные возможности пакета в плане разработки и исследования БЗ. Эти возможности включаются лишь при обнаружении на компьютере доступного файла LPStructure.dll. В этом случае данная библиотека динамически загружается, а модуль настраивает связи с ее функциями.

При наличии загруженной библиотеки LPStructure.dll по окончании компиляции БЗ для каждого продукционного правила в памяти создаются два соответствующих правилу двоичных вектора (предпосылка и заключение). Эти векторы (точнее, их адреса) в дальнейшем могут быть использованы при формировании LP-структуры. Общая размерность векторов (длина) вычисляется на основе результатов компиляции. Точками соответствующей решетки будут пары «*объект = значение*». Для создания LP-структуры модуль последовательными вызовами функции *lpsInsert* формирует набор пар логического отношения. Далее можно использовать все возможности класса LPStructure, а именно – выявлять избыточные правила, исследовать образы и прообразы заданных подмножеств фактов БЗ, производить обратный вывод на основе решений продукционно-логических уравнений. Имеется также возможность проверки множества фактов на непротиворечивость, то есть отсутствие в нем одновременно нескольких значений немногозначного объекта. Дополнительные возможности рассчитаны на опытного разработчика БЗ.

Поскольку пользователь пакета LPExpert работает в терминах базы знаний, то для обмена информацией с классом LPStructure модуль содержит также ряд интерфейсных функций. Прототипы некоторых из них (на языке Delphi) приводятся ниже (LP-кодом называется соответствующий подмножеству фактов двоичный вектор).

```
type
  TElem = ULONG; // Элемент LP-структуры
  TPair = Int64; // Пара отношения на LP-структуре
  TlpsCodeItem = BYTE; // Тип кластера LP-кода (можно увеличить)
  PlpsCode = ^TlpsCodeItem; // Тип указателя LP-кода
  TValuesArray = array of WORD; // Массив значений объектов

  // Обработчик LP-события
  TlpsEventHandler = procedure(const aPair: TPair;
                               const nEventType: Integer;
                               const dwUser: DWORD); cdecl;


// Создать LP-структуру
function lpsMake(const bNew: Boolean = True): Boolean;
// Найти правило по LP-паре
function RuleByLP(const aPair: TPair): PRule;
// Добавить/удалить значение объекта в LP-коде
procedure addLegalToLPCode(const lpsCode: PlpsCode;
                           const nValue: WORD;
                           const bNegate: Boolean = False);
// Прочитать значение объекта из LP-кода
function getValuesFromLPCode(const lpsCode: PlpsCode;
aValues: TValuesArray): WORD;
```

```
// Сравнить два LP-кода
function cmpLPCode(const lpsLCode, lpsRCode: PlpsCode): Integer;
// Проверить LP-код на непротиворечивость
function tstLPCode(const lpsCode: PlpsCode): Boolean;
// Проверить LP-код на непротиворечивость с заданным значением
function tstLPCodeLegal(const lpsCode: PlpsCode;
                        const nValue: WORD): Boolean;
// Выдать LP-код в виде строки нулей и единиц
function showLPCode(const lpsCode: PlpsCode): string;
```

Заметим, что в данной версии пакета LPExpert использование библиотеки LPStructure возможно лишь без учета *коэффициентов доверия*, которыми, как указано в п.п. 6.4.1–6.4.2, могут помечаться факты БЗ. Это обстоятельство объясняется тем, что теория LP-структур на сегодня пока не оперирует нечеткой продукционной логикой.

6.4.4. Релевантный LP-вывод

Существенная часть интерфейсного модуля пакета LPExpert, связывающего его с библиотекой LPStructure, посвящена реализации возможности LP-вывода в экспертной системе. Этот обратный вывод представляет собой альтернативу по отношению к стандартному методу описанной ранее машины вывода. Он осуществляется путем построения минимальных начальных прообразов (решения уравнений) в LP-структуре (см. также п. 2.6) для разрешенных значений объекта экспертизы с последующим их исследованием. В построенном множестве достаточно найти тот прообраз, который содержит лишь выполненные факты, тогда сразу можно будет сделать заключение о соответствующем значении объекта экспертизы. Самый простой путь в этом плане – просматривать прообразы последовательно, задавая пользователю вопросы о соответствующих начальных фактах (или обращаясь к базе данных за этими фактами).

В представленном подходе избран эффективный метод, который заключается в приоритетном просмотре прообразов, содержащих значения наиболее «релевантных» объектов. Таковыми в первую очередь считаются объекты, чьи значения присутствуют в максимальном количестве прообразов. Тогда единственный отрицательный ответ пользователя на заданный вопрос исключает из рассмотрения сразу большое количество прообразов, что, соответственно, ускоряет вывод. Второй фактор релевантности объекта – его присутствие в прообразах с минимальной мощностью. Таким образом, предпочтение отдается решениям, которые требуют меньшего количества вопросов пользователю (или обращений к внешней базе данных). Сочетая указанные два показателя релевантности, можно получить результаты, по эффективности существенно превышающие возможности обычной машины вывода. Указанный результат в первую очередь выражается в существенном снижении количества обращений за фактами к внешнему источнику (эти преимущества демонстрируются ниже в п. 6.4.6). Заметим, что в данной ситуации речь идет о статистическом показателе, поскольку и при обычном выводе присутствует некоторая вероятность случайного достижения лучшего минимального прообраза «с первого раза».

Еще одним важным свойством описанного выше метода *релевантного* LP-вывода является получение более полных результатов по сравнению с обычным выводом, при наличии противоречий в базе знаний. Стандартный процесс обратного вывода [79], пройдя по одному из путей от объекта экспертизы к начальным объектам, получает возможный в БЗ результат, попутно присваивая значения всем пройденным внутренним объектам. Еще один путь, проходящий через значения внутренних объектов, противоречащие найденным ранее, при такой организации вывода оказывается невозможным. Таким образом, в противоречивой базе знаний обычный обратный вывод может дать лишь часть возможных результатов, причем эта часть зависит от порядка просмотра правил. Отсюда следует, что при разных запусках экспертизы возможны *различные* неполные результаты. В то же время LP-вывод решает продукционно-логические уравнения на «алгебраическом» уровне без учета противоречий внутренних объектов. Такой подход даст *все* результаты, возможные в данной, пусть и противоречивой, базе знаний. Указанное обстоятельство, в противовес желаемому, может существенно увеличить количество внешних запросов. Однако, следует иметь в виду, что в данном случае это окажется приемлемой платой за более полные результаты вывода.

Эксперименты показали, что при больших объемах баз знаний (более 500 правил) и их достаточно «глубокой» структуре процесс одновременного построения *всех* минимальных прообразов может потребовать чрезмерного объема вычислительных ресурсов компьютера. В связи с данным обстоятельством описанный выше метод *релевантного* LP-вывода был модифицирован. *Кластерно-релевантный* LP-вывод предполагает последовательное построение кластеров начальных прообразов (подмножеств ограниченной мощности) с их динамическим релевантным исследованием (см. также описание параметров *aNegative* и *aMaxCount* функции *minPreImages* в п. 6.3). Получая очередную порцию прообразов (ограниченную параметром *aMaxCount*), модуль LP-вывода сразу исследует их на предмет «истинности», обращаясь, при необходимости, за фактами к внешнему источнику. Если при обработке очередного кластера «истинный» прообраз выявить не удается, процесс вычисления прообразов продолжается. При этом функции *minPreImages* посредством параметра *aNegative* передается информация об установленных на предыдущих шагах ложных фактах. На основе данной информации перед очередным процессом решения уравнения сужается множество актуальных правил. Последнее обстоятельство существенно ускоряет работу.

Следует иметь в виду, что «кластеризация» LP-вывода представляет собой некоторый промежуточный вариант между методами обычного и релевантного обратного вывода. Это означает определенный компромисс, при котором (по сравнению со стандартным методом обратного вывода) статистически снижается количество внешних запросов, но также не исключаются частичные результаты вместо полных результатов для противоречивых БЗ. Отсюда также следует, что множество результатов обычной экспертизы не всегда целиком содержится в множестве результатов кластерно-релевантного LP-вывода, в то время как при релевантном LP-выводе указанное вложение имеет место гарантированно.

Как показано далее в п.6.4.6 при описании результатов экспериментов, модифицированный LP-вывод дает достаточно эффективные результаты для промышленных БЗ больших размеров.

6.4.5. Функциональные возможности и интерфейс пользователя

Интерфейс пользователя интегрированной среды разработки и эксплуатации продукционных систем LPExpert построен в многодокументном стиле (MDI-приложение). Кратко опишем возможности пакета, перечислив основные пункты его главного меню и их назначение. Сразу отметим, что наиболее популярные пункты меню снабжаются также кнопками на панели быстрого доступа.

Меню *Файл* содержит полный набор стандартных опций, таких как *Новый*, *Открыть*, *Закрыть*, *Сохранить*, *Печать* и подобные им. Имена нескольких последних открытых файлов, как обычно, дополняют это меню. Пакет работает с двумя стандартными типами файлов – проектами (.prj) и файлами правил (.exp).

При открытии файла проекта (то есть разрабатываемого приложения БЗ) создается немодальное диалоговое окно, содержащее список имен файлов данного проекта (файлов правил с их атрибутами) и функциональные кнопки для работы с проектом. Здесь можно задать корневую папку проекта, добавить или удалить файлы правил, а также запустить компиляцию проекта. Можно открыть несколько проектов и работать с ними одновременно.

При открытии (или создании) файла правил формируется соответствующее окно текстового редактора. Главное меню программы при этом расширяется двумя типичными для редактора дополнительными меню *Редактор* (*Отменить*, *Вырезать*, *Копировать*, *Вставить*) и *Поиск* (*Найти*, *Заменить*, *Найти еще*, *Перейти к строке №*).

Пункт главного меню *Просмотр* открывает возможности просмотра находящейся в памяти откомпилированной БЗ (*Значения*, *Объекты*, *Правила*, *Статистика*). При выборе каждого пункта открывается соответствующий ему модальный диалог.

Диалог *Значения* содержит список всех теоретически возможных в данной БЗ фактов, а именно – пар вида «$\textit{объект} = \textit{значение}$». Пары, связанные с объектами экспертизы, выделяются цветом. Возможности просмотра значений зависят от доступности в системе библиотеки LPStructure.dll. При ее отсутствии никакой другой информации кроме списка в данном окне нет. Если библиотека была загружена, то кроме указанного списка возможных фактов, окно диалога содержит дополнительную функциональность – кнопки и области результатов. Можно выделить в списке любое подмножество фактов и для него вычислить как полный образ, так и все минимальные прообразы в создаваемой LP-структуре (см. также п.2.6 настоящей работы). Эти и ряд других представленных здесь возможностей оказывают существенную помощь при разработке, исследовании и оптимизации БЗ.

Диалог *Объекты* демонстрирует список всех содержащихся в памяти объектов, для каждого из которых можно просмотреть вопрос, разрешенные значения и фактические значения (полученные в результате экспертизы). Такая функциональность также оказывается весьма полезной при отладке БЗ. Двойной щелчок в списке на некотором объекте помечает его как главный объект, для которого впоследствии и проводится экспертиза.

Диалог *Правила* открывает список всех откомпилированных продукционных правил с разделением их предпосылок, заключений и комментариев. Если предварительно была загружена библиотека LPStructure.dll, то в данном окне доступен также флажок *Лишние*. При его установке создается LP-структура и с ее помощью производится обнаружение избыточных правил БЗ, в результате чего все такие правила выделяются цветом. Для каждого избыточного правила можно также посмотреть вывод, то есть набор правил, из которых выводится избыточное правило. Очевидно, данная возможность весьма полезна для оптимизации БЗ.

Наконец, диалог *Статистика* показывает количество находящихся в памяти значений, объектов и правил.

Пункт главного меню *Удаление* позволяет выгрузить из памяти хранящиеся там фактические значения объектов, правила или целиком всю БЗ. Потребность в частичном удалении информации может возникнуть в силу того, что пакет LPExpert позволяет динамически компилировать базы знаний. Можно, например, удалить из памяти одно множество правил и скомпилировать другое, сохранив объекты с их текущими значениями. Часто при отладке БЗ возникает необходимость отменить результат проведенной экспертизы перед новым ее запуском, для чего удаляются только значения объектов.

Меню *Работа* связано с возможностями подготовки и проведения экспертизы.

Пункт этого меню *Режимы* позволяет установить ряд настроек, влияющих на последующее поведение машины вывода. Режим *Объяснения* включает ориентированные на конечного пользователя объяснения процесса вывода, возможно, с использованием законодательных, литературных или иных достоверных источников информации. В указанных объяснениях используются подготовленные ранее комментарии к правилам. Режим *Отладка* также инициализирует выдачу дополнительной информации во время вывода. Однако следует отметить, что эта информация представляет собой полную трассировку процесса, которая будет полезна скорее разработчику БЗ, чем ее потребителю. Режим *Хронология* активизирует сохранение машиной вывода последовательности примененных правил и выбора значений объектов пользователем, что дает возможность возвратов при проведении экспертизы. Такая возможность чаще применяется при отладке БЗ, чем при ее эксплуатации. Важной является возможность для пользователя переопределять режимы динамически – в процессе экспертизы в диалоговом окне ответа на вопросы имеется соответствующая кнопка.

Отдельная вкладка диалогового окна режимов посвящена настройке параметров LP-вывода. Установленный флажок *Скорость* задает рекурсивное вычисление прообразов объекта экспертизы с запоминанием результатов обработки предыдущих слоев LP-структуры. В противном случае слои обрабатываются независимо друг от друга (см. также заключительный абзац п. 6.1). В то время как первый вариант повышает быстродействие LP-вывода, второй – способствует экономии памяти. Еще две опции предназначены для настройки кластерно-релевантного LP-вывода (см. п. 6.4.4). Параметр *Не более*, если не равен нулю, ограничивает размер вычисляемого на отдельном шаге кластера прообразов. Флажок *Прообразы* задает способ интерпретации этого параметра. Если флажок установлен, то размер кластера ограничивается заданным в параметре *Не более* количеством прообразов, в противном случае указанной величиной ограничивается время вычисления каждого кластера (в миллисекундах).

Пункт меню *Построить* доступен при активном окне проекта и запускает его компиляцию.

Пункт *Выполнить* запускает экспертизу (выполнение логической программы, процесс обратного вывода) для вычисления значения главного объекта экспертизы. Его можно выбрать в диалоге просмотра объектов, если он не единственен. В процессе вывода пользователь отвечает на вопросы, связанные с начальными объектами БЗ. Соответствующее диалоговое окно содержит подготовленные в файле правил вопрос и список возможных ответов. Выбор делается двойным щелчком на нужном значении либо одинарным щелчком и нажатием кнопки *Вперед*. Здесь также имеются кнопки *Назад* (отменяет предыдущий выбор, работает при установленном режиме *Хронология*), *Справка* (показывает исходный и текущий объекты), *Режимы* (динамическая переустановка режимов работы) и *Выход* (возможность досрочно прервать экспертизу). Нормальным образом процесс логического вывода заканчивается при нахождении значения объекта экспертизы либо, в случае отсутствия возможности такового, с выдачей окна результата.

Пункт *LP-вывод* доступен лишь при загруженной библиотеке LPStructure.dll. Он запускает экспертизу, но она реализуется не стандартным обратным выводом, а путем решения уравнений – на основе построения и «релевантного» исследования минимальных начальных прообразов в LP-структуре (см. п. 6.4.4). Данный механизм оказывается скрытым от пользователя, так как здесь используется тот же диалог ответов на вопросы, что и при обычном выводе, с сохранением его интерфейса.

Таким образом, пакет LPExpert представляет собой полноценную интегрированную систему логического программирования с новыми эффективными средствами автоматизации и отладки.

6.4.6. Исследование и оптимизация тестовых баз знаний

Для объективной демонстрации возможностей пакета LPExpert были выбраны три базы знаний, существенно отличающиеся по объему множества правил, предметной области, по назначению и периоду их создания. Автор настоящей монографии не несет ответственности за качество рассматриваемых баз знаний. Они используются «как есть» с целью показать возможности пакета по их исследованию и оптимизации на основе теории LP-структур.

В первую очередь в качестве демонстрационного примера рассмотрим базу знаний «Здоровье» из книги [79], записанную в расширенном синтаксисе пакета LPExpert (см. Приложение А.1 и http://expert.vrn.ru/SD/) Этот выбор сделан не только потому, что [79] послужила идейной основой для LPExpert, но и в силу популярности данной БЗ в других исследованиях по экспертным системам (в частности, см. [29]).

База знаний «Здоровье» предназначена для выработки советов пациенту на основе оценки состояния его здоровья. Она выдает прогноз продолжительности жизни человека исходя из полученной от него информации о ряде факторов, влияющих на эту продолжительность. Данная БЗ служит демонстрационным целям и не претендует на непосредственное применение в медицине, хотя и содержит 77 правил.

Заметим, что в русскоязычном переводе книги [79] в тексте указанной БЗ (благодаря возможностям просмотра в пакете LPExpert) была найдена опечатка, не влияющая на результаты представленных здесь исследований. Продукционное правило 2 в оригинале имело вид

```
прав(02): если
      относительный_вес = ниже_нормы
то
      зн = да.
```

Было обнаружено, что:

1) в последующем тексте БЗ значение *ниже_нормы* нигде не используется, а в заключениях (правила 15, 16, 20) имеется значение *недостаточный*,

2) значение *недостаточный* вообще не присутствует в предпосылках каких-либо правил,

С учетом этих обстоятельств, правило 2 заменено на следующее:

```
прав(02): если
      относительный_вес = недостаточный
то
      зн = да.
```

К тексту рассматриваемой БЗ также добавлены некоторые конструкции, демонстрирующие расширения синтаксиса в пакете LPExpert (более свободный формат; комментарии к объектам, значениям и правилам; объектно-ориентированный стиль записи). В остальном БЗ «Здоровье» используется «как есть», включая ее исходные недостатки. В частности, ее система правил неполна, в чем нетрудно убедиться, проводя разнообразные экспертизы – далеко не каждый вариант ответов пользователя приводит к корректному результату.

Таким образом, после создания в IDE LPExpert нового проекта (.prj) и добавления к нему файла с БЗ «Здоровье» (на http://expert.vrn.ru/SD/ – health.exp), запускается процесс компиляции полученной базы знаний (время компиляции исчисляется несколькими десятками миллисекунд). Затем выбирается меню *Просмотр | Правила*. В открывшемся диалоговом окне отображен список всех продукционных правил. В этом диалоге устанавливается флажок *Лишние*. В результате (мгновенно) в списке идентификаторы правил 13, 14 и 16 оказываются перерисованными красным цветом. Исследуем полученные результаты.

Как было справедливо замечено в [29], правило 17 данной БЗ является лишним, поскольку идентично правилу 14. Выделим в списке «красное» правило 14, при этом станет доступной кнопка *Вывод*. Нажав ее, в открывшемся тексте вывода действительно увидим ссылку на правило 17. В тексте БЗ правила 14 и 17 полностью совпадают.

Однако в работе [29] отмечалось лишь одно лишнее правило. Возникает естественный вопрос, почему же у нас их получилось больше? Выделяя в списке правило 13 и открывая его вывод, получаем ссылку на правило 21. Сопоставим эти два правила:

```
прав(13): если
       вес=85_или_больше и
       сложение=мелкое и
       пол=ж
то
       относительный_вес=излишний.

прав(21): если
       вес=85_или_больше и
       пол=ж
то
       относительный_вес=излишний.
```

Как легко заметить, при одинаковых заключениях правило 13 имеет более сильное условие предпосылки, то есть правило 13 выводится непосредственно из правила 21 и, соответственно, является избыточным. Аналогичным образом правило 16 сразу выводится из правила 15. Таким образом, по сравнению с работой [29], пакет LPExpert дополнительно обнаружил два избыточных правила.

С учетом изложенного можно констатировать, что удалось получить неплохой результат. Несмотря на избыточность правил 13 и 16, которую можно обнаружить визуальным сравнением, не так просто просмотреть все пары из 77 правил. Этот процесс окажется еще более трудоемким, если правил будет, например, 777. Вместе с тем теория LP-структур, на которой основано проведенное исследование, является строгой математической теорией. В соответствии с данным обстоятельством, при построении логической редукции должны обнаруживаться *все* избыточные правила. С целью подтверждения указанного свойства проведем эксперимент, который состоит из следующих этапов:

• в LPExpert выполним экспертизу объекта *продолжительность*, попутно запоминая вводимые пользователем ответы;

• добавим к БЗ новое правило, предпосылка которого построена из фактов, соответствующих введенным при экспертизе ответам пользователя, а заключение представляет собой результат выполненной экспертизы;

• скомпилировав новую БЗ, откроем просмотр и включим флажок определения избыточных правил.

Эксперимент должен показать новое правило среди «лишних».

Итак, запустим экспертизу (меню *Работа* | *Выполнить*).

В целях экономии места сразу перечислим факты, которые экспертная система сформирует в результате диалога с пользователем, в хронологическом порядке ввода ответов:

```
возраст = 25-55
пол = м
вес = 55-85
сложение = мелкое
холестерин = норма
соль = норма
курение = нет
раса = европеоидная
характер = агрессивный
потребление_алкоголя = умеренное
```

Полученный результат экспертизы таков:

```
продолжительность = 62.
```

Далее ликвидируем базу знаний из памяти (*Удаление* | *База знаний*). Откроем редактирование файла правил и наберем следующее правило:

```
прав(78): если
возраст = 25-55 и
пол = м и
вес = 55-85 и
сложение = мелкое и
холестерин = норма и
соль = норма и
курение = нет и
раса = европеоидная и
характер = агрессивный и
потребление_алкоголя = умеренное
то
продолжительность = 62.
```

Затем, как и выше, скомпилируем БЗ, откроем просмотр правил и установим флажок *Лишние*. В результате (без видимых задержек) получим, что красным цветом дополнительно будет выделено правило 78. Открыв его вывод, увидим полный перечень идентификаторов правил (09, 43, 45, 47, 19, 27, 29, 04, 01, 37, 40, 52, 54, 66), из которых выводится избыточное правило 78. Таким образом, эксперимент показывает, что логическая редукция LP-структуры дает результат и при сложной логической зависимости подмножеств фактов.

Другое направление формальных исследований БЗ связано с рассмотрением минимальных прообразов значений объекта экспертизы.

Исследуем результат проведенной выше экспертизы, построив минимальные прообразы факта *продолжительность = 62*. С этой целью после компиляции БЗ откроем меню *Просмотр* | *Значения* – на экране появится модальный диалог просмотра разрешенных значений объектов. Далее в списке значений пометим указанный выше факт флажком. Затем выберем вкладку «прообраз», а в панели управления установим флажок *Быстро*. Наконец, здесь же нажмем кнопку *Найти*. В результате на вкладке прообразов будет выдано все их множество, общим количеством 49.

Итак, рассматриваемый факт *продолжительность = 62* имеет 49 минимальных прообразов. Если все факты хотя бы одного из них выполнены, то соответственно имеет место и факт *продолжительность = 62*. Путем просмотра минимальных прообразов (для этого есть и автоматизированные средства) можно убедиться в том, что ни один из них не совпадает с полученным от пользователя множеством фактов (предпосылкой нового правила 78). Наиболее близкий к нему минимальный прообраз содержит на один факт меньше, а именно – в нем отсутствует значение *раса = европеоидная*. Таким образом, определяемый предпосылкой правила 78 прообраз факта *продолжительность = 62* не является минимальным, поскольку данный факт в исходной БЗ может быть получен без значения *раса = европеоидная.* Следовательно, во время экспертизы один из заданных пользователю вопросов оказался лишним. Это не ошибка, поскольку машина вывода во время работы пытается вывести и другие значения объекта *продолжительность*, ведь заранее неизвестно, какое из них окажется истинным. Избыточность вопросов свидетельствует лишь о неоптимальной последовательности вывода.

Отсюда возникает идея о применении для экспертизы LP-вывода. Удалим базу знаний из памяти, затем в ее тексте ликвидируем новое правило 78. Далее скомпилируем БЗ и запустим LP-вывод (*Работа* | *LP-вывод*).

Как и выше, сразу перечислим факты, которые экспертная система сформирует на основании диалога с пользователем в порядке ввода им ответов (предполагается, что пользователь не менял ответы на аналогичные вопросы):

```
вес = 55-85
сложение = мелкое
холестерин = норма
соль = норма
пол = м
возраст = 25-55
курение = нет
старт = да
```

Результат соответствует предыдущей экспертизе, а именно –

```
продолжительность = 62.
```

На этот раз он получен, во-первых, действительно из *минимального* прообраза, во-вторых, из прообраза минимальной мощности. Таким образом, экспериментально показано, что применение LP-вывода способно существенно сократить количество обращений к пользователю (вместо 10 вопросов на этот раз было задано 8, с тем же результатом).

Еще одна возможность применения LP-структур состоит в проверке БЗ на противоречивость. С этой целью можно, например, пытаться строить минимальные прообразы для пар значений объекта *продолжительность.* Наличие хотя бы одного непустого прообраза будет означать «возможность» одновременного вывода двух несовместимых значений, то есть противоречивость данной БЗ.

Откроем диалог просмотра значений БЗ. Будем помечать флажками одновременно по два значения объекта *продолжительность* и, как указано выше, вычислять соответствующие минимальные прообразы. После некоторых безуспешных попыток (пустые результаты) оказывается, что множество фактов $\{$*продолжительность* $= 48$, *продолжительность* $= 72\}$ имеет более 100 минимальных прообразов. Вряд ли БЗ можно назвать корректной, если при одной и той же исходной информации она способна выдать два прогноза продолжительности жизни с разницей в 24 года!

С учетом изложенного выше, основные результаты проведенных экспериментов таковы:

• в тестовой БЗ выявлены все лишние правила, в том числе специально добавленное правило со сложным выводом;

• путем исследования прообразов обнаружено, что при обычном обратном выводе пользователю задаются лишние вопросы;

• с применением LP-вывода получен аналогичный результат экспертизы, при этом пользователю было задано вопросов на 20 % меньше;

• анализ прообразов несовместимых фактов показал противоречивость тестовой БЗ, что содержит потенциальную угрозу получения заведомо некорректных результатов экспертиз.

В качестве еще одного примера приведем БЗ промышленного уровня «Электрики», разработанную А.В.Засыпкиным в рамках исследований по интеллектуализации систем управления охраной объектов [91]. Базы знаний данной серии [113] в 90-е годы были внедрены управлением вневедомственной охраны Воронежской области, а также Воронежским институтом МВД РФ. В Приложении А.2 приведен фрагмент текста БЗ «Электрики», на http://expert.vrn.ru/SD/ – полный вариант (electricians.exp).

Она предназначена для получения всесторонних оценок знаний электромонтеров, обслуживающих пункт централизованной охраны. Данная БЗ содержит 508 правил для получения оценок уровня знаний электротехники, электромонтажа, приборов охранной сигнализации и так далее. При ее проверке на избыточность лишними оказываются лишь два правила, которые просто совпадают с еще одним правилом. Дублирование текста вполне возможно при составлении такого количества правил. Заметим, что на осмысленное использование БЗ столь значительного объема способен лишь преподаватель или специалист в достаточно узкой области. Стандартная экспертиза с использованием БЗ «Электрики» предполагает ввод пользователем порядка 70–100 ответов на вопросы системы, а результаты получаются в виде примерно двух десятков значений многозначного объекта экспертизы *знания*. В связи с изложенным, будем проводить экспертизу знаний «наугад», давая на каждый вопрос первый из предлагаемых ответов и существенно не вдаваясь в смысл ее результатов. Количественные итоги стандартной экспертизы таковы: задано вопросов пользователю – 76, получено значений объекта *знания* – 13, включая оценку *основы_электротехники--два*.

Следующий запланированный шаг состоит в попытке провести для сравнения экспертизу на основе релевантного LP-вывода. Однако данная попытка оказывается несостоятельной, поскольку после запуска LP-вывода программа «виснет». При более подробном изучении данной ситуации путем выдачи отладочных данных выясняется, что процесс решения соответствующих объекту экспертизы логических уравнений требует обработки свыше 1 миллиарда слоев (см. п. 2.6). Такой объем данных оказывается не под силу обычному персональному компьютеру. Попробуем использовать для экспертизы кластеризованный LP-вывод. С этой целью в его настройках зададим ограничение размера кластера, например, величиной в 200 прообразов (установив также флажки *Скорость* и *Прообразы*). В результате новая экспертиза выполняется без видимых задержек, а ее количественные итоги оказываются следующими: задано вопросов пользователю – 57, получено значений объекта *знания* – 21, включая аналогичную предыдущей оценку *основы_электротехники--два*.

Аналогичным образом проведем новые две соответствующие экспертизы (на основе обычного обратного, затем – кластерно-релевантного вывода), выбирая каждый последний возможный ответ вместо первого. Итоги стандартной экспертизы таковы: задано вопросов пользователю – 70, получено значений объекта *знания* – 12. Итоги LP-вывода: задано вопросов пользователю – 59, получено значений объекта *знания* – 21.

Наконец, попробуем проверить БЗ «Электрики» на противоречивость. Оказывается, что пара значений объекта *знания основы_электротехники--три* и *основы_электротехники--четыре* имеет около 300 непротиворечивых прообразов, однако другие пары оценок данного показателя несовместимы, что с учетом объема БЗ можно считать неплохим результатом. Гораздо хуже обстоит дело с оценками знаний из серии «токов_охраны_спи», которые оказываются совместимыми по всему спектру, от «двойки» до «пятерки»: имеется 18 минимальных прообразов множества всех оценок данного показателя.

Анализируя результаты экспериментов с БЗ «Электроника», в итоге можно констатировать, что они состоят в следующем:

- выявлены два лишних правила;
- использование для экспертизы релевантного LP-вывода невоз-

можно;

• при применении кластерно-релевантного LP-вывода снижение количества задаваемых пользователю вопросов составляет 15–25 %;

• при этом достигается увеличение числа выдаваемых результатов более чем на 60 %;

• данная БЗ содержит существенные противоречия.

Следующий тестовый пример составляет обучающая база знаний «Закон распределения». Эта БЗ была создана в 2008 г. по заданию автора студентами кафедры математического обеспечения ЭВМ Воронежского госуниверситета (см. Приложение A.3 и файл на http://expert.vrn.ru/SD/ – partition-law.exp). Она основывается на материале, представленном в учебнике [103] и предназначена для обучения пользователя небольшому разделу математической статистики. Точнее, цель этой базы знаний – демонстрационное выяснение закона распределения некоторой случайной величины по имеющимся у пользователя данным о ней. БЗ «Закон распределения» содержит 70 правил. Ее разработка осуществлялась в тот период, когда пакет LPExpert еще не содержал возможностей библиотеки LPStructure, соответственно не было возможности применения в этом процессе теории LP-структур.

При проверке данной БЗ на избыточность не удается обнаружить лишних правил. Попытки выявить в ней «совместимые» противоречия также оказываются безуспешными. Причины более высокого качества рассматриваемой БЗ можно увидеть, во-первых, в ее меньшей сложности. Во-вторых, следует иметь в виду, что она была составлена достаточно квалифицированными специалистами в области программирования.

Проведем в БЗ «Закон распределения» стандартную экспертизу, давая на каждый вопрос, например, последний из предлагаемых ответов. Ниже приводится «текстовая» интерпретация происходящего диалога «Система – Пользователь». Буквой «*С*» помечены вопросы системы, буквой «*П*» – ответы пользователя.

С. Выберите способ анализа закона распределения.

П. Параметрический

С. Для генеральной совокупности какого типа будет проверяться гипотеза о законе распределения?

П. Дискретный тип.

С. Чему равен объем выборки (количество ее элементов)?

П. 50 или более.

С. Выберите критерий согласия.

П. Хи-квадрат (Пирсона).

С. Гипотезу о каком из законов распределения хотели бы проверить?

П. Пуассона.

С. Выберите уровень значимости.

П. 1/100.

С. Выпонено ли условие: значение статистики хи-квадрат (Пирсона) меньше или равно квантилю распределения?

П. Нет.

С. Закон распределения = Пуассона, с уровнем значимости 1/100, подтверждения нет.

Проведем аналогичную экспертизу на основе релевантного LP-вывода. Получим следующий диалог.

С. Выберите способ анализа закона распределения.

П. Параметрический

С. Выберите критерий согласия.

П. Хи-квадрат (Пирсона).

С. Чему равен объем выборки (количество ее элементов)?

П. 50 или более.

С. Выберите уровень значимости.

П. 1/100.

С. Гипотезу о каком из законов распределения хотели бы проверить?

П. Пуассона.

С. Выпонено ли условие: значение статистики хи-квадрат (Пирсона) меньше или равно квантилю распределения?

П. Нет.

С. Закон распределения = Пуассона, с уровнем значимости 1/100, подтверждения нет.

Нетрудно заметить, что при использовании второго экспертизы варианта были заданы те же вопросы, но в другом порядке. При этом пропущен один вопрос (о «типе генеральной совокупности»). Таким образом, при рассмотрении более качественной базы знаний LP-вывод также позволяет сэкономить время процессора.

Итак, в настоящем подразделе был продемонстрирован ряд важных и полезных на практике возможностей теории LP-структур в плане исследования, оптимизации и верификации баз знаний. Эти возможности, в частности, обеспечивают:

- обнаружение всех лишних правил;
- существенное снижение внешних запросов;
- увеличение числа выдаваемых результатов;
- выявление противоречий.

6.5. Моделирование математических знаний

Содержание данного раздела иллюстрирует возможности применения теории LP-структур для алгебраической формализации математических знаний. В качестве примера таких знаний выбрана разработанная автором теория весовых псевдодифференциальных операторов (ВПДО) и некоторые полученные с ее помощью новые результаты. Поскольку данная теория относится к области дифференциальных уравнений, необходимая для этого раздела ее часть в полном объеме представлена в приложении B. В настоящем разделе будут использоваться лишь формулировки теорем со ссылками на приложение B.

В качестве результата предполагается построение LP-структуры первого порядка (элементов полной решетки Линденбаума–Тарского и пар элементов логического отношения), которая формализует математические знания, описанные в Приложении B. Небольшой пример в этом плане уже был приведен в п. 1.5. Очевидно, что при введении обозначений предикатов глубина детализации логических формул может быть различной. В контексте целей настоящего раздела будем стремиться к такому уровню изложения, который позволил бы при минимальном наборе предикатов адекватно отразить в виде логического бинарного отношения все сформулированные в данном приложении теоремы.

Нетрудно заметить, что при решении подобной задачи возможны два подхода, каждый из которых связан со своей стратегией продукционно-логического вывода. Во-первых, можно отталкиваться от исходных аксиом (в нашем случае таковыми могут считаться условия 1–3) с их «решеточной» формализацией и далее двигаться по тексту математической работы к теоремам, вводя постепенно новые обозначения предикатов как элементов полной решетки Линденбаума-Тарского. Этот вариант соответствует прямому продукционно-логическому выводу и, по-видимому, более адекватно отражает реальные процессы математических исследований. Второй путь связан с обратным выводом и является более экономичным, так как позволяет сосредоточиться на формализации лишь некоторых избранных теорем.

В силу того обстоятельства, что целью настоящего раздела является не автоматизация реальных исследований, а иллюстрация возможностей применения теории LP-структур, выберем именно второй вариант решения поставленной задачи.

Начнем указанный процесс с того, что запишем каждую теорему из разделов B.1–B.4 в форме продукционного правила. Принимая во внимание присутствие сформулированных условий 1–3 почти в каждой теореме, с целью сокращения записи не будем на этом этапе раскрывать их содержание. На следующем шаге для предпосылок и заключений полученных правил введем обозначения соответствующих предикатов, что равносильно построению элементов полной булевой решетки $\mathbb{F}$. Тогда каждому правилу будет соответствовать пара логического бинарного отношения на $\mathbb{F}$, что и означает завершение построения LP-структуры.

Итак, рассмотрим теорему B.2.1. Соответствующее ей продукционное правило можно записать следующим образом (запятая в предпосылке заменяет операцию конъюнкции):

правило 2.1:

если

условия 1–2, $q \in R$, $M \geq 0$, $\varphi \in S^M$

то

$P_{\alpha,q}\varphi \in S^M$.

В смысле архитектуры продукционных систем это правило содержит несколько переменных (q, M, φ, $P_{\alpha,q}$) с бесконечными областями определения. Как следствие, фактически оно представляет собой бесконечное множество правил разрабатываемой продукционной базы знаний.

Рассматривая далее теорему B.2.2, легко заметить, что при попытке ее записи в форме продукции не удается обойтись без использования кванторов. Дело в том, что утверждение теоремы «оператор $P_{\alpha,q}$ ограничен из H_α^{s+q} в H_α^s» формально означает справедливость неравенства $\| P_{\alpha,q}u \|_{s,\alpha} \leq C\| u \|_{s+q,\alpha}$, в котором константа $C > 0$ – одна и та же для любой функции $u \in H_\alpha^{s+q}$. При этом данная константа вполне может зависеть от величин $q, s \in R$. Таким образом, для теоремы B.2.2 имеем другой тип правила, а именно –:

правило 2.2:

если

условия 1–2, $q \in R$, $s \in R$

то

$\exists C > 0, \forall u \in H_\alpha^{s+q} : \| P_{\alpha,q}u \|_{s,\alpha} \leq C\| u \|_{s+q,\alpha}$.

Как и правило 2.1, данная продукция также порождает бесконечное множество простых (без переменных $q, s \in R$) правил базы знаний.

Чтобы получить правило для теоремы B.2.3 (и некоторых других последующих теорем), введем обозначения:

$Sym(P_{\alpha,q})$ – символ ВПДО $P_{\alpha,q}$ (принадлежащий классу Φ_q);

$Sym_0(P_{\alpha,q})$ – главный символ ВПДО $P_{\alpha,q}$ с символом $\lambda \in \Phi_q$.

Тогда можно записать следующее продукционное правило, отражающее содержание теоремы B.2.3:

правило 2.3:

если

условия 1–2, $q_1 \in R$, $q_2 \in R$

то

$$Sym_0(P_{\alpha,q_1}P_{\alpha,q_2}) = Sym_0(P_{\alpha,q_1}) \cdot Sym_0(P_{\alpha,q_2}) .$$

Правила для оставшихся основных утверждений из разделов В.3–В.4 приведем почти без дополнительных комментариев, так как эти продукции получаются по аналогии с рассмотренными выше. Отметим, что нумерация правил соответствует нумерации теорем.

правило 3.1:

если

условия 1–3, $q \in R$, $s \in R$

то

$$\exists C > 0, \forall u \in H_\alpha^{s+q} : \left\| P_q\left(t, \alpha D_t\right) u - P_{\alpha,q}\left(t, D_{\alpha,t}\right) u \right\|_{s,\alpha} \le C \left\| u \right\|_{s+q-1,\alpha} .$$

правило 3.2:

если

условия 1–3, $q_1 \in R$, $q_2 \in R$, $s \in R$,

$Sym(P_{\alpha,q_1+q_2}) = Sym(P_{\alpha,q_1}) \cdot Sym(P_{\alpha,q_2})$

то

$$\exists C > 0, \forall u \in H_\alpha^{s+q_1+q_2} : \left\| P_{q_1}\left(t, \alpha D_t\right) P_{q_2}\left(t, \alpha D_t\right) u - P_{q_1+q_2}\left(t, \alpha D_t\right) u \right\|_{s,\alpha} \le$$

$$\le C \left\| u \right\|_{s+q_1+q_2-1,\alpha} .$$

Для теорем В.3.3–В.3.4 лучше записать по два правила, так как в этом случае они будут содержать меньшее количество кванторов:

правило 3.3(1):

если

условия 1–3, $s + q \ge 1$, $\sigma > 1$, $\left[(s+q)/\sigma\right] \le M + 1$

то

$$\exists C > 0, \forall \varphi \in S : \left\| P_q\left(t, \alpha, D_t\right) \varphi - P_{\alpha,q}\left(t, D_{\alpha,t}\right) \varphi \right\|_{s,(\alpha,\sigma)} \le$$

$$\le C \left\| \varphi \right\|_{s+q-1,(\alpha,\sigma)} ,$$

правило 3.3(2):

если

условия 1–3, $s + q \ge 1$, $k \in \overline{\mathbb{Z}}^+$, $k \le M + 1$

то

$$\exists C > 0, \forall \varphi \in S : \left\| P_q\left(t, \alpha D_t\right) \varphi - P_{\alpha,q}\left(t, D_{\alpha,t}\right) \varphi \right\|_{s,\alpha,k} \le C \left\| \varphi \right\|_{s+q-1,\alpha,k} .$$

правило 3.4(1):

если

условия 1–3, $s + q_1 + q_2 \ge 1$, $\sigma > 1$, $\left[(s+q_1+q_2)/\sigma\right] \le M + 1$,

$$Sym(P_{\alpha,q_1+q_2}) = Sym(P_{\alpha,q_1}) \cdot Sym(P_{\alpha,q_2})$$

то

$$\exists C > 0, \forall \varphi \in S : \left\| P_{q_1}\left(t, \alpha D_t\right) P_{q_2}\left(t, \alpha D_t\right)\varphi - P_{q_1+q_2}\left(t, \alpha D_t\right)\varphi \right\|_{s,(\alpha,\sigma)}$$

$$\le C \left\|\varphi\right\|_{s+q_1+q_2-1,(\alpha,\sigma)} .$$

правило 3.4(2):

если

условия 1–3, $s + q_1 + q_2 \ge 1$, $k \in \overline{\mathbb{Z}}^+$, $k \le M + 1$

то

$$\exists C > 0, \forall \varphi \in S : \left\| P_{q_1}\left(t, \alpha D_t\right) P_{q_2}\left(t, \alpha D_t\right)\varphi - P_{q_1+q_2}\left(t, \alpha D_t\right)\varphi \right\|_{s,\alpha,k}$$

$$\le C \left\|\varphi\right\|_{s+q_1+q_2-1,\alpha,k} .$$

Наконец, теоремы раздела B.4 могут быть представлены следующими продукциями:

правило 4.1:

если

условия 1–3, $s \ge 0$, $q \in \mathbb{Z}^+$

то

$$\exists C > 0, \forall u \in H_\alpha^{s+q} : \left\| M_q\left(\alpha D_t\right)u - \Lambda_\alpha^q\left(D_{\alpha,t}\right)u \right\|_{s,\alpha} \le C \left\| u \right\|_{s+q-1,\alpha} .$$

правило 4.2:

если

условия 1–3, $s \ge 0$, $q \in \mathbb{Z}^+$, $\sigma > 1$, $\left[(s+q)/\sigma\right] \le M + 1$

то

$$\exists C > 0, \forall u \in H_{\alpha,\sigma}^{s+q} : \left\| M_q\left(\alpha D_t\right)u - \Lambda_\alpha^q\left(D_{\alpha,t}\right)u \right\|_{s,(\alpha,\sigma)} \le C \left\| u \right\|_{s+q-1,(\alpha,\sigma)} .$$

правило 4.3:

если

условия 1–3, $s \ge 0$, $q \in \mathbb{Z}^+$, $k \in \overline{\mathbb{Z}}^+$, $k \le M + 1$

то

$$\exists C > 0, \forall u \in H_\alpha^{s+q,k} : \left\| M_q\left(\alpha D_t\right)u - \Lambda_\alpha^q\left(D_{\alpha,t}\right)u \right\|_{s,k,\alpha} \le C \left\| u \right\|_{s+q-1,k,\alpha} .$$

Переходим к следующему шагу построения LP-структуры – введем обозначения логических формул, соответствующих предпосылкам и заключениям разработанных выше правил.

Начнем с условий 1–3, которые имеются в предпосылке каждого правила. Не будем использовать представленную в качестве примера в разделе 1.5 формализацию условий 1–2, так как она является слишком частной и может привести к нежелательному «размножению» используемых предикатов.

Введем множество предикатных символов $R_{Op}(\alpha,\beta)$, представляющих бинарные отношения вида $\alpha Op \beta$. Например, $R_{=}(\alpha,\beta)$, $R_{>}(\alpha,\beta)$ соответствуют утверждениям $\alpha=\beta$ и $\alpha>\beta$. Для ряда формул потребуется также предикатный символ $R_{\in}(u,H)$, соответствующий утверждению о принадлежности $u\in H$.

Условие 1 может быть представлено следующими элементами решетки Линденбаума–Тарского, которые порождаются различными функциями α: $C_1(\alpha)=\bigwedge\limits_{t\le 0} R_{=}(\alpha(t),0)\wedge\bigwedge\limits_{t>0} R_{>}(\alpha(t),0)\wedge\bigvee\limits_{c>0}\bigwedge\limits_{t\ge d>0} R_{=}(\alpha(t),c)$.

Алгебраическая запись условия 2 будет выглядеть следующим образом: $C_2(\alpha,M)=R_{\in}(\alpha|_{R^+}, C^{\infty}(\bar{R}^+))\wedge R_{\in}(\alpha, C^{M+1}(R))$. Заметим, что это условие нельзя заменить выражением $\bigvee\limits_{M\in\bar{\mathbb{Z}}^+} C_2(\alpha,M)$, поскольку указанное в условии значение M используется некоторыми теоремами.

По аналогии с предикатами в условии 1, будем использовать также символ $R_{\le}(\alpha,\beta)$, означающий неравенство $\alpha\le\beta$. Заметим, что справедливо $R_{\le}(\alpha,\beta)=R_{<}(\alpha,\beta)\vee R_{=}(\alpha,\beta)$. При этом условие 3 будет представлено элементом решетки $C_3(\alpha)=\bigvee\limits_{C>0}\bigvee\limits_{N\in\mathbb{Z}^+}\bigwedge\limits_{0<\theta<t<+\infty} R_{\le}(\alpha^N(\theta), C\alpha(t))$.

Далее рассмотрим элементы решетки, которые позволят сформировать пары логического отношения для представления разработанных выше правил базы знаний.

Благодаря правилу 2.1, формируемое отношение будет содержать пары вида $(C_1(\alpha,M)\wedge C_2(\alpha)\wedge R_{\in}(q,R^1)\wedge R_{\in}(M,\bar{\mathbb{Z}}^+)\wedge R_{\in}(\varphi,S^M)$, $R_{\in}(P_{\alpha,q}\varphi,S^M))$ для всевозможных значений M, а также функций α и ВПДО $P_{\alpha,q}$.

Правило 2.2 порождает следующую совокупность пар логического отношения:

$$(C_1(\alpha)\wedge C_2(\alpha,M)\wedge R_{\in}(q,R^1)\wedge R_{\in}(s,R^1), \bigvee_{C>0}\bigwedge_{u\in H_{\alpha}^{s+q}} R_{\le}(\left\|P_{\alpha,q}u\right\|_{s,\alpha}, C\left\|u\right\|_{s+q,\alpha}).$$

Соответствующее правилу 2.3 множество пар элементов решетки будет выглядеть как

$$\begin{gathered}(C_1(\alpha)\wedge C_2(\alpha,M)\wedge R_{\in}(q_1,R^1)\wedge R_{\in}(q_2,R^1),\\ R_{=}(Sym_0(P_{\alpha,q_1}P_{\alpha,q_2}), Sym_0(P_{\alpha,q_1})\cdot Sym_0(P_{\alpha,q_2}))).\end{gathered}$$

Аналогично формируются остальные пары LP-структуры:

правило 3.1:

$$\begin{gathered}(C_1(\alpha)\wedge C_2(\alpha,M)\wedge C_3(\alpha)\wedge R_{\in}(q,R^1)\wedge R_{\in}(s,R^1),\\ \bigvee_{C>0}\bigwedge_{u\in H_{\alpha}^{s+q}} R_{\le}(\left\|P_q\left(t,\alpha D_t\right)u-P_{\alpha,q}\left(t,D_{\alpha,t}\right)u\right\|_{s,\alpha}, C\left\|u\right\|_{s+q-1,\alpha})).\end{gathered}$$

правило 3.2:

$$(C_1(\alpha)\wedge C_2(\alpha,M)\wedge C_3(\alpha)\wedge R_{\in}(q_1,R^1)\wedge R_{\in}(q_2,R^1)\wedge R_{\in}(s,R^1)\wedge$$
$$R_{=}(Sym(P_{\alpha,q_1+q_2}),Sym(P_{\alpha,q_1})\cdot Sym(P_{\alpha,q_2})),$$
$$\bigvee_{C>0}\bigwedge_{u\in H_\alpha^{s+q_1+q_2}}R_{\le}(\left\|P_{q_1}\left(t,\alpha D_t\right)P_{q_2}\left(t,\alpha D_t\right)u-P_{q_1+q_2}\left(t,\alpha D_t\right)u\right\|_{s,\alpha},C\left\|u\right\|_{s+q_1+q_2-1,\alpha})).$$

правило 3.3(1):

$$(C_1(\alpha)\wedge C_2(\alpha,M)\wedge C_3(\alpha)\wedge R_{\ge}(s+q,1)\wedge R_{>}(\sigma,1)\wedge R_{\le}(\left[(s+q)/\sigma\right],M+1),$$
$$\bigvee_{C>0}\bigwedge_{\varphi\in S}R_{\le}(\left\|P_q\left(t,\alpha,D_t\right)\varphi-P_{\alpha,q}\left(t,D_{\alpha,t}\right)\varphi\right\|_{s,(\alpha,\sigma)},C\left\|\varphi\right\|_{s+q-1,(\alpha,\sigma)})).$$

правило 3.3(2):

$$(C_1(\alpha)\wedge C_2(\alpha,M)\wedge C_3(\alpha)\wedge R_{\ge}(s+q,1)\wedge R_{\in}(k,\overline{\mathbb{Z}}^+)\wedge R_{\le}(k,M+1),$$
$$\bigvee_{C>0}\bigwedge_{\varphi\in S}R_{\le}(\left\|P_q\left(t,\alpha D_t\right)\varphi-P_{\alpha,q}\left(t,D_{\alpha,t}\right)\varphi\right\|_{s,\alpha,k},C\left\|\varphi\right\|_{s+q-1,\alpha,k})).$$

правило 3.4(1):

$$(C_1(\alpha)\wedge C_2(\alpha,M)\wedge C_3(\alpha)\wedge R_{\ge}(s+q_1+q_2,1)\wedge R_{>}(\sigma,1)\wedge$$
$$R_{\le}(\left[(s+q_1+q_2)/\sigma\right],M+1)\wedge R_{=}(Sym(P_{\alpha,q_1+q_2}),Sym(P_{\alpha,q_1})\cdot Sym(P_{\alpha,q_2})),$$
$$\bigvee_{C>0}\bigwedge_{\varphi\in S}R_{\le}(\left\|P_{q_1}\left(t,\alpha D_t\right)P_{q_2}\left(t,\alpha D_t\right)\varphi-P_{q_1+q_2}\left(t,\alpha D_t\right)\varphi\right\|_{s,(\alpha,\sigma)},C\left\|\varphi\right\|_{s+q_1+q_2-1,(\alpha,\sigma)})).$$

правило 3.4(2):

$$(C_1(\alpha)\wedge C_2(\alpha,M)\wedge C_3(\alpha)\wedge R_{\ge}(s+q_1+q_2,1)\wedge R_{\in}(k,\overline{\mathbb{Z}}^+)\wedge R_{\le}(k,M+1),$$
$$\bigwedge_{C>0}\bigwedge_{\varphi\in S}R_{\le}(\left\|P_{q_1}\left(t,\alpha D_t\right)P_{q_2}\left(t,\alpha D_t\right)\varphi-P_{q_1+q_2}\left(t,\alpha D_t\right)\varphi\right\|_{s,\alpha,k},C\left\|\varphi\right\|_{s+q_1+q_2-1,\alpha,k})).$$

правило 4.1:

$$(C_1(\alpha)\wedge C_2(\alpha,M)\wedge C_3(\alpha)\wedge R_{\ge}(s,0)\wedge R_{\in}(q,\mathbb{Z}^+),$$
$$\bigvee_{C>0}\bigwedge_{u\in H_\alpha^{s+q}}R_{\le}(\left\|M_q\left(\alpha D_t\right)u-\Lambda_\alpha^q\left(D_{\alpha,t}\right)u\right\|_{s,\alpha},C\left\|u\right\|_{s+q-1,\alpha})).$$

правило 4.2:

$$(C_1(\alpha)\wedge C_2(\alpha,M)\wedge C_3(\alpha)\wedge R_{\ge}(s,0)\wedge R_{\in}(q,\mathbb{Z}^+)\wedge R_{>}(\sigma,1)\wedge$$
$$R_{\le}(\left[(s+q)/\sigma\right],M+1),$$
$$\bigvee_{C>0}\bigwedge_{u\in H_{\alpha,\sigma}^{s+q}}R_{\le}(\left\|M_q\left(\alpha D_t\right)u-\Lambda_\alpha^q\left(D_{\alpha,t}\right)u\right\|_{s,(\alpha,\sigma)},C\left\|u\right\|_{s+q-1,(\alpha,\sigma)})).$$

правило 4.3:

$$(C_1(\alpha)\wedge C_2(\alpha,M)\wedge C_3(\alpha)\wedge R_{\in}(q,\mathbb{Z}^+)\wedge R_{\in}(k,\overline{\mathbb{Z}}^+)\wedge R_{\le}(k,M+1),$$
$$\bigvee_{C>0}\bigwedge_{u\in H_\alpha^{s+q,k}}R_{\le}(\left\|M_q\left(\alpha D_t\right)u-\Lambda_\alpha^q\left(D_{\alpha,t}\right)u\right\|_{s,k,\alpha},C\left\|u\right\|_{s+q-1,k,\alpha})).$$

Таким образом, сформирована LP-структура, в которой продукционно-логическое отношение $\xrightarrow{R}$ основано на отношении R с перечисленными выше парами. Эта структура является алгебраической моделью математических знаний, а именно – подмножества теории весовых псевдодифференциальных операторов.

ЗАКЛЮЧЕНИЕ

В настоящей книге изложена основанная на решетках новая алгебраическая теория, предназначенная для моделирования и управления знаниями интеллектуальных систем продукционного типа. Основные итоги представленного исследования состоят в следующих положениях.

1. Введено и обосновано понятие эквивалентности продукционно-логических систем на основе их логического замыкания.

2. Доказаны возможности автоматических локально-эквивалентных преобразований систем продукционного типа.

3. Введено понятие продукционно-логического уравнения и обоснован способ его решения, что на практике соответствует полному обратному выводу.

4. Доказано существование и получен эффективный способ построения логической редукции LP-структур, что в приложениях означает минимизацию соответствующих баз знаний.

5. Предложен и использован новый тезис о продукционной семантике иерархии типов с отношением агрегирования. В результате обоснованы автоматизированные решения важных задач верификации типов и рефакторинга.

6. Показана возможность применения продукционно-логических структур к новым исследованиям свойств императивных алгоритмов.

7. Сформулирована новая концепция трехмерной структуры расширяемой программной системы, как обобщение актуальной ранее двумерной модели.

8. Разработаны и реализованы оригинальные эффективные методы компьютерного представления основанных на решетках алгебраических структур.

9. Предложены и реализованы новые методы обратного вывода и верификации для систем продукционного типа, базирующиеся на решении логических уравнений.

10. Разработана и реализована интегрированная среда создания и выполнения продукционно-логических систем LPExpert, обладающая новыми возможностями исследования, оптимизации и верификации знаний.

11. Для иллюстрации применения LP-структур первого порядка изложены элементы теории неклассических псевдодифференциальных операторов, содержащие, в свою очередь, новые результаты в соответствующей области.

К числу практических результатов монографии относятся следующие.

12. Компьютерная реализация стандартной теории LP-структур и проведенные эксперименты показали ее работоспособность и эффективность.

13. Практическое использование теории LP-структур и ее компьютерной реализации подтверждается выполненными автором на этой основе исследованиями ряда баз знаний, в том числе промышленного уровня.

Теория LP-структур имеет перспективы дальнейшего развития, а ее применения могут охватить более широкие области информатики. В этом плане можно назвать следующие направления.

Усложнение представленных моделей. Это, в частности, относится к LP-структурам на решетках типов. В настоящей книге принята стратегия отказа от «подъема» общего атрибута по иерархии типов при наличии конфликта, однако возможен более тонкий учет особенностей конкретных иерархий типов. Также при использовании в данных структурах свойства $\wedge$-дистрибутивности окажется доступным моделирование других методов рефакторинга. Усложняя эквациональные LP-системы, можно ввести дополнительные операции, которые позволят учитывать структуру термов в равенствах. Созданный в работе класс LPStructure может быть реорганизован в абстрактный класс и применен в качестве основы иерархии типов, охватывающих теорию LP-структур целиком.

Построение новых моделей. Например, функциональные зависимости в реляционных базах данных могут быть описаны специальной LP-структурой с продукционно-логическими свойствами бинарных отношений, основанными на правилах Армстронга [100]. Представляются возможными построения аналогичных моделей для мультиагентных систем, некоторых классов полуструктурированных данных и так далее.

Использование эффективных алгоритмов. При компьютерной реализации LP-структур, в частности, могут быть применены более быстрые алгоритмы построения транзитивного замыкания и редукции отношений, а также специальные алгоритмы их обновления.

Динамические LP-структуры. Данная идея связана с возможностями преобразования самой решетки, в то время как в моделях, представленных в настоящей книге, может изменяться лишь вторичное отношение. Таким образом, например, можно более полно реализовать известные методы рефакторинга в иерархиях типов [56].

Нечеткие LP-структуры. Это направление способно существенно расширить возможности применения изложенных в монографии методов.

ПРИЛОЖЕНИЯ

Приложение А. Фрагменты демонстрационных баз знаний

А.1. «Здоровье»

```
зн(вес) = 55_или_меньше, 55-85, 85_или_больше
зн(холестерин) = норма, излишек
зн(соль) = норма, излишек
зн(раса) = негроидная, европеоидная, монголоидная
зн(пол) = м, ж
см(пол) = Какого Вы пола
зн(характер) = агрессивный, мягкий

объект(курение)   // Объектно-ориентированный стиль описания
     зн = да, нет
     см = Вредите ли Вы своему здоровью курением
     см(да) = Действительно вредите
     см(нет)   // Многострочное описание смысла значения
        Не вредите и
       хорошо делаете
     #
     вопрос = Испытуемый курит?
кобъект

эксп(продолжительность)

вопрос(вес) = Каков вес испытуемого ?
вопрос(холестерин) = Потребление животных жиров:
вопрос(соль) = Потребление соли:
вопрос(раса) = Какой расы испытуемый?
вопрос(пол) = Какого пола испытуемый?
```

```
вопрос(характер) = Характер испытуемого:

прав(01): если  // Это комментарий, не влияющий на работу ЭС
        относительный_вес = нормальный
то
        зн = да.

прав(02): если
        относительный_вес = недостаточный
то
        зн = да:
```

Это объяснение правила 02, его может показать машина вывода

....

конец объяснения правила 02

```
#

прав(03): если
        коронарный_риск = ниже_среднего
то
        сердзаб = низкий.

прав(04): если
        коронарный_риск = средний
то
        сердзаб = низкий.

прав(05): если
        старт = да
то
        сердзаб = иной.

прав(06): если
        возраст = 25_или_меньше и
        пол = ж
то
        основная_продолжительность = 72.

прав(07): если
        возраст = 25_или_меньше и
        пол = м
то
        основная_продолжительность = 67.
```

прав(08): если
    возраст = 25-55 и
    пол = ж
то
    основная_продолжительность = 67.

прав(09): если
    возраст = 25-55 и
    пол=м
то
    основная_продолжительность = 62.

прав(10): если
    возраст = 55_или_больше и
    пол = ж
то
    основная_продолжительность = 64.

прав(11): если
    возраст = 55_или_больше и
    пол = м
то
    основная_продолжительность = 60.

прав(12): если
    вес = 55_или_меньше и
    сложение = мелкое и
    пол = ж
то
    относительный_вес = нормальный.

прав(13): если
    вес = 85_или_больше и
    сложение = мелкое и
    пол = ж
то
    относительный_вес = излишний.

прав(14): если
    вес = 55-85 и
    сложение = мелкое и
    пол = ж

то
    относительный_вес = излишний.

прав(15): если
    вес = 55_или_меньше и
    сложение = крупное
то
    относительный_вес = недостаточный.

прав(16): если
    вес = 55_или_меньше и
    сложение = крупное и
    пол = м
то
    относительный_вес = недостаточный.

прав(17): если
    вес = 55-85 и
    сложение = мелкое и
    пол = ж
то
    относительный_вес = излишний.

прав(18): если
    вес = 55-85 и
    сложение = крупное и
    пол = ж
то
    относительный_вес = нормальный.

прав(19): если
    вес = 55-85 и
    сложение = мелкое и
    пол = м
то
    относительный_вес = нормальный.

прав(20): если
    вес = 55-85 и
    сложение = крупное и
    пол = м

то

относительный_вес = недостаточный.

прав(21): если

вес = 85_или_больше и

пол = ж

то

относительный_вес = излишний.

прав(22): если

вес = 85_или_больше и

пол = м и

сложение = мелкое

то

относительный_вес = излишний.

прав(23): если

вес = 85_или_больше и

пол = м и

сложение = крупное

то

относительный_вес = нормальный.

прав(24): если

холестерин = низкий и

жиры = излишек

то

коронарный_риск = ниже_среднего.

прав(25): если

холестерин = низкий и

жиры = норма

то

коронарный_риск = средний.

прав(26): если

холестерин = излишек

то

коронарный_риск = выше_среднего.

прав(27): если
    холестерин = норма
то
    коронарный_риск = средний.

прав(28): если
    соль = излишек
то
    кровяное_давление = выше_среднего.

прав(29): если
    соль = норма
то
    кровяное_давление = среднее.

прав(63): если
    основная_продолжительность = 72 и
    фактор = ноль
то
    продолжительность = 72.

прав(64): если
    основная_продолжительность = 67 и
    фактор = ноль
то
    продолжительность = 67.

прав(65): если
    основная_продолжительность = 64 и
    фактор = ноль
то
    продолжительность = 64.

прав(66): если
    основная_продолжительность = 62 и
    фактор = ноль
то
    продолжительность = 62.

прав(67): если
    основная_продолжительность = 60 и
    фактор = ноль

то
    продолжительность = 60.

прав(68): если
    основная_продолжительность = 72 и
    фактор = плюс_12
то
    продолжительность = 84.

прав(69): если
    основная_продолжительность = 67 и
    фактор = плюс_12
то
    продолжительность = 79.

прав(70): если
    основная_продолжительность = 64 и
    фактор = плюс_12
то
    продолжительность = 76.

прав(71): если
    основная_продолжительность = 62 и
    фактор = плюс_12
то
    продолжительность = 74.

прав(72): если
    основная_продолжительность = 60 и
    фактор = плюс_12
то
    продолжительность = 72.

прав(73): если
    основная_продолжительность = 72 и
    фактор = минус_12
то
    продолжительность = 60.

прав(74): если
    основная_продолжительность = 67 и
    фактор = минус_12

то
продолжительность = 55.

прав(75): если
основная_продолжительность = 64 и
фактор = минус_12
то
продолжительность = 52.

прав(76): если
основная_продолжительность = 62 и
фактор = минус_12
то
продолжительность = 50.

прав(77): если
основная_продолжительность = 60 и
фактор = минус_12
то
продолжительность = 48.

А.2. «Электрики»

эксп(знания)
многозначный(знания)
см(знания)=ОБЩИЙ УРОВЕНЬ ЗНАНИЙ
зн(зак_1)=фарадея,ома,кирхгофа
зн(ед_тока)=верно,отчасти_верно,не_верно
зн(вел_тока)=I=(q*S)/(t*L),I=q/t,I=(S/L)*(q/t)
см(ед_тока)=ОПРЕДЕЛЕНИЕ ВЕЛИЧИНЫ ТОКА
см(зак_1)=ЗАКОН ОМА
см(вел_тока)=ФОРМУЛА СИЛЫ ТОКА
вопрос(вел_тока)
Выберите обобщенную формулу определения силы тока
I – ток, q – заряд, t – время, S – сечение проводника, L – длина проводника, / – деление, * – умножение.
#
вопрос(ед_тока)
Единица величины тока есть такой ток, при котором за единицу времени через 1 кв. мм. проходит заряд в 1 кулон.
#

вопрос(зак_1)
ОПРЕДЕЛЕНИЕ КАКОГО ЗАКОНА ЗДЕСЬ ПРИВЕДЕНО
Величина электрического тока в каком либо участке проводника прямо пропорциональна напряжению между концами выбранного участка и обратно пропорциональна его сопротивлению.
#
вопрос(ток)
Электрическим током называется перемещение зарядов в проводнике.
#
зн(ток)=верно,отчасти_верно,не_верно
вопрос(реле)
Какое из перечисленных реле относится к магнитоуправляемым?
#
зн(реле)=ркм1,рэс6,рэс64а
вопрос(емкость)
Электрической емкостью называется отношение количества электричества проводника к его потенциалу.
#
зн(емкость)=верно,отчасти_верно,не_верно

прав(1):если
ед_тока=не_верно и
вел_тока=I=q/t и
зак_1=ома
то
пр_п_ток=пять и
пр_пост_ток=пять.

прав(2):если
ед_тока=не_верно и
вел_тока=I=q/t и
зак_1=фарадея
то
пр_п_ток=три,кд=50:
В данном случае речь идет о законе Ома:
I = U/R
#

прав(3):если
ед_тока=не_верно и

вел_тока=I=q/t и
зак_1=кирхгофа
то
пр_п_ток=три,кд=50:
В данном случае речь идет о законе Ома:
$I = U/R$
#
прав(4):если
ед_тока=не_верно и
вел_тока=I=(q*S)/(t*L) и
зак_1=ома
то
пр_п_ток=четыре,кд=50 и:
Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени.
Учебник физики для 9 класса средней школы. – М.: Просвещение, – 1986 г. С. 162.
#
прав(5):если
ед_тока=не_верно и
вел_тока=I=(q*S)/(t*L) и
зак_1=фарадея
то
пр_п_ток=два:
Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: $I = q/t$.
Речь идет о законе Ома: $I = U/R$.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#
прав(6):если
ед_тока=не_верно и
вел_тока=I=(q*S)/(t*L) и
зак_1=кирхгофа
то
пр_п_ток=два:
Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: $I = q/t$.

Речь идет о законе Ома: I = U/R.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#
прав(7):если
ед_тока=не_верно и
вел_тока=I=(S/L)*(q/t) и
зак_1=ома
то
пр_п_ток=три,кд=50:
Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: I = q/t.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#
прав(8):если
ед_тока=не_верно и
вел_тока=I=(S/L)*(q/t) и
зак_1=кирхгофа
то
пр_п_ток=два:
Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: I = q/t.
Речь идет о законе Ома: I = U/R.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#
прав(9):если
ед_тока=не_верно и
вел_тока=I=(S/L)*(q/t) и
зак_1=фарадея
то
пр_п_ток=два:
Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: I = q/t.
Речь идет о законе Ома: I = U/R.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#

прав(10):если
ед_тока=верно и
вел_тока=I=q/t и
зак_1=ома
то
пр_п_ток=четыре:
Правильное определение:
Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#
прав(11):если
ед_тока=верно и
вел_тока=I=q/t и
зак_1=фарадея
то
пр_п_ток=два:
Правильное определение:
Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет.
Речь идет о законе Ома: I = U/R.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#
прав(12):если
ед_тока=верно и
вел_тока=I=q/t и
зак_1=кирхгофа
то
пр_п_ток=два:
Правильное определение:
Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет.
Речь идет о законе Ома: I = U/R.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#

прав(13):если
ед_тока=верно и
вел_тока=I=(q*S)/(t*L) и
зак_1=ома
то
пр_п_ток=три,кд=30:

Правильное определение:

Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет.

Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: $I = q/t$.

Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.

#

прав(14):если
ед_тока=верно и
вел_тока=I=(q*S)/(t*L) и
зак_1=кирхгофа
то
пр_п_ток=два:

Правильное определение:

Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет.

Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: $I = q/t$.

Речь идет о законе Ома: $I = U/R$.

Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.

#

прав(15):если
ед_тока=верно и
вел_тока=I=(q*S)/(t*L) и
зак_1=фарадея
то
пр_п_ток=два:

Правильное определение:

Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника,
в этом случае, значения не имеет.

Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: I = q/t.

Речь идет о законе Ома: I = U/R.

Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.

#

прав(16):если
ед_тока=верно и
вел_тока=I=(S/L)*(q/t) и
зак_1=ома
то
пр_п_ток=три,кд=30:

Правильное определение:

Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет.

Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: I = q/t.

Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.

#

прав(17):если
ед_тока=верно и
вел_тока=I=(S/L)*(q/t) и
зак_1=кирхгофа
то
пр_п_ток=два:

Правильное определение:

Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет.

Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: I = q/t.

Речь идет о законе Ома: I = U/R.

Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.

#

прав(18):если
ед_тока=верно и
вел_тока=I=(S/L)*(q/t) и
зак_1=фарадея
то
пр_п_ток=два:

Правильное определение:

Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет.

Силой тока называется отношение заряда q, переносимого через поперечное сечение проводника за интервал времени t, к этому интервалу времени: I = q/t.

Речь идет о законе Ома: I = U/R.

Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.

#

прав(10):если
ед_тока=отчасти_верно и
вел_тока=I=q/t и
зак_1=ома
то
пр_п_ток=четыре,кд=40:

Правильное определение:

Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет. То, что Вы ответили "отчасти_верно", говорит о не знании определения.

Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.

#

прав(11):если
ед_тока=отчасти_верно и
вел_тока=I=q/t и
зак_1=фарадея
то
пр_п_ток=три,кд=30:

Правильное определение:
Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет. То, что Вы ответили “отчасти_верно”, говорит о не знании определения.
Речь идет о законе Ома: I = U/R.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#
прав(12):если
ед_тока=отчасти_верно и
вел_тока=I=q/t и
зак_1=кирхгофа
то
пр_п_ток=три,кд=30:
Правильное определение:
Заряд, перенесенный в единицу времени, служит основной количественной характеристикой тока. Сечение проводника, в этом случае, значения не имеет. То, что Вы ответили “отчасти_верно”, говорит о не знании определения.
Речь идет о законе Ома: I = U/R.
Учебник физики для 9 класса средней школы. – М.: Просвещение, 1986. – 271 с.: ил.
#

А.3. «Закон распределения»

объект(способ_анализа)
зн=графический,параметрический
см=способ анализа имеющихся данных
см(графический)=на основе графиков для неизвестного закона распределения
см(параметрический)=на основе критериев согласия
вопрос=Выберите способ анализа закона распределения:
кобъект
эксп(закон_распределения)=ОТВЕТ НЕИЗВЕСТЕН – НАЖМИТЕ ENTER
вопрос(старт)=Вы готовы начать ?

прав(01): если
способ_анализа=графический и
вид_графика=график_функции_распределения и

ступенчатая=нет

то

ок=да:

Вы выбрали графический способ анализа данных (поэтому результат экспертизы будет без подтверждения)и указали вид графика – график функции распределения.

Если Ваша функция не является ступенчатой, то будет сделан вывод о непрерывном типе случайной величины. Возможные законы распределения для неё: нормальный, показательный, равномерный

#

объект(вид_графика)

зн=график_функции_распределения,график_плотности_распределения,график_гистограммы

см=график какой функции Вам дан

см(график_гистограммы)=график гистограммы обычно напоминает функцию плотности распределения

вопрос=Выберите вид графика:

кобъект

вопрос(ступенчатая)=Является ли функция ступенчатой ?

см(ступенчатая)=ступенчатой является функция для дискретных случайных величин

зн(ступенчатая)=да,нет

прав(02): если

ок=да и

вид_функции=гладкая_

то

закон_распределения=нормальный_без_подтверждения:

Выбран вид функции распределения – гладкая. Он соответствует нормальному закону.

#

объект(вид_функции)

зн=гладкая_,кусочно_непрерывная,монотонно_возрастающая_на_промежутке_0-бесконечность

см=выберите то, что наиболее соответствует заданной функции

см(гладкая_)=дифференцируема(имеет производную в каждой точке) см(монотонно_возрастающая_на_промежутке_0-бесконечность)=может быть определена и на всей числовой оси

вопрос=Какая характеристика соответствует Вашей функции?

кобъект

прав(03): если

ок=да и

вид_функции=кусочно_непрерывная

то

закон_распределения=равномерный_без_подтверждения:

Выбран вид функции распределения - кусочно непрерывная. Он соответствует равномерному закону.

#

прав(04): если

ок=да и

вид_функции=монотонно_возрастающая_на_промежутке_0-бесконечность

то

закон_распределения=показательный_без_подтверждения:

Выбран вид функции распределения - монотонно возрастающая на промежутке 0-бесконечность. Он соответствует показательному закону (см. стр. 149 Гмурман В. Е. Теория вероятностей и математическая статистика: Учеб. пособие для вузов)

#

прав(05): если

способ_анализа=графический и

вид_графика=график_функции_распределения и

ступенчатая=да

то

ок1=да:

Вы выбрали графический способ анализа данных и указали вид графика – график функции распределения. Если Ваша функция является ступенчатой, то будет сделан вывод о дискретном типе случайной величины. Возможные законы распределения для неё: биномиальный, пуассоновский. Далее будет анализироваться объем выборки и вероятность.

#

```
прав(06): если
       ок1=да и
       размер_выборки=100_или_больше и
       вероятность=1/10_или_меньше
то
       закон_распределения=Пуассона_без_подтверждения:
При размере выборки не меньше 100 и вероятности не больше 1/10
делаем вывод о том, что закон распределения – Пуассона.
#

вопрос(размер_выборки)=Чему равен объем выборки(количество
её элементов)?
зн(размер_выборки)=100_или_больше,меньше_100,неизвестен
вопрос(вероятность)=Каково значение вероятности?
зн(вероятность)=1/10_или_меньше,больше_1/10,неизвестна

прав(07): если
       ок1=да и
       размер_выборки=100_или_больше и
       вероятность=больше_1/10
то
       закон_распределения=биномиальный_без_подтверждения:
При размере выборки не меньше 100 и вероятности больше 1/10
делаем вывод о том, что закон распределения – биномиальный.
#

прав(08): если
       ок1=да и
       размер_выборки=меньше_100
то
      закон_распределения=?_невозможно_выдвинуть_однознач-
      ную_гипотезу:
При малом объеме выборки выдвинуть однозначную гипотезу не-
возможно.
По предварительным данным закон распределения: Пуассона или
биномиальный.
#

прав(09): если
       ок1=да и
       размер_выборки=неизвестен
```

то
закон_распределения=?_невозможно_выдвинуть_однозначную_гипотезу:
При неизвестном объеме выборки выдвинуть однозначную гипотезу невозможно.
По предварительным данным закон распределения: Пуассона или биномиальный.
#

прав(10): если
ок1=да и
вероятность=неизвестна
то
закон_распределения=?_невозможно_выдвинуть_однозначную_гипотезу:
При неизвестной вероятности выдвинуть однозначную гипотезу невозможно.
По предварительным данным закон распределения: Пуассона или биномиальный.
#

прав(11): если
способ_анализа=графический и
вид_графика=график_плотности_распределения
то
ок2=да:
Вы выбрали графический способ анализа данных и указали вид графика — график плотности распределения. Эта характеристика (плотность) соответствует непрерывно распределенным случайным величинам. Возможные законы распределения для них: нормальный, показательный, равномерный.
#

прав(12): если
ок2=да и
вид_функции_плотности=гладкая
то
закон_распределения=нормальный_без_подтверждения:
Выбран вид плотности распределения – гладкая. Он соответствует нормальному закону (см. стр. 130 Гмурман В. Е. Теория вероятностей и математическая статистика: Учеб. пособие для вузов)
#

объект(вид_функции_плотности)
    зн=гладкая,монотонно_убывающая_на_промежутке_0-бесконечность,постоянная_на_некотором_отрезке
    см=выберите то, что наиболее соответствует заданной функции плотности
    см(гладкая)=дифференцируема(имеет производную в каждой точке)
    см(монотонно_убывающая_на_промежутке_0-бесконечность)=может быть определена и на всей числовой оси
    вопрос=Какая характеристика соответствует Вашей функции плотности?
кобъект

прав(13): если
    ок2=да и
    вид_функции_плотности=монотонно_убывающая_на_промежутке_0-бесконечность
то
    закон_распределения=показательный_без_подтверждения:
Выбран вид плотности распределения – монотонно убывающая на промежутке 0-бесконечность. Он соответствует показательному закону (см. стр. 149 Гмурман В. Е. Теория вероятностей и математическая статистика: Учеб. пособие для вузов)
#

прав(14): если
    ок2=да и
    вид_функции_плотности=постоянная_на_некотором_отрезке
то
    закон_распределения=равномерный_без_подтверждения:
Выбран вид плотности распределения – постоянная на некотором отрезке. Он соответствует равномерному закону (см. стр. 123 Гмурман В. Е. Теория вероятностей и математическая статистика: Учеб. пособие для вузов)
#

прав(15): если
    способ_анализа=графический и
    вид_графика=график_гистограммы
то
    ок3=да:

Вы выбрали графический способ анализа данных и указали вид графика – график гистограммы. Обычно (если не указано иное) гистограмма соответствует непрерывно распределенным случайным величинам. Возможные законы распределения для них: нормальный, показательный, равномерный.
#

прав(16): если
    ок3=да и
    вид_гистограммы=равнобедренный_треугольник
то
    закон_распределения=нормальный_без_подтверждения:
Выбран вид гистограммы – равнобедренный треугольник.
Он соответствует нормальному закону.
#

объект(вид_гистограммы)
  зн=равнобедренный_треугольник,прямоугольник,прямоугольный_треугольник_со_сторонами_на_положительных_полуосях
  см(равнобедренный_треугольник)
    он должен быть виден, если мысленно соединить
    непрерывной ломаной середины прямоугольников гистограммы
  #
   см(прямоугольный_треугольник_со_сторонами_на_положительных_полуосях)
   он должен быть виден, если мысленно соединить
   непрерывной ломаной середины прямоугольников гистограммы
  #
  см(прямоугольник)
    он должен быть виден, если мысленно можно провести
    линию такую, что все прямоугольники, составляющие
    гистограмму будут либо чуть ниже, либо чуть выше неё
  #
  вопрос=Какую фигуру напоминает гистограмма?
кобъект

прав(17): если
    ок3=да и
вид_гистограммы=прямоугольный_треугольник_со_сторонами_на_положительных_полуосях

то

закон_распределения=показательный_без_подтверждения:
Выбран вид гистограммы – прямоугольный треугольник со сторонами на положительных полуосях. Он соответствует показательному закону.
#

прав(18): если
ок3=да и
вид_гистограммы=прямоугольник
то
закон_распределения=равномерный_без_подтверждения:
Выбран вид гистограммы – прямоугольник. Он соответствует равномерному закону.
#

прав(19): если
способ_анализа=параметрический и
генеральная_совокупность=непрерывный_тип и
критерий_согласия=Колмогорова
то
ок4=да:
Вы выбрали параметрический способ анализа данных (поэтому результат экспертизы будет с подтверждением), непрерывный тип генеральной совокупности (это значит, что будет проверяться гипотеза об одном из законов: нормальный, показательный, равномерный) и критерий согласия Колмогорова проверки гипотезы о законе распределения.
#

объект(генеральная_совокупность)
зн=непрерывный_тип,дискретный_тип
см
если тип генеральной совокупности не дан, можете выбрать его самостоятельно
исходя из того, гипотезу о каком из законов распределения будете проверять
#
см(непрерывный_тип)
законы распределения величин непрерывного типа:
нормальный, показательный, равномерный (они будут предложены)

```
    #
    см(дискретный_тип)
      законы распределения величин дискретного типа:
      биномиальный, Пуассона(они будут предложены)
    #
    вопрос(генеральная_совокупность)
        Для генеральной совокупности какого типа
        будет проверяться гипотеза о законе распределения?
    #
кобъект
```

вопрос(критерий_согласия)=Выберите критерий согласия:
см(критерий_согласия)=Вам предложено два критерия проверки гипотезы о законе распределения
зн(критерий_согласия)=Колмогорова,хи-квадрат(Пирсона)

```
прав(20): если
      ок4=да и
      предполагаемый_закон=нормальный и
      уровень_значимости=1/10 и
      выполнение_условия=да
то
```

закон_распределения=нормальный_подтверждение_да_с_уровнем_значимости_1/10:
Выполнение условия соответствует принятию с выбранным уровнем значимости 1/10 гипотезы о том, что заданная выборка принадлежит генеральной совокупности, распределенной по нормальному закону.
#

```
вопрос(предполагаемый_закон)
      Гипотезу о каком из законов распределения непрерывных
      величин хотели бы проверить?
#
```

зн(предполагаемый_закон)=нормальный,равномерный,показательный вопрос(уровень_значимости)=Выберите уровень значимости:
см(уровень_значимости)
  Уровень значимости – это вероятность совершить ошибку
  первого рода, то есть отвергнуть гипотезу о том, что заданная
  выборка из генеральной совокупности, распределенной по выбранному закону
#

```
зн(уровень_значимости)=1/10,5/100,1/100

объект(выполнение_условия)
    зн=да,нет
    см
     Статистика Колмогорова Dn представляет собой максималь-
     ное отклонение эмпирической функции распределения Fn(x)
     от гипотетической (соответствующей теоретической) F(x) и
     находится так: Dn=max|Fn(x)-F(x)|.
     max берется по всем x. Fn(x) находят по выборке, F(x) берут
     для предполагаемого закона распределения. Квантиль
     распределения Колмогорова находят по соответствующим
     таблицам для выбранного уровня значимости
    #
    вопрос
      Выполнено ли условие: значение статистики Колмогорова
      меньше, чем квантиль распределения, деленный на корень
      из объема выборки?
    #
кобъект

прав(67): если
      ок5=да и
      предполагаемый_закон_критерия=биномиальный и
      уровень_значимости=1/100 и
      выполнение_условия_критерия=нет_
то
закон_распределения=биномиальный_подтверждение_нет_с_
уровнем_значимости_1/100:
Нарушение условия критерия соответствует отвержению с выбран-
ным уровнем значимости 1/100 гипотезы  о том, что заданная вы-
борка принадлежит генеральной совокупности, распределенной по
биномиальному закону.
#

прав(68): если
      ок5=да и
      предполагаемый_закон_критерия=Пуассона и
      уровень_значимости=1/100 и
      выполнение_условия_критерия=да_
то
```

закон_распределения=Пуассона_подтверждение_да_с_уровнем_значимости_1/100:
Выполнение условия критерия соответствует принятию с выбранным уровнем значимости 1/100 гипотезы о том, что заданная выборка принадлежит генеральной совокупности, распределенной по закону Пуассона.
#

прав(69): если
    ок5=да и
    предполагаемый_закон_критерия=Пуассона и
    уровень_значимости=1/100 и
    выполнение_условия_критерия=нет_
то
закон_распределения=Пуассона_подтверждение_нет_с_уровнем_значимости_1/100:
Нарушение условия критерия соответствует отвержению с выбранным уровнем значимости 1/100 гипотезы о том, что заданная выборка принадлежит генеральной совокупности, распределенной по закону Пуассона.
#
прав(70): если
    способ_анализа=параметрический и
    генеральная_совокупность=дискретный_тип и
    критерий_согласия=Колмогорова
то
    закон_распределения=?_выбранный_критерий_применять_нельзя:
Критерий согласия Колмогорова можно применять только для непрерывных случайных величин
    #

Приложение B. Теория весовых псевдодифференциальных операторов

Данное приложение служит основой иллюстрации применения LP-структур для алгебраической формализации математических знаний. Изложенная здесь теория относится к области дифференциальных уравнений с частными производными высокого порядка. Она изучает псевдодифференциальные операторы (ПДО) специальных классов, не относящиеся к главному типу [108]. Такие операторы могут возникать при исследовании разрешимости общих граничных задач для эллиптических уравнений в компактных областях с гладкой границей (см., например, [97]). Одна из таковых – задача с косой (наклонной) производной. Она имеет богатую историю (ее изучал еще А. Пуанкаре [68], занимаясь теорией морских приливов) и до настоящего времени продолжает вызывать интерес исследователей. Об этом свидетельствуют связанные с ней непрекращающиеся публикации как математиков (обширная библиография приведена в [132]), так и специалистов-прикладников (см., например, [125, 184]). Этой задаче была посвящена также и работа [102], результаты которой могут быть обобщены путем применения рассматриваемых здесь классов вырождающихся эллиптических ПДО.

Предлагается метод исследования этих операторов, основанный на их сравнении с весовыми псевдодифференциальными операторами (ВПДО), построенными по введенному В. П. Глушко [101] весовому преобразованию Фурье F_α. Преобразование F_α «поглощает» множитель α, в результате чего получаются операторы с символами из хорошо известного класса. Поскольку преобразование F_α определено на функциях в полупространстве, то одной из проблем является распространение понятия ВПДО на функции во всем пространстве.

В полном объеме теория весовых ПДО во всем пространстве с описанием приложений изложена в работах [102, 137–145]. Часть приведенных здесь результатов представляет собой обобщение указанных работ с точки зрения классов символов вырождающихся ПДО и соответствует статье [157].

B.1. Весовые пространства обобщенных функций

В данном разделе вводятся используемые в дальнейшем основные понятия и обозначения. Необходимые сведения из классической теории обобщенных функций и псевдодифференциальных операторов содержатся, например, в [105, 108].

Пусть $n \geq 1$ – целое, R_1, R^1, R – одномерные евклидовы пространства вещественных чисел, $C^M(R)$ – множество M раз непрерывно дифференцируемых функций, $C^1(R) = C(R)$. Введем еще несколько обозначений:

$\mathbb{Z}$ – множество целых чисел;

$\mathbb{Z}^+$ – множество целых положительных чисел;

$\bar{\mathbb{Z}}^+$ – множество целых неотрицательных чисел;

$[\cdot]$ – целая часть вещественного числа;

$R_n = \{(t,x) \mid t \in R_1, x \in R_{n-1}\}$; $R^n = \{(\eta,\xi) \mid \eta \in R^1, \xi \in R^{n-1}\}$;

R^n – евклидово пространство, двойственное к R_n ;

$F_{x\to\xi}$ – преобразование Фурье, переводящее x в ξ ;

$F^{-1}_{\xi\to x}$ – обратное преобразование Фурье, переводящее ξ в x ;

$R_n^+ = \{(t,x) \mid t > 0, x \in R_{n-1}\}$; $R_n^- = \{(t,x) \mid t < 0, x \in R_{n-1}\}$;

для $\varphi \in C(R_1)$ обозначим $\varphi^\pm = \varphi|_{R_1^\pm}$;

$$\varphi_0^\pm(t) = \begin{cases} \varphi(t), & t \in R_1^\pm \\ 0 & t \in R_1^\mp \end{cases};$$

S – пространство Шварца гладких быстроубывающих функций;

$\|\cdot\|$ и $\|\cdot\|^\pm$ – нормы в пространствах Лебега соответственно $L_2(R_n)$ и $L_2(R_n^\pm)$;

H^s – пространства Соболева–Слободецкого–Ароншайна с нормой

$$\|u\|_s = F^{-1}_{\xi\to x} F^{-1}_{\eta\to t}[u] \left\| \left(1 + \eta^2 + |\xi|^2\right)^{s/2} F_{x\to\xi} F_{t\to\eta}[u] \right\|.$$

Введем весовую функцию $\alpha \in C(R)$, удовлетворяющую следующим условиям.

Условие 1. $\alpha(t) = 0, t \leq 0;$ $\alpha(t) > 0, t > 0;$ $\alpha(t) = const, t \geq d > 0.$

Условие 2. $\alpha|_{R^+} \in C^\infty(\bar{R}^+); \alpha \in C^{M+1}(R) \quad (M \geq 0).$

Далее положим

$$\tau = \tau(t) = \int_d^t \frac{dp}{\alpha(p)}, \ \ t \in R^+. \qquad (1)$$

В силу условий 1–2 существует функция $t = \psi(\tau), \tau \in R$, обратная к (1). По функции $\varphi \in C(R^+)$ определим функцию $\varphi_\alpha \in C(R)$ равенством

$$\varphi_\alpha(\tau)=\alpha^{1/2}\left(\psi(\tau)\right)\varphi\left(\psi(\tau)\right). \tag{2}$$

Через $S=S(R)$ обозначим пространство Шварца функций φ со счетной системой норм

$$\left\|\varphi\right\|_p=\sup_{t\in R}\left(1+|t|\right)^p\sum_{j=0}^{p}\left|D_t^j\varphi(t)\right|,\ \ p=0,1...$$

Определение 1. Функция $\varphi\in C^\infty\left(R^+\right)$ принадлежит пространству $S_\alpha^+=S_\alpha\left(R^+\right)$, если соответствующая функция φ_α принадлежит S. Система норм в S_α^+ имеет вид

$$\left\|\varphi\right\|_{p,\alpha}^+=\left\|\varphi_\alpha\right\|_p,\ \ p=0,1...$$

В S_α^+ непрерывны следующие операции:

1) умножение на функцию $a\in C\left(R^+\right)$, для которой $a_\alpha\in C^\infty(R)$, причем каждая производная функции a_α растет на бесконечности не быстрее полинома;

2) дифференцирование D_t;

3) весовое дифференцирование

$$D_{\alpha,t}=\alpha^{1/2}(t)D_t\alpha^{1/2}(t). \tag{3}$$

На функциях $\varphi\in S_\alpha^+$ определим весовое преобразование Фурье F_α: $F_\alpha[\varphi](\eta)=F_{\tau\to\eta}[\varphi_\alpha]$, где $F_{\tau\to\eta}$ – обычное преобразование Фурье, переводящее τ в η. Для преобразования F_α существует обратное F_α^{-1}: $F_\alpha^1[\nu](t)=\alpha^{-1/2}(t)F_{\eta-\tau}^{-1}[\nu]$. Оператор F_α отображает взаимно однозначно и непрерывно S_α^+ на S. Справедливы равенства

$$F_\alpha\left[D_{\alpha,t}^j\varphi\right]=\eta^jF_\alpha[\varphi];\ D_\eta^jF_\alpha[\varphi]=F_\alpha\left[\tau^j\varphi\right],\quad j=0,1,\dots, \tag{4}$$

где $\varphi\in S_\alpha^+$; $D_{\alpha,t}$ определена в (3).

Определение 2. Функция $\varphi\in C^\infty\left(R^-\right)$ принадлежит пространству S^-, если функция φ_0 – продолжение нулем на R функции φ – принадлежит S. Топология в S^- задается системой норм $\left\|\varphi\right\|_p^-=\left\|\varphi_0\right\|_p$, $p=0,1,\dots$

Определение 3. Функция $\varphi\in C^\infty(R)$ принадлежит пространству $S_\alpha=S_\alpha(R)$, если φ^+ принадлежит S_α^+, а φ^- принадлежит S^-. Введем в S_α счетную систему норм $\left\|\varphi\right\|_{p,\alpha}=\left\|\varphi^+\right\|_{p,\alpha}^++\left\|\varphi^-\right\|_p^-$, $p=0,1,\dots$

Нетрудно ввести в S^- и S_α непрерывные операции, аналогичные 1)–3). Справедливо $S_\alpha\subset S$.

Пространство обобщенных функций $S'\left(S_\alpha'^+, S'^-, S_\alpha'\right)$ состоит из всех линейных непрерывных функционалов на S (соответственно $S_\alpha^+, S^-, S_\alpha$).

Операции, определенные в $S_\alpha^+, S^-, S_\alpha$, легко переносятся на сопряженные пространства $S_\alpha'^+, S'^-, S_\alpha'$. Нетрудно установить также изоморфизм пространств $S_\alpha'^+$ и S' по аналогии с изоморфизмом пространств S_α^+ и S, осуществляемым равенством (2).

В.2. Весовые псевдодифференциальные операторы

Определение 1. Функция $\lambda \in C^\infty(R \times R)$ принадлежит классу символов $\Phi_q\,(q \in R)$, если выполняются оценки

$$\left|D_t^\ell D_\eta^j \lambda(t,\eta)\right| \le C_{\ell,j}\left(1+\eta^2\right)^{\frac{q-j}{2}},\ \ \ell, j = 0,1,\ldots,\ \text{для любых}\ t \in R,\ \eta \in R.$$

Определение 2. На функциях $\varphi \in S_\alpha$ определим ВПДО $P_{\alpha,q} = P_{\alpha,q}\left(t, D_{\alpha,t}\right)$ с символом $\lambda \in \Phi_q$ по формуле

$$P_{\alpha,q,\varphi} = \begin{cases} F_\alpha^{-1}\left[\lambda(t,\eta)F_\alpha\left[\varphi^+\right]\right], t > 0, \\ \lambda(t,0)\varphi^-, t < 0. \end{cases} \qquad (1)$$

Легко проверить, что в случае

$$\lambda(t,\eta) = \sum_{j=0}^m a_j(t)\eta^j,$$

где a_j – гладкие ограниченные функции, соответствующий ВПДО имеет вид

$$P_{\alpha,m}\varphi = \sum_{j=0}^m a_j(t) D_{\alpha,t}^j \varphi,\ \ \varphi \in S_\alpha.$$

Нетрудно показать также, что ВПДО отображает S_α в S_α. Ниже установим, что оператор $P_{\alpha,q}$, определенный в (1), не «разрывает» в точке $t = 0$ функции, имеющие в этой точке достаточное количество производных. Для этого введем еще один класс функций.

Определение 3. Множество $S^M = S^M(R)$ ($M \ge 0$ – целое) состоит из функций $\varphi \in C^M(R)$, для которых $\varphi^\pm \in C^\infty\left(\overline{R}^\pm\right)$, причем

$$\sup_{t \in R^\pm}\left(1+|t|\right)^p \sum_{j=0}^p \left|D_j^t \varphi^\pm\right| < \infty,\ p = 0,1,\ldots \qquad (2)$$

Теорема 1. Пусть выполнены условия 1–2 (раздел В.1.). Тогда при любом $q \in R$ ВПДО $P_{\alpha,q}\left(t, D_{\alpha,t}\right)$ отображает S^M в S^M.

Доказательство. Из результатов [137] вытекает представление

$$P_{\alpha,q}\,\varphi\big|_{t=\psi(\tau)} = \alpha^{-1/2}\left(\psi(\tau)\right)F_{\eta\to\tau}^{-1}\left[\lambda\left(\psi(\tau),\eta\right)F_{\tau\to\eta}\left[\varphi_\alpha(\tau)\right]\right], \qquad (3)$$

где ψ – функция, обратная к (В.1.1); φ_α определена в (В.1.2); $\varphi \in S^M$.

Из (3) и свойства псевдолокальности ПДО (см., например, [138]) следует, что $\left(P_{\alpha,q}\varphi\right)^{+} \in C^{\infty}\left(R^{+}\right)$, если $\varphi^{+} \in C^{\infty}\left(R^{+}\right)$. Соотношение $\left(P_{\alpha,q}\varphi\right)^{-} \in C^{\infty}\left(R^{-}\right)$ вытекает непосредственно из (1).

Заключительная часть доказательства существенно не отличается от доказательства соответствующей теоремы в [137]. □

Для выяснения дальнейших свойств ВПДО введем весовые пространства типа Соболева–Слободецкого.

Введем обозначение Λ_{α}^{s} – ВПДО с символом $\lambda(\eta) = (1+\eta^2)^{s/2}$ $(s \in R)$.

Определение 4. Функция $u \in S'_{\alpha}$ принадлежит пространству $H_{\alpha}^{s} = H_{\alpha}^{s}(R)$ $(s \in R)$, если конечна норма

$$\|u\|_{s,\alpha} = \left\|\Lambda_{\alpha}^{s} u\right\|. \tag{4}$$

Легко проверить, что при целом $s \geq 0$ норма (4) эквивалентна норме

$$|||u|||_{s,\alpha} = \sum_{j=0}^{s} \left\|D_{\alpha,t}^{j} u\right\|.$$

Пользуясь результатами [140], можно получить равенство

$$\|u\|_{s,\alpha}^2 = \left\|u_{\alpha}^{+}\right\|_{s}^2 + \left\|u^{-}\right\|^2. \tag{5}$$

Равенство (5) позволяет установить для H_{α}^{s} ряд аналогов свойств пространства H^{s}. В частности, пространство H_{α}^{s} является полным, и в нем плотно множество S_{α}. С помощью (5), (3) и свойств обычных ПДО (см. [105]) можно также доказать следующие две теоремы.

Теорема 2. Пусть выполнены условия 1–2. Тогда при любом $q \in R$ оператор $P_{\alpha,q}$ ограничен из H_{α}^{s+q} в H_{α}^{s}.

Теорема 3. Пусть выполнены условия 1–2. Тогда при любых $q_1, q_2 \in R$ оператор $P_{\alpha,q_1}P_{\alpha,q_2}$ является ВПДО с символом из класса $\mathbf{\Phi}_{q_1+q_2}$, главная часть которого равна $\lambda_1(t,\eta)\lambda_2(t,\eta)$, где λ_1 и λ_2 – соответственно главные символы операторов P_{α,q_1} и P_{α,q_2}.

В следующем пункте переходим к рассмотрению вырождающихся ПДО специальных классов.

В.3. Вырождающиеся эллиптические псевдодифференциальные операторы

На функциях $\varphi \in S_{\alpha}$ определены ПДО вида

$$P_q(t,\alpha D_t)\varphi = F_{\eta\to t}^{-1}\left[\lambda\left(t,\alpha(t)\eta\right)F_{t\to\eta}[\varphi]\right], \text{ где } \lambda \in \mathbf{\Phi}_q. \tag{1}$$

Наряду с условиями 1–2 (раздел В.1) будем использовать также следующее условие.

Условие 3. Существуют $C>0$ и $N\ge 1$ такие, что $\alpha^N(\theta)\le C\alpha(t)$, $0<\theta<t$.

Очевидно, для неубывающей в некоторой окрестности точки $t=0$ функции α условие 3 выполнено при $N=1$. Если существуют неубывающие непрерывные функции α_1 и α_2, для которых $0<\alpha_1(t)\le\alpha(t)\le\alpha_2(t)$ при $t>0$ и $\alpha_2^N(t)\le C\alpha_1(t)$, то условие 3 выполнено при данном N.

Далее сформулируем теорему сравнения операторов (1) с соответствующими ВПДО в норме пространств H_α^s. В [140] аналогичная теорема доказана для операторов с символами вида $\lambda(\alpha(t)\eta)$ (символ $\lambda(\eta)$ не зависит от t).

Теорема 1. Пусть выполнены условия 1–3. Тогда для любой функции $u\in H_\alpha^{s+q}\ (s,q\in R)$ справедлива оценка

$$\left\|P_q(t,\alpha D_t)u-P_{\alpha,q}(t,D_{\alpha,t})u\right\|_{s,\alpha}\le C\|u\|_{s+q-1,\alpha} \tag{2}$$

с постоянной $C>0$, не зависящей от u.

Доказательство этой теоремы существенно не отличается от проведенного в [137], поэтому его здесь не приводим.

Из теорем 1 и В.2.2 вытекает

Следствие 1. Пусть выполнены условия 1–3. Тогда при любом $q\in R$ оператор $P_q(t,\alpha D_t)$ ограничен из H_α^{s+q} в H_α^s, $s\in R$.

Выясним далее некоторые свойства оператора, являющегося композицией операторов вида (1).

Теорема 2. Пусть выполнены условия 1–3. Тогда при любых $q_1,q_2\in R$ для любой функции $u\in H_\alpha^{s+q_1+q_2}$ справедлива оценка

$$\left\|P_{q_1}(t,\alpha D_t)P_{q_2}(t,\alpha D_t)u-P_{q_1+q_2}(t,\alpha D_t)u\right\|_{s,\alpha}\le C\|u\|_{s+q_1+q_2-1,\alpha}, \tag{3}$$

где $P_{q_1+q_2}$ – ПДО с символом $\lambda_1(t,\alpha(t)\eta)\lambda_2(t,\alpha(t)\eta),\lambda_1$ и λ_2 – соответственно символы операторов P_{q_1} и P_{q_2}.

Доказательство. Имеем

$$\begin{gathered}P_{q_1}P_{q_2}-P_{q_1+q_2}=\left(P_{q_1}P_{q_2}-P_{\alpha,q_1}P_{q_2}\right)+P_{\alpha,q_1}\left(P_{q_2}-P\alpha,q_2\right)+\\+\left(P_{\alpha,q_1}P_{\alpha,q_2}-P_{\alpha_1q_1+q_2}\right)+\left(P_{\alpha,q_1+q_2}-P_{q_1+q_2}\right).\end{gathered} \tag{4}$$

Оценивая левую часть (3) с помощью (4) и применяя теоремы В.2.2, В.2.3, и 1, получим неравенство (3). □

Таким образом, установлено, что операторы вида (1) образуют алгебру в том смысле, что главная часть композиции этих операторов также имеет вид (1). Причем символ этой главной части определяется как произведение символов операторов, находящихся в композиции.

На практике при исследовании вырождающихся эллиптических задач приходится иметь дело с пространствами типа Соболева–Слободецкого, нормы в которых наряду с весовыми производными содержат также обычные производные меньшего порядка.

Определение 1. Пространство $H^s_{\alpha,\sigma} = H^s_{\alpha,\sigma}(R)$, где $s \geq 0,\ \sigma > 1$, $[s/\sigma] \leq M+1$ ($[s/\sigma]$ – целая часть s/σ, M – из условия 2), состоит из функций $u \in H^{[s/\sigma]} \subset S'_\alpha$, для которых конечна норма

$$\|u\|^2_{s,(\alpha,\sigma)} = \sum_{\ell=0}^{[s/\sigma]} \left\|\partial^\ell_t u\right\|^2_{s-\sigma\ell,\alpha} + \|u\|^2_{[s/\sigma]}.$$

Определение 2. Пространство $H^{s,k}_\alpha = H^{s,k}_\alpha(R)$, где $s \geq 0$, $k \leq M+1,\ k \geq 0$ – целое, состоит из функций $u \in H^k \subset S'_\alpha$ с конечной нормой

$$\|u\|^2_{s,\alpha,k} = \sum_{j=0}^{\kappa} \left\|D^j_t u\right\|^2_{s,\alpha} + \|u\|^2_k.$$

В [137] показано, что пространства $H^s_{\alpha,\delta}$ и $H^{s,\kappa}_\alpha$ полны и в каждом из них плотно множество S. Результаты данного раздела могут быть перенесены и на эти пространства. Точнее, справедливы следующие утверждения.

Теорема 3. Пусть выполнены условия 1–3. Тогда для любой функции $\varphi \in S$ справедливы оценки

$$\left\|P_q(t,\alpha,D_t)\varphi - P_{\alpha,q}(t,D_{\alpha,t})\varphi\right\|_{s,(\alpha,\sigma)} \leq C\,\|\varphi\|_{s+q-1,(\alpha,\sigma)},$$

$$\left\|P_q(t,\alpha D_t)\varphi - P_{\alpha,q}(t,D_{\alpha,t})\varphi\right\|_{s,\alpha,k} \leq C\,\|\varphi\|_{s+q-1,\alpha,k},$$

где $s+q \geq 1$.

Следствие 2. При $s+q \geq 0$ оператор $P_q(t,\alpha D_t)$ ограничен из $H^{s+q}_{\alpha,\sigma}$ в $H^s_{\alpha,\sigma}$ и из $H^{s+q,k}_\alpha$ в $H^{s,k}_\alpha$.

Теорема 4. Пусть выполнены условия 1-3. Тогда при любых $q_1, q_2 \in R,\ s+q_1+q_2 \geq 1$ для любой функции $\varphi \in S$ справедливы неравенства

$$\left\|P_{q_1}(t,\alpha D_t)P_{q_2}(t,\alpha D_t)\varphi - P_{q_1+q_2}(t,\alpha D_t)\varphi\right\|_{s,(\alpha,\sigma)} \leq C\,\|\varphi\|_{s+q_1+q_2-1,(\alpha,\sigma)},$$

$$\left\|P_{q_1}(t,\alpha D_t)P_{q_2}(t,\alpha D_t)\varphi - P_{q_1+q_2}(t,\alpha D_t)\varphi\right\|_{s,\alpha,k} \leq C\,\|\varphi\|_{s+q_1+q_2-1,\alpha,k},$$

где ПДО $P_{q_1+q_2}$ определен в теореме 2.

Кратко остановимся на некоторых обобщениях полученных результатов. Все изложенные здесь теоремы остаются верными, если используемый класс символов $\mathbf{\Phi}_q$ заменить введенным Л. Хермандером более широким классом $S_{p,\delta}$ при $0 \leq \delta < \rho \leq 1$ (см. [108]).

Для рассмотрения n-мерного случая при $n > 1$ введем множество точек евклидова пространства (t,x), где

$t=(t_1,t_2,...,t_m)$, $x=(x_1,x_2,...,x_{n-m})$, $m\le n$. Введем также вектор-функцию $\alpha=(\alpha_1,...,\alpha_m)$, каждая компонента α_j которой зависит лишь от t_j. Предполагается, что все функции α_j удовлетворяют условиям 1–3. В этом случае можно рассмотреть ПДО вида

$$P_q(t,x,\alpha D_t,D_x)=F^{-1}_{\eta\to t}F^{-1}_{\xi\to x}\left[\lambda(t,x,\alpha(t)\eta,\xi)F_{x\to\xi}F_{t\to\eta}[\cdot]\right], \qquad (5)$$

где $\lambda(t,x,\eta,\xi)\in S^q_{\rho,\delta}$ $(0\le\delta<\rho\le1)$; $\alpha(t)\eta=(\alpha_1(t_1)\eta_1,...,\alpha_m(t_m)\eta_m)$.

Вся изложенная выше методика аналогично применяется к исследованию операторов вида (5) с помощью преобразования $F_\alpha=F_{\alpha_1}F_{\alpha_2}...F_{\alpha_m}$.

В.4. Вырождающиеся операторы с однородными символами

В этом разделе рассматриваются виды вырождающихся ПДО, которые находят непосредственное применение при исследовании общих граничных задач для эллиптических уравнений, включая задачу с косой производной. Основные их свойства определяются результатами раздела В.3, но доказательства требуют некоторых дополнительных рассмотрений.

На функциях $u\in L_2(R_1)$ введем оператор

$$Tu=F^{-1}_{\eta\to t}\left[\check{\theta}(\eta)F_{t\to\eta}[u]\right], \qquad (1)$$

где

$$\check{\theta}(\eta)=\begin{cases}1, & \eta>0\\ -1, & \eta<0\end{cases}=\frac{|\eta|}{\eta}=\operatorname{sgn}\eta .$$

Отметим, что оператор (1) совпадает с известным оператором Гильберта

$$Hu=v.p.\frac{1}{\pi i}\int_{R_1}\frac{u(\theta)}{t-\theta}d\theta$$

(см., например, [108])

Лемма 1. При $s\ge0$ оператор T ограничен в H^s_α.

Доказательство. Пусть $s\ge0$ – целое. Имеем

$$\|T\varphi\|_{s,\alpha}\le C\sum_{j=0}^{s}\left\|\alpha^j(t)D^j_tT\varphi\right\|,\ \ \varphi\in S_\alpha . \qquad (2)$$

Введем функцию

$$h\in C_0^\infty(R_1);\ h(\eta)=1,\ |\eta|\le1;\ h(\eta)=0,\ |\eta|\ge2. \qquad (3)$$

Тогда из (2) получим

$$\|T\varphi\|_{s,\alpha} \le C\sum_{j=1}^{s}\left\|F^{-1}_{\eta\to t}\left[\alpha(t)|\eta|\left(\alpha(t)\eta\right)^{j-1}\left(1-h\left(\alpha(t)\eta\right)\right)F_{t\to\eta}[\varphi]\right]\right\| + \\ +C\sum_{j=1}^{s}\left\|F^{-1}_{\eta\to t}\left[\left(\alpha(t)\eta\right)^{j}h\left(\alpha(t)\eta\right)F_{t\to\eta}[Tu]\right]\right\| + C\|Tu\|. \tag{4}$$

Учитывая, что $|\eta|\eta^{j-1}\left(1-h(\eta)\right)\in\Phi_j$, $\eta^j h(\eta)\in\Phi_0$, из (4) с помощью следствия B.3.1 получим

$$\|T\varphi\|_{s,\alpha} \le C\left(\sum_{j=1}^{s}\|\varphi\|_{j,\alpha} + \|\varphi\|\right) \le C_1\|\varphi\|_{s,\alpha}. \tag{5}$$

Отсюда, пользуясь интерполяционными свойствами пространств H^s_α (см. [105]) и плотностью S_α в H^s_α, получаем утверждение леммы. □

Поскольку оператор T коммутирует с производной ∂^ℓ_t, из леммы 1 следующее утверждение.

Следствие 1. При $s\ge 0$ оператор T ограничен в $H^s_{\alpha,\sigma}$ и $H^{s,k}_\alpha$.

Замечание 1. Аналогичным образом доказывается ограниченность в $H^s_{\alpha,\sigma}$ и $H^{s,k}_\alpha$ $(s\ge 0)$ ПДО с любым символом вида

$$\check{\theta}(\eta) = \begin{cases} C_1, & \eta > 0 \\ C_2, & \eta < 0 \end{cases}, \tag{6}$$

где C_1 и C_2 – некоторые константы.

Введем в S_α вырождающийся ПДО $M_q(\alpha D_t)$ ($q\ge 0$ – целое) с символом $\lambda\left(\alpha(t)\eta\right) = |\alpha(t)\eta|^q$, определяемый равенством (B.3.1). Как и раньше (см. раздел B.2), обозначим через $\Lambda^q_\alpha\left(D_{\alpha,t}\right)$ ВПДО с символом

$$\lambda(\eta) = \Lambda^q_\alpha(\eta) = \left(1+\eta^2\right)^{q/2}. \tag{7}$$

Докажем теоремы сравнения для операторов $M_q(\alpha D_t)$ и $\Lambda^q_\alpha\left(D_{\alpha,t}\right)$.

Теорема 1. Пусть выполнены условия 1–3. Тогда для любой функции $u\in H^{s+q}_\alpha$ ($s\ge 0$, $q\ge 1$ – целое) справедлива оценка

$$\left\|M_q(\alpha D_t)u - \Lambda^q_\alpha\left(D_{\alpha,t}\right)u\right\|_{s,\alpha} \le C\|u\|_{s+q-1,\alpha}. \tag{8}$$

Доказательство. Пусть h – функция из (3).

Если обозначить $M_{\alpha,q}\left(D_{\alpha,t}\right)$ ВПДО с символом

$$\lambda(\eta) = |\eta|^q, \tag{9}$$

$\check{T}_{\alpha,q}\left(D_{\alpha,t}\right)$ – ВПДО с символом

$$\lambda(\eta) = h(\eta)|\eta|^q, \tag{10}$$

то легко проверить справедливость оценок

$$\left\| \Lambda_\alpha^q \left(D_{\alpha,t}\right) u - M_{\alpha,q}\left(D_{\alpha,t}\right) u \right\|_{s,\alpha} \le C \left\| u \right\|_{s+q-1,\alpha}; \tag{11}$$

$$\left\| \check{T}_{\alpha,q}\, u \right\|_{s,\alpha} \le C \left\| u \right\|. \tag{12}$$

При четном q оценка (8) вытекает непосредственно из теоремы 1 и неравенства (11). При нечетном q справедливо

$$\frac{|\eta|^q}{\eta^q} = \frac{|\eta|}{\eta}. \tag{13}$$

Учитывая (13), для $\varphi \in S_\alpha$ имеем

$$\begin{gathered} M_q\left(\alpha D_t\right)\varphi = F^{-1}_{\eta\to t}\left[\left(\alpha(t)\eta\right)^q \left(1 - h\left(\alpha(t)\eta\right)\right) F_{t\to\eta}[\varphi]\right] + \\ + F^{-1}_{\eta\to t}\left[\alpha^q(t)\eta^q h\left(\alpha(t)\eta\right) F_{t\to\eta}[T\varphi]\right], \end{gathered} \tag{14}$$

где оператор T определен в (1). Используя представление (14), а также теорему 1 и следствие 1, получим

$$\begin{gathered} \left\| M_q\left(\alpha D_t\right)\varphi - \Lambda_\alpha^q\left(D_{\alpha,t}\right)\varphi \right\|_{s,\alpha} \le \left\| M_{\alpha,q}\left(D_{\alpha,t}\right)\varphi - \Lambda_\alpha^q\left(D_{\alpha,t}\right)\varphi \right\|_{s,\alpha} + \\ + C\left(\left\|\varphi\right\|_{s+q-1,\alpha} + \left\| \check{T}_{\alpha,q}\,\varphi \right\|_{s,\alpha} + \left\| T\varphi \right\|_{s,\alpha} \right). \end{gathered} \tag{15}$$

Оценивая правую часть (15) с помощью (11), (12) и (6), получим (8). □

Лемма 2. Для любой функции $\varphi \in S^M$ при $q \ge 1$, $\ell = 0,1,\ldots,M+1$ справедлива оценка

$$\left\| \partial_t^\ell M_q\left(\alpha D_t\right)\varphi - M_q\left(\alpha D_t\right)\partial_t^\ell \varphi \right\|_{s,\alpha} \le C \sum_{\ell'=0}^{\ell} \left\| \partial_t^{\ell'} \varphi \right\|_{s+q-1,\alpha}. \tag{16}$$

Доказательство. Очевидно, достаточно рассмотреть случай нечетного q. Имеем

$$\partial_t^\ell M_q\left(\alpha D_t\right)\varphi = \partial_t^\ell \alpha^q(t) D_t^q T\varphi, \tag{17}$$

где оператор T определен в (1). Учитывая представление

$$\partial_t^\ell \alpha^q(t) D_t^q = \alpha^q(t) D_t^q \partial_t^\ell + \sum_{j=1}^{q-1} \sum_{\ell'=0}^{\ell} a_{j,\ell'}^{q,\ell}(t) D_{\alpha,t}^j \partial_t^{\ell'}, \tag{18}$$

где функции $a_{j,\ell'}^{q,\ell}$ зависят лишь от функции α и ее производных, из (17) получим оценку

$$\left\| \partial_t^\ell M_q\left(\alpha D_t\right)\varphi - M_q\left(\alpha D_t\right)\partial_t^\ell \varphi \right\|_{s,\alpha}^+ \le C \sum_{j=0}^{q-1} \sum_{\ell'=0}^{\ell} \left\| D_{\alpha,t}^j T \partial_t^{\ell'} \varphi \right\|_{s,\alpha}^+, \tag{19}$$

откуда с помощью леммы 1 вытекает (16). □

Замечание 2. Все результаты, полученные в настоящем разделе для пространств функций на R_1, переносятся без затруднений на R_n. Поэтому следующие две теоремы мы сформулируем сразу для пространства R_n.

Теорема 2. Пусть выполнены условия 1–3. Тогда для любой функции $u \in H^{s+q}_{\alpha,\sigma}(R_n)$ ($q \geq 1$ – целое) справедлива оценка

$$\left\| M_q(\alpha D_t)u - \Lambda^q_\alpha\left(D_{\alpha,t}\right)u \right\|_{s,(\alpha,\sigma)} \leq C\|u\|_{s+q-1,(\alpha,\sigma)}. \tag{20}$$

Теорема 3. Пусть выполнены условия 1–3. Тогда для любой функции $u \in H^{s+q,k}_\alpha(R_n)$ ($s \geq 0$, $q \geq 1$ – целое) выполняется оценка

$$\left\| M_q(\alpha D_t)u - \Lambda^q_\alpha\left(D_{\alpha,t}\right)u \right\|_{s,k,\alpha} \leq C\|u\|_{s+q-1,k,\alpha}. \tag{21}$$

Теоремы 2 и 3 легко доказываются с помощью теоремы 1 и леммы 2. □

При исследовании граничных задач находят применение вырождающиеся ПДО с однородными символами в R_n. Пусть $\lambda(\eta,\xi)$ – положительно однородная функция степени $q \geq 0$, q – целое, бесконечно дифференцируемая при $|\eta|+|\xi|>0$ ($\eta \in R_1, \xi \in R_{n-1}$). Легко проверить, что такая функция удовлетворяет оценкам

$$\left|\partial^j_\eta \partial^\beta_\xi \lambda(\eta,\xi)\right| \leq C_{j,\beta}\left(\eta^2+|\xi|^2\right)^{\frac{q-j-|\beta|}{2}}, \ |\eta|+|\xi|>0, \ j+|\beta|=0,1,\ldots \tag{22}$$

На функциях $\varphi \in S_\alpha(R_n)$ рассмотрим оператор

$$\alpha^q(t)A(D_tD_x)\varphi = \alpha^q(t)F^{-1}_{\eta\to t}F^{-1}_{\xi\to x}\left[\lambda(\eta,\xi)F_{x\to\xi}F_{t\to\eta}[\varphi]\right] \tag{23}$$

с описанным однородным символом λ. Обозначим также

$$\alpha^q(t)A(D_t,0)\varphi = \alpha^q(t)F^{-1}_{\eta\to t}F^{-1}_{\xi\to x}\left[\lambda(\eta,0)F_{x\to\xi}F_{t\to\eta}[\varphi]\right]. \tag{24}$$

Лемма 3. Для любой функции $\varphi \in S_\alpha$ при целом $q \geq 1$ справедливо представление

$$\alpha^q(t)A(D_t,D_x)\varphi - \alpha^q(t)A(D_t,0)\varphi = \alpha(t)\check{P}_q\varphi, \tag{25}$$

где оператор $\check{P}_q$ ограничен из H^{s+q}_α в H^s_α, $s \geq 0$.

Доказательство. С помощью формулы Тейлора запишем

$$\lambda(\eta,\xi) = \lambda(\eta,0) + \sum_{1\leq|\beta|\leq q-1}\frac{1}{\beta!}\partial^\beta_\xi\lambda(\eta,\xi)\Big|_{\xi=0}\xi^\beta + \sum_{|\beta|=q} r_\beta(\eta,\xi)\xi^\beta, \tag{26}$$

где $|\eta|>0$,

$$r_\beta(\eta,\xi) = \frac{q}{\beta!}\int_0^1 \partial^\beta_\rho\lambda(\eta,\rho)\Big|_{\rho=\kappa\xi}(1-\kappa)^{q-1}\,d\kappa. \tag{27}$$

Нетрудно проверить, что в силу положительной однородности функции λ справедливы представления

$$\partial_\xi^\beta \lambda(\eta,\xi)\Big|_{\xi=0} = \eta^{q-|\beta|}\check{\theta}_\beta(\eta),\ |\eta|>0,\ 1\le|\beta|\le q-1, \tag{28}$$

где каждая из функций $\check{\theta}_\beta$ имеет вид (6).

Пусть h – функция, определенная в (3). Тогда из (26) и (28) получим следующее равенство, справедливое при всех $\eta\in R^1$, $\xi\in R^{n-1}$:

$$\lambda(\eta,\xi)-\lambda(\eta,0) = h(\eta)\lambda(\eta,\xi)-h(\eta)\lambda(\eta,0)+$$
$$+\sum_{1\le|\beta|\le q-1}\frac{1}{\beta!}\left\{\eta^{q-|\beta|}-h(\eta)\eta^{q-|\beta|}\right\}\check{\theta}_\beta(\eta)+\left(1-h(\eta)\right)\sum_{|\beta|=q} r_\beta(\eta,\xi)\xi^\beta, \tag{29}$$

причем

$$\left|\partial_\eta^j\left\{\left(1-h(\eta)\right)r_\beta(\eta,\xi)\right\}\right| \le C_j\left(1+\eta^2\right)^{-j/2},\ j=0,1,\ldots \tag{30}$$

Пользуясь (23), (24) и (29), запишем

$$\alpha^q(t)A(D_t,D_x)\varphi-\alpha^q(t)A(D_t,0)\varphi=$$
$$=\alpha^q(t)F^{-1}_{\eta\to t}F^{-1}_{\xi\to x}\left[h(\eta)\lambda(\eta,\xi)F_{x\to\xi}F_{t\to\eta}[\varphi]\right]-$$
$$-\alpha^q(t)F^{-1}_{\eta\to t}\left[h(\eta)\lambda(\eta,0)F_{t\to\eta}[\varphi]\right]+\sum_{1\le|\beta|\le q-1}\frac{1}{\beta!}\alpha^{|\beta|}(t)\alpha^{q-|\beta|}(t)D_t^{q-|\beta|}\check{T}_\beta\varphi- \tag{31}$$
$$-\alpha^q(t)\sum_{1\le|\beta|\le q-1}\frac{1}{\beta!}F^{-1}_{\eta\to t}\left[h(\eta)\eta^{q-|\beta|}F_{t\to\eta}\left[\check{T}_\beta\varphi\right]\right]+\alpha^q(t)\sum_{|\beta|=q}R_\beta D_x^\beta\varphi,$$

где $\check{T}_\beta$ – ПДО с символом $\check{\theta}_\beta(\eta)$; R_β – ПДО с символом $\left(1-h(\eta)\right)r_\beta(\eta,\xi)$.

Пользуясь замечанием 1, легко получить следующие оценки:

$$\left\|F^{-1}_{\eta\to t}F^{-1}_{\xi\to x}\left[h(\eta)\lambda(\eta,\xi)F_{x\to\xi}F_{t\to\eta}[\varphi]\right]\right\|_{s,\alpha}\le$$
$$\le C\left\|F^{-1}_{\eta\to t}F^{-1}_{\xi\to x}\left[h(\eta)\lambda(\eta,\xi)F_{x\to\xi}F_{t\to\eta}[\varphi]\right]\right\|_{s}\le \tag{32}$$
$$\le C_1\left\|\left(1+|\xi|^2\right)^{\frac{s+q}{2}}F_{x\to\xi}[\varphi]\right\|\le C_2\|\varphi\|_{s+q,\alpha};$$

$$\left\|F^{-1}_{\eta\to t}\left[h(\eta)\lambda(\eta,0)F_{t\to\eta}[\varphi]\right]\right\|_{s,\alpha}\le C\|\varphi\|_{s,\alpha}; \tag{33}$$

$$\left\|F^{-1}_{\eta\to t}\left[h(\eta)\eta^{q-|\beta|}F_{t\to\eta}\left[\check{T}_\beta\varphi\right]\right]\right\|_{s,\alpha}\le C\|\varphi\|_{s,\alpha}, \tag{34}$$

где $1\le|\beta|\le q-1$.

С помощью замечания 1 имеем также оценку

$$\left\| \alpha^{q-|\beta|}(t) D_t^{q-|\beta|} \overset{\vee}{T}_\beta \varphi \right\|_{s,\alpha} \leq C \|\varphi\|_{s+q-1,\alpha}, \tag{35}$$

где $1 \leq |\beta| \leq q-1$.

Наконец, с помощью (30) можно доказать неравенство

$$\left\| R_\beta D_x^\beta \varphi \right\|_{s,\alpha} \leq C \left\| D_x^\beta \varphi \right\|_{s,\alpha} \leq C \|\varphi\|_{s+q,\alpha}, \tag{36}$$

где $|\beta| = q$.

Утверждение леммы вытекает из представления (31) и оценок (32)–(36). □

Следствие 2. Для любого $\varepsilon > 0$ существует $\delta > 0$ такое, что при любой $h(t) \in C_0^\infty([-\delta, \delta])$, $|h(t)| \leq 1$ справедлива оценка

$$\left\| h(t) \left\{ \alpha^q(t) A(D_t, D_x) u - \alpha^q(t) A(D_t, 0) u \right\} \right\|_{s,\alpha} \leq \varepsilon \|u\|_{s+q,\alpha}, \tag{37}$$

где $s \geq 0$, $q \geq 1$ – целое, $u \in H_\alpha^{s+q}$.

Неравенство (37) вытекает из леммы 3 и легко проверяемой оценки

$$\left\| h(t) \alpha(t) \varphi \right\|_{s,\alpha} \leq \varepsilon \|\varphi\|_{s,\alpha},$$

где $\varphi \in S_\alpha$; $\varepsilon > 0$ – любое число, определяемое величиной δ.

Аналогично лемме 3 нетрудно установить, что оператор $\overset{\vee}{P}_q$ в равенстве (25) ограничен из $H_{\alpha,\sigma}^{s+q}$ в $H_{\alpha,\sigma}^s$, а также из $H_\alpha^{s+q,k}$ в $H_\alpha^{s,k}$ $(s \geq 0)$. Исходя из этих фактов доказывается

Следствие 3. Для любого $\varepsilon > 0$ существует $\delta > 0$ такое, что при целом $q \geq 1$ и любой $h(t) \in C_0^\infty([-\delta, \delta])$, $|h(t)| \leq 1$ справедливы оценки

$$\begin{gathered} \left\| h(t) \left\{ \alpha^q(t) A(D_t, D_x) u - \alpha^q(t) A(D_t, 0) u \right\} \right\|_{s,(\alpha,\sigma)} \leq \varepsilon \|u\|_{s+q,(\alpha,\sigma)} + \\ + C(\varepsilon) \|u\|, \quad \forall u \in H_{\alpha,\sigma}^{s+q}, \ s \geq 0; \end{gathered} \tag{38}$$

$$\begin{gathered} \left\| h(t) \left\{ \alpha^q(t) A(D_t, D_x) u - \alpha^q(t) A(D_t, 0) u \right\} \right\|_{s,k,\alpha} \leq \varepsilon \|u\|_{s+q,k,\alpha} + \\ + C(\varepsilon) \|u\|, \quad \forall u \in H_\alpha^{s+q,k}, \ s \geq 0. \end{gathered} \tag{39}$$

Неравенства (38) и (39) полезны при доказательстве оценок решений вырождающихся псевдодифференциальных уравнений в соответствующих пространствах.

БИБЛИОГРАФИЯ

1. *Agarwal R. A* Petri-Net based approach for verifying the integrity of production systems / R. A. Agarwal // International Journal of Man-Machine Studies. – 1992. – № 36. – P. 447–468.
2. *Aho A. V.* The transitive reduction of a directed graph. SIAM J. Computing 1 : 2 / A. V. Aho, M. R. Garey, J. D. Ulman, 1972. – P. 131–137.
3. *Andréka H.* Algebraic Logic / H. Andréka, J. D. Monk, I. Németi. – Dordrecht : North-Holland Publ. Co, 1991.
4. *Armstrong W. W.* Dependency Structure of Database Relationships / W. W. Armstrong // Proc. IFIP Congress. – Geneva, 1974. – P. 580–583.
5. *Avritzer A.* Estimating the CPU utilization of a rule-based system / A. Avritzer, J. P. Ros, E. J. Weyuker // In Proceedings of the 4th international Workshop on Software and Performance (Redwood Shores, California, January 14–16, 2004). ACM. – New York : NY, 2004. – P. 1–12.
6. *Berthelot G.* Transformations and decompositions of nets. Lect. Notes Comput. Sci. Springer-Verlag / G. Berthelot. – Berlin, 1987. – P. 359–377.
7. *Boros E.* Horn minimization by iterative decomposition / E. Boros, O. Čepek, A. Kogan // Annals of Mathematics and Artificial Intelligence 23, 3–4 Jan., 1998. – P. 321–343.
8. *Buchanan B. G.* Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project / B. G. Buchanan, E. H. Shortliffe. – Addison-Wesley, 1984.
9. *Buchberger B. Algebraic Simplification* / B. Buchberger, R. Loos // Computer Algebra – Symbolic and Algebraic Computation / eds. B. Buchberger, G. E. Collins, R. Loos. – Vienna – New York : Springer-Verlag, 1982. – P. 11–43.
10. *Buod T.* Blending imperative and relational programming / T. Buod // IEEE Software. – 1991. – V. 8. – № 1. – P. 58–65.
11. *Chen G.* Executing Pascal programs on a Prolog architecture / G. Chen, M. H. Williams // Inf. and Software Technology. – 1987. – V. 29. – № 6. – P. 285–290.
12. *Cheng A. M.* Self-Stabilizing Real-Time OPS5 Production Systems / A. M. Cheng, S. Fujii // IEEE Transactions on Knowledge and Data Engineering. – 2004. – V. 16. – №. 12. – P. 1543–1554.

13. *Cheng A. M.* A Graph-Based Approach for Timing Analysis and Refinement of OPS5 Knowledge-Based Systems / A. M. Cheng, H.-Y. Tsai // IEEE Transactions on Knowledge and Data Engineering. – 2004. – Vol. 16. – № 2. – P. 271–288.
14. *Czelakowski Janusz* Monotone Relations, Fixed Points and Recursive Definitions / Janusz Czelakowski // Towards Mathematical Philosophy / eds. David Makinson, Jacek Malinowski, Heinrich Wansing. – Berlin : Springer, 2009. – P. 125–164.
15. *Davis R.* An overview of production systems / R. Davis, J. King // Mahine Intelligence. – Chichester : Ellis Horwood Limited. – 1977. – Vol. 8. – P. 300–332.
16. *De Baets B.* Analytical Solution Methods for Fuzzy Relational Equations / B. De Baets // Fundamentals of Fuzzy Sets / eds. D. Dubois, and H. Prade. – Boston : Kluwer Academic Publishers, 2000. – P. 291–340.
17. *Dershowitz N.* Termination of rewriting / N. Dershowitz // J. Symbolic Comput. – 1987. – V. 3. – P. 69–116.
18. *Dershowitz N. A* Taste of Rewrite Systems. In Functional Programming, Concurrency, Simulation and Automated Reasoning: international Lecture Series 1991–1992, Mcmaster University, Hamilton, Ontario, Canada / Dershowitz N. ; ed. P. E. Lauer // Lecture Notes In Computer Science. – London : Springer-Verlag. – V. 693. – P. 199–228.
19. *Dershowitz N.* Canonical Conditional Rewrite Systems / N. Dershowitz, M. Okada, G. Sivakumar // Proceedings of the 9th international Conference on Automated Deduction (May 23–26, 1988) / eds E. L. Lusk, R. A. Overbeek // Lecture Notes In Computer Science. – London : Springer-Verlag. – 1988. –Vol. 310. – P. 538–549.
20. *Doorenbos R. B.* Production Matching for Large Learning Systems. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-22942., Carnegie Mellon University / R. B. Doorenbos, 1995.
21. *Erwing M.* Specifying Type Systems with Multi-Level Order-Sorted Algebra. 3rd Int. Conf. On Algebraic Method. and Software Technol / M. Erwing, 1993, P. 179–188.
22. *Erwing M., Guting R.* Explicit Graphs in a Functional Model for Spatial databases. Report 110, Fern University Hagen. – 1991. Revised Version. 1993.
23. *Fokkink W.* Lazy rewriting on eager machinery / W. Fokkink, J. Kamperman P. Walters // ACM Trans. Program. Lang. Syst. 22, 1 (Jan. 2000). – P. 45–86.
24. *Forgy C. L.* OPS5 user's manual. Technical Report CMU-CS-81-135, Computer Science Department, Carnegie Mellon University, 1981.
25. *Forgy C. L.* Rete: A fast algorithm for the many pattern/many object pattern match problem / C. L. Forgy // Artificial Intelligence. – 1982. – № 19(1). – P. 17–37.
26. *Ganter B.* Formal Concept Analysis: mathematical foundations / B. Ganter, R. Wille. – Heidelberg : Springer, 1999.

27. *Ginsberg A.* Knowledge-Base Reduction: A New Approach to Checking Knowledge Bases for Inconsistency & Redundancy / A. Ginsberg // Proceedings of the Seventh National Conference on Artificial Intelligence, 1988. – P. 585–589.
28. *Ginsberg M.* Multivalued logics / M. Ginsberg // Proceedings of AAAI-86. Fifth National Conference on Artificial Intellegence. – Los Altos : Morgan Kaufman Publishers, 1986. – P. 243–247.
29. *Giraud-Carrier C. G.* An Integrated Framework for Learning and Reasoning / C. G. Giraud-Carrier, T. R. Martinez // Journal of Artificial Intelligence Research. – 1995. – V. 3. – P. 147–185.
30. *Glower F.* Logical Testing for Rule-Based Management / F. Glower, H. J. Greenberg // Annals of Operations Research. – 1988. – № 12. – P. 199–215.
31. *Godin R.* Formal Concept Analysis-Based Class Hierarchy Design in Object-Oriented Software Development / R. Godin, P. Valtchev // Formal Concept Analysis / eds. B. Ganter, G. Stumme, R. Wille // Lecture Notes In Computer Science. – Springer Berlin/Heidelberg. – 2005. – V. 3626. – P. 304–323.
32. *Gordon J.* The Dempster-Shaffer Theory of Evidence / J. Gordon, E. H. Shortliffe ; eds. B. G. Buchanan and E. H Shortliffe // Rule-Based Expert Systems: MYCIN Experiments, Addison-Wesley Reading, 1984.
33. *Gupta A.* Parallelism in Production Systems / A. Gupta. – Los Altos : Morgan Kaufmann, CA, 1987.
34. *Gupta U. G.* Validation and verification of knowledge-based systems: A survey / U. G. Gupta // Journal of Applied Intelligence. – 1993. – V. 3. – P. 343–363.
35. *Hájek P.* A generalized algebraic approach to uncertainty processing in rule-based expert systems (dempsteroids) / P. Hájek, J. V. Valdes // Comput. Artif. Intell. – 1991. – V. 10, N. 1. – P. 29–42.
36. *Halib M.* Bit-vector encoding for partially ordered sets / M. Halib, L. Nourine // Lect. Notes Comput. Sci. – 1994. – V. 831. – P. 1–12.
37. *Halmos P.* Algebraic logic, IV: Equality in polyadic algebras / P. Halmos // Trans. Amer. Math. Soc. – 1957. – N. 86. – P. 1–27.
38. *Halmos P.* Algebraic Logic / P. Halmos. – New York : Chelsea Publishing Co, 1962.
39. *Hammer P. L.* Optimal compression of propositional Horn knowledge bases: complexity and approximation / P. L. Hammer, A. Kogan // Artif. Intell. – 1993. – V. 64, N. 1. – P. 131–145.
40. *Hammer P. L.* Essential and redundant rules in Horn knowledge bases / P. L. Hammer, A. Kogan // Proceedings of the 28th Hawaii international Conference on System Sciences (Hicss'95) (January 04–07, 1995). HICSS. IEEE Computer Society, Washington, DC, 209. – 1995.
41. *Henkin L.* Cylindric algebras / L. Henkin, A. Tarski // Proc. of Symposia in Pure Mathematics II (1961), Lattice theory. – P. 83–113.
42. *Homeier P. V.* ECLIPS: An extended CLIPS for backward chaining and goal-directed reasoning / P. V. Homeier, T. C. Le // NASA. Johnson

Space Center, Second CLIPS Conference Proceedings. – 1991. – Vol. 2. – P. 273–283.

43. *Hsiang J.* Refutational theorem proving using term-rewriting systems / J. Hsiang // Artif. Intell. – 1985. – V. 25. – P. 255–300.

44. *Kadar B.* Approaches to Increase the Performance of Agent-Based Production Systems / B. Kadar, L. Monostori // Proceedings of the 14th international Conference on industrial and Engineering Applications of Artificial intelligence and Expert Systems: Engineering of intelligent Systems (June 04–07, 2001) / eds. L. Monostori, J. Váncza, M. Ali // Lecture Notes In Computer Science. – London : Springer-Verlag. – 2001. – Vol. 2070. – P. 612–621.

45. *Kang J. A.* Shortening Matching Time in OPS5 Production Systems / J. A. Kang, A. M. Cheng // IEEE Transactions on Software Engineering. – July 2004. – Vol. 30, № 7. – P. 448–457.

46. *Klop J. W.* Term rewriting systems / J. W. Klop // Handbook of Logic in Computer Science (Vol. 2): Background: Computational Structures / eds S. Abramsky, D. M. Gabbay and S. E. Maibaum // Osborne Handbooks of Logic In Computer Science. – New York : Oxford University Press. – 1992. – Vol. 2. – P. 1–116.

47. *Landveber L. H., Robertson E. L.* Properties of conflict-free and persistent Petri nets / L. H. Landveber, E. L .Robertson // J. ACM. – 1978. – Vol. 25, № 3. – P. 352–364.

48. *Lee S.* Developing a strategy for expert system verification and validation / S. Lee, R. M. O'Keefe // IEEE Trans. on Systems, Man and Cybernetics. – 1994. – Vol. 24. – P. 643–655.

49. *Lee Y. H.* Integration of Forward and Backward Inferences Using Extended Rete Networks / Y. H. Lee, S. I. Yoo // Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 9th, Proceedings of the Ninth International Conference. – 1997. – P.339–345.

50. *Lee Y. H.* Optimizing Real-Time Equational Rule-Based Systems / Y.-H. Lee, A. M. Cheng // IEEE Transactions on Software Engineering. – Feb. 2004. – Vol. 30, № 2. – P. 112–125.

51. *Liberatore P.* Redundancy in logic II: 2CNF and Horn propositional formulae / P. Liberatore // Artificial Intelligence. – Feb. 2008. – Vol. 172, № 2–3. – P. 265–299.

52. *Liu N. K.* An Approach Towards the Verification of Expert Systems Using Numerical Petri Nets / N. K. Liu, T. Dillon // International Journal of Intelligent Systems. – 1991. – № 6. – P. 255–276.

53. *Lock H.C.R.* Issues in the implementation of Prolog and their optimization / H. C. R. Lock, A. Martins // Microprocess and Microprogram. – 1991. – Vol. 32, № 2–5. – P. 505–514.

54. *Makhortov S.* Multi-level LP-Structures in Rewriting Systems / S. Makhortov // Mathematical Modelling and Computational Physics (MMCP'2009): Book of Abstracts of the International Conference (Dubna, July 7–11, 2009). – Dubna : JINR, 2009. – P. 123.

55. *Maciol A.* An application of rule-based tool in attributive logic for business rules modeling / A. Maciol // Expert Systems with Applications. – April 2008. – Vol. 34, №. 3. – P. 1825–1836.
56. *Mens T.* A Survey of Software Refactoring / T. Mens, T. Tourw'e // IEEE Trans. on Software Engineering. – Feb. 2004. – Vol. 30(2). – P. 126–139.
57. *Metivier Y.* About the Rewriting Systems Produced by the Knuth-Bendix Completion Algorithm / Y. Metivier // Information Processing Letters 16, 1983.
58. *Miranker D.* Performance estimates for the DADO machine: A comparison of TREAT and Rete / D. Miranker // Fifth Generation Computer Systems, ICOT. – Tokyo, 1984.
59. *Miranker D.* On the Performance of Lazy Matching in Production Systems / D. Miranker, D. Brant, B. Lofaso and D. Gadbois // Proc. National Conference on Artificial Intelligence, 1990.
60. *Monk J. D.* Handbook of Boolean Algebras / J. D. Monk. – Amsterdam : North-Holland Co, 1989. – Vols. I-III
61. *Munro I.* Efficient determination of the transitive closure of a directed graph / I. Munro // Inf. Processing. Letters. – 1971. – V. 1. – P. 56–58.
62. *Neiman D. E.* Issues in the Design and Control of Parallel Rule-Firing Production Systems / D. E. Neiman // J. Parallel Distrib. Comput. – 1994. – № 23. – P. 338–363.
63. *Nemeti I.* Algebraization of Quantifier Logics; an Overview / I. Nemeti // Studia Logica L. – 1991. – nos 3/4. – P. 485–569.
64. *Nguyen T. A.* Knowledge Base Verification / T. A. Nguyen, W. A. Perkins, T. J. Laffey and D. Pecora // AI Magazine. – 1987. – V. 8, № 2. – P. 69–75.
65. *Ohlebusch E.* Transforming Conditional Rewrite Systems with Extra Variables into Unconditional Systems / E. Ohlebusch // Proceedings of the 6th international Conference on Logic Programming and Automated Reasoning (September 6-10, 1999) / eds H. Ganzinger, D. A. McAllester and A. Voronkov // Lecture Notes In Computer Science. – London : Springer-Verlag. – Vol. 1705. – P.111–130.
66. *Oles F. J.* An Application of Lattice Theory to Knowledge Representation / F. J. Oles // Theor. Comput. Sci. 249, 1 (Oct. 2000). – P. 163–196.
67. *Pederson K.* Well-structured knowledge bases / K. Pederson // AI Expert. – 1989. – № 4. – P. 44–55.
68. *Poincaré H.* Lecons de Mechanique céleste / H. Poincaré. – Paris, 1910. – V. 3.
69. *Poli R.* Backward-chaining evolutionary algorithms / R. Poli, W. B. Langdon // Artificial Intelligence. – August 2006. – Vol. 170, № 11. – P. 953–982.
70. *Purdom P.* A transitive closure algorithm / P. Purdom // BIT. – 1970. – Vol. 10. – P. 76–94.
71. *Sahni S.* Computationally related problems / S. Sahni // SIAM J. Computing 3 : 2. – 1974. – P. 262–279.

72. *Schmitt P. H.* A Survey of Rewrite Systems / P. H. Schmitt // Proceedings of the 1st Workshop on Computer Science Logic (October 12–16, 1987) / eds. E. Börger, H. K. Büning and M. M. Richter // Lecture Notes In Computer Science. – London : Springer-Verlag. – Vol. 329. – P. 235–262.
73. *Schmolze J. G.* Guaranteeing Serializable Results in Synchronous Parallel Production Systems / J. G. Schmolze // J. Parallel Distrib. Comput. – 1991. – № 13(4). – P. 348–365.
74. *Schmolze J. G.* Detecting redundancy among production rules using term rewrite semantics / J. G. Schmolze, W. Snyder // Knowledge-Based Systems. – 1999. – № 12. – P. 3–11.
75. *Scott D.* Data Types as Lattices / D. Scott // SIAM J. Comput. – 1976. – Vol. 5, Issue 3. – P. 522–587.
76. *Shafer G.* A Mathematical Theory of Evidence / G. Shafer. – Princeton University Press, 1976.
77. *Sowa J. F.* Conceptual Structures: Information Processing in Mind and Machine / J. F.Sowa. – Reading, MA : Addison-Wesley, 1984.
78. *Sowa J. F.* Knowledge Representation: Logical, Philosophical and Computational Foundations / J. F. Sowa. – Brooks Cole Publishing Co., Pacific Grove, CA,1999.
79. *Sowyer B.* Programming Expert Systems in Pascal / B. Sowyer, D. Foster. –John Wiley & Sons, Inc., 1986.
80. *Tomić B.* JavaDON: an open-source expert system shell / B. Tomić, H. Jovanović, V. Devedžić // Expert Systems with Applications. – October 2006. – Vol. 31. – № 3. – P. 595–606.
81. *Toyama Y.* On Equivalence Transformations for Term Rewriting Systems / Y. Toyama // Proceedings of the 1983 and 1984 RIMS Symposia on Software Science and Engineering II / eds E. Goto, K. Araki and T. Yuasa // Lecture Notes In Computer Science. – London : Springer-Verlag. – 1986. – Vol. 220. – P. 44-61.
82. *Warren H. S.* A modification of Warshall's algorithm for transitive closure of binary relations / H. S. Warren // Communs. ACM. – 1975. – Vol.18, № 4. – P. 218–220.
83. *Warshall S.* A Theorem on Boolean matrices / S. Warshall // J. Assoc. Computing Machinery. – 1962. – Vol. 9. – P. 11–12.
84. *Zimmermann T.* Toward intelligent object-oriented scientific applications / T. Zimmermann, P. Bomme ; eds. B. H. Topping and Z. Bittnar // Engineering Computational Technology. – Edinburgh : Civil-Comp Press, UK, 2002. – P. 271–311.
85. *Анишев П. А.* Редуцируемость сетей Петри / П. А. Анишев // Программирование. – 1982. – № 4. – С. 36–43.
86. *Ануреев И. С.* Системы переписывания формул / И. С. Ануреев. – Новосибирск, 1997. – 22 с. – (Препр./РАН. Сиб. отд-ние. ИСИ; № 40).
87. *Арлазаров В. Л.* Об экономичном построении транзитивного замыкания ориентированного графа / В. Л. Арлазаров, Е. А. Диниц, М. А. Кронфорд, И. А. Фараджев // Докл. АН СССР. – 1970. – Т. 194, № 3. – С. 487–488.

88. *Афонин С. А.* Алгоритмы эффективного вычисления конъюнктивных регулярных путевых запросов / С. А. Афонин // Вычислительные технологии. – 2007. – № 12(2). – С. 23–32.
89. *Ахо А. В.* Компиляторы: принципы, технологии и инструменты : пер. с англ. / А. В. Ахо, Р. Сети, Дж. Д. Ульман. – М. : Вильямс, 2001. – 768 с.
90. *Ашинянц Р. А.* Логические методы в искусственном интеллекте / Р. А. Ашинянц. – М. : МГАПИ, 2001. – 204 с.
91. *Ашинянц Р. А.* Проект гибридной экспертной системы ОХРАНА / Р. А. Ашинянц, А. В. Засыпкин // Тез. докл. 2-й Всесоюзной школы-семинара «Экспертные системы и Пролог в учебном процессе». – Йошкар-Ола, 1990. – С. 4–9.
92. *Бениаминов Е. М.* Алгебраические методы в теории баз данных и представлении знаний / Е. М. Бениаминов. – М. : Научный мир, 2003. – 184 с.
93. *Биркгоф Г.* Теория решеток : пер. с англ. / Г. Биркгоф. – М. : Наука, 1984. – 568 с.
94. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами приложений на С++ : пер. с англ. / Г. Буч. – М. : Бином, 2000. – 560 с.
95. *Вагин В. Н.* Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин, Е. Ю. Головина, А. А. Загорянская, М. В. Фомина ; под ред. В. Н. Вагина, Д. А. Поспелова. – М. : Физматлит, 2004. – 704 с.
96. *Васенин В. А.* К разработке моделей эффективного поиска информации в сети Интернет / В. А. Васенин, С. А. Афонин // Сб. трудов Всероссийской научной конференции «Научный сервис в сети Интернет-2003». – М. : МГУ, 2003. – С. 252–255.
97. *Вишик М. И.* Эллиптические псевдодифференциальные операторы на замкнутом многообразии, вырождающиеся на подмногообразии / М. И. Вишик, В. В. Грушин // Докл. АН СССР. – 1969. – Т. 189, № 1. – С. 16–19.
98. *Владимиров В. С.* Уравнения математической физики / В. С. Владимиров. – М. : Наука, 1981. – 512 с.
99. *Воробьев С. Г.* Условные системы подстановок термов и их применение в проблемно-ориентированной верификации программ / С. Г. Воробьев : автореф. дис. . . . к. ф.-м. н. – Новосибирск : ВЦ СО АН СССР, 1987.
100. *Гарсиа Молина Г.* Системы баз данных. Полный курс : пер. с англ. / Г. Гарсиа Молина, Д. Ульман, Д. Уидом. – М. : Вильямс, 2002. – 1088 с.
101. *Глушко В. П.* Пространства функций с дробными весовыми производными и граничные задачи переменного порядка / В. П. Глушко // Корректные краевые задачи для неклассических уравнений математической физики. – Новосибирск, 1981. – С. 46–53.

102. *Глушко В. П.* Операторы неглавного типа и задача с косой производной / В. П. Глушко, С. Д. Махортов // Успехи мат. наук. – 1986. – Т. 41, № 4. – С. 202–203.
103. *Гмурман В. Е.* Теория вероятностей и математическая статистика : учеб. пособие для вузов / Гмурман В. Е. – 9-е изд. – М. : Высш. шк., 2003. – 479 с.
104. *Горбунов-Посадов М. М.* Расширяемые программы / М. М. Горбунов-Посадов. – М. : Полиптих, 1999. – 336 с.
105. *Грушин В. В.* Псевдодифференциальные операторы / В. В. Грушин. – М. : МИЭМ, 1975. – 107 с.
106. *Джексон П.* Введение в экспертные системы : пер. с англ. / П. Джексон. – М. : Вильямс, 2001. – 624 с.
107. *Джосьютис Н.* C++ Стандартная библиотека. Для профессионалов : пер. с англ. / Н. Джосьютис. – СПб. : Питер, 2004. – 730 с.
108. *Егоров Ю. В.* Линейные дифференциальные уравнения главного типа / Ю. В. Егоров. – М. : Наука, 1984. – 400 с.
109. *Еремеев А. П.* О корректности продукционной модели принятия решений на основе таблиц решений / А. П. Еремеев // Автоматика и телемеханика. – 2001. – № 10. – С. 78–90.
110. *Жоголев Е. А.* Объектная организация систем гиперпрограммирования / Е. А. Жоголев // Программирование. – 1997. – № 5. – С. 24–32.
111. *Закревский А. Д.* Логические уравнения / А. Д. Закревский. – изд. 2-е, стереотипное. – М. : Едиториал УРСС, 2003. – 96 с.
112. *Замулин А. В.* Алгебраическая семантика императивного языка программирования / А. В. Замулин // Программирование. – 2003. – № 6. – С. 51–64.
113. *Засыпкин А. В.* Экспертная система поддержки решений по рациональной организации охранно-пожарной сигнализации : автореф. дис. ... канд. техн. наук: 05.13.11 / А. В. Засыпкин. – М. : МИП, 1993. – 20 с.
114. *Захаров В. А.* О преобразовании операторных процедур в логические программы / В. А. Захаров, С. И. Маневич // Программирование. – 1994. – № 6. – С. 23–38.
115. *Калечиц В. Е.* Эмулятор. Средства отладки в интерактивном режиме / В. Е. Калечиц, Н. И. Лободин, С. Д. Махортов. – Воронеж : ВЦНТИ, 1981. – ИЛ №23-81 НТД. – 4 с.
116. *Калечиц В. Е.* Отладчик. Операторы прерываний – структура и функции / В. Е. Калечиц, Н. И. Лободин, С. Д. Махортов. – Воронеж : ВЦНТИ, 1981. – ИЛ №24-81 НТД. – 4 с.
117. *Калечиц В. Е.* Отладчик. Операторы трассировки и статистики – структура и функции / В. Е. Калечиц, Н. И. Лободин, С. Д. Махортов. – Воронеж : ВЦНТИ, 1981. – ИЛ №25-81 НТД. – 4 с.
118. *Калечиц В. Е.* Эмуляция машинных команд / В. Е. Калечиц, Н. И. Лободин, С. Д. Махортов, В. П. Чертов, Н. П. Вахтина. – Воронеж : ВЦНТИ, 1981. – ИЛ №26-81 НТД. – 4 с.
119. *Калечиц В. Е.* Система интерактивной и пакетной отладки в режиме эмуляции / В. Е. Калечиц, Н. И. Лободин, С. Д. Махортов // Мате-

матическое моделирование и программное обеспечение САПР. – Горький, 1984. – С. 42–47.

120. *Касьянов В. Н.* Графы в программировании: обработка, визуализация и применение / В. Н. Касьянов, В. А. Евстигнеев. – СПб. : БХВ-Петербург, 2003. – 1104 с.
121. *Клини С.* Математическая логика / С. Клини. – М. : Мир, 1973. – 480 с.
122. Критически важные объекты и кибертерроризм. Часть 1. Системный подход к организации противодействия / О. О. Андреев и др. ; под ред. В. А. Васенина. – М. : МЦНМО, 2008. – 398 с.
123. Критически важные объекты и кибертерроризм. Часть 2. Аспекты программной реализации средств противодействия / О. О. Андреев и др. ; под ред. В. А. Васенина. – М. : МЦНМО, 2008. – 607 с.
124. *Кормен Т.* Алгоритмы: построение и анализ : пер. с англ. / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М. : МЦНМО, 2001. – 960 с.
125. *Крылов Г. Н.* Классические задачи прикладной электродинамики / Г. Н. Крылов. – СПб. : СПбГУ, 2005. – 260 с.
126. *Кузнецов С. О.* Быстрый алгоритм построения всех пересечений объектов из конечной полурешетки / С. О. Кузнецов // НТИ. Сер. 2. – 1993. – № 1. – С. 17–20.
127. *Кузнецов С. О.* Автоматическое обучение на основе анализа формальных понятий / С. О. Кузнецов // Автоматика и телемеханика. – 2001. – № 10. – С. 3–27.
128. *Кулик Б. А.* Система логического программирования на основе алгебры кортежей / Б. А. Кулик // Изв. РАН. Техн. кибернетика. – 1993. – № 3. – С. 226–239.
129. *Кулик Б. А.* Логика естественных рассуждений / Б. А. Кулик. – СПб. : Невский Диалект, 2001. – 128 с.
130. *Куратовский К.* Теория множеств : пер. с англ. / К. Куратовский, А. Мостовский. – М. : Мир, 1970. – 416 с.
131. *Кучеров Г. А.* Системы подстановок термов / Г. А. Кучеров. – Новосибирск, 1985. – 46 с.
132. *Левендорский С. З.* Вырождающиеся эллиптические уравнения и краевые задачи / С. З. Левендорский, Б. П. Панеях // Итоги науки и техн. Соврем. пробл. матем. Фундам. направления. – М. : ВИНИТИ, 1990. – Т. 63. – С. 131–200.
133. *Левин В. И.* Бесконечнозначная логика в задачах кибернетики / В. И. Левин. – М. : Радио и связь, 1982. – 176 с.
134. *Лингер Р.* Теория и практика структурного программирования : пер. с англ. / Р. Лингер, Х. Миллс, Б. Уитт. – М. : Мир, 1982. – 406 с.
135. *Лишнер Р.* Секреты Delphi 2 : пер. с англ. / Лишнер Р. – К. : Диасофт Лтд, 1996. – 800 с.
136. *Макконнелл С.* Совершенный код. Мастер-класс : пер. с англ. / С. Макконнелл. – М. : Русская редакция ; СПб. : Питер, 2005. – 896 с.
137. *Махортов С. Д.* Об одном классе псевдодифференциальных операторов неглавного типа / С. Д. Махортов // Операторные уравнения

в функциональных пространствах. – Воронеж : ВГУ, 1986. – С. 127–129.

138. *Махортов С. Д.* Разрешимость некоторых вырождающихся граничных задач для эллиптических уравнений / С. Д. Махортов // Неклассические уравнения математической физики. – Новосибирск, 1986. – С. 171–175.
139. *Махортов С. Д.* О задаче с косой производной при произвольном множестве вырождения / С. Д. Махортов // Тезисы XIII Всесоюзной школы по теории операторов в функциональных пространствах. – Куйбышев, 1988. – С. 128–129.
140. *Махортов С. Д.* Алгебра вырождающихся псевдодифференциальных операторов / С. Д. Махортов // Применение методов функционального анализа к неклассическим уравнениям математической физики. – Новосибирск, 1988. – С. 93–101.
141. *Махортов С. Д.* О разрешимости одной некоэрцитивной задачи для вырождающегося эллиптического уравнения / С. Д. Махортов // Тезисы XIV Всесоюзной школы по теории операторов в функциональных пространствах. – Новгород, 1989. – С. 71.
142. *Махортов С. Д.* О задаче с косой производной / С. Д. Махортов // Краевые задачи для неклассических уравнений математической физики. – Новосибирск, 1989. – С. 141–143.
143. *Махортов С. Д.* О фредгольмовости одного псевдодифференциального оператора неглавного типа / С. Д. Махортов // Тезисы XV Всесоюзной школы по теории операторов в функциональных пространствах. – Ульяновск, 1990. – С. 11.
144. *Махортов С. Д.* О задаче с косой производной / С. Д. Махортов // Материалы международного семинара «Дифференциальные уравнения и их приложения». – Самара, 27–30 июня 1995 г. – С. 63.
145. *Махортов С. Д.* О задаче с косой производной со специальным случаем вырождения / С. Д. Махортов // Материалы межвузовской научно-практической конференции «Актуальные проблемы борьбы с преступностью в современных условиях». – Воронеж : ВИ МВД, 2000. – С. 185.
146. *Махортов С. Д.* О технологии многоуровневой разработки программных систем / С. Д. Махортов // Вестник ВГУ. Серия Физика, математика. – Воронеж. – 2002. – № 1. – С. 159–162.
147. *Махортов С. Д.* Порождающие множества в продукционных системах / С. Д. Махортов // Вестник ВГУ. Серия Физика, математика. – Воронеж. – 2002. – № 2. – С. 69–76.
148. *Махортов С. Д.* О теоретико-множественном подходе к формализации логического вывода / С. Д. Махортов // Вестник факультета ПММ. – Воронеж : ВГУ. – 2003. – Вып. 4. – С. 178–185.
149. *Махортов С. Д.* Логические отношения на решетках / С. Д. Махортов // Вестник ВГУ. Серия Физика, математика. – Воронеж. – 2003. – № 2. – С. 203–209.

150. *Махортов С. Д.* О редукции логических отношений на решетках / С. Д. Махортов // Вестник факультета ПММ. – Воронеж : ВГУ, 2004. – Вып. 5. – С. 172–179.
151. *Махортов С. Д.* Логические уравнения на решетках / С. Д. Махортов // Вестник ВГУ. Серия Физика, математика. – Воронеж. – 2004, № 2. – С. 170–178.
152. *Махортов С. Д.* О разрешимости логических уравнений на решетках / С. Д. Махортов // Материалы Международной конференции «Образование, наука, производство и управление в XXI веке». – Ст. Оскол, 2004. – С. 308–311.
153. *Махортов С. Д.* Некоммутативные решетки и немонотонные логические отношения / С. Д. Махортов // Вестник ВГУ. Серия Физика, математика. – Воронеж. – 2006. – № 1. – С. 166–173.
154. *Махортов С. Д.* О сравнении возможностей императивного и логического программирования / С. Д. Махортов // Вестник ВГУ. Серия Системный анализ и информационные технологии. – Воронеж. – 2006. – № 1. – С. 78–86.
155. *Махортов С. Д.* Методы исследования и преобразования иерархий типов на основе логических структур / С. Д. Махортов // Вестник ВГУ. Серия Системный анализ и информационные технологии. – Воронеж. – 2006. – № 2. – С. 24–27.
156. *Махортов С. Д.* Алгебраическая интерпретация продукционной логики / С. Д. Махортов // Вестник факультета ПММ. – Воронеж : ВГУ. – 2006. – Вып. 6. – С. 86–98.
157. *Махортов С. Д.* Алгебры весовых и вырождающихся псевдодифференциальных операторов / С. Д. Махортов // Вестник ВГУ. Серия Физика, математика. – Воронеж. – 2007. – № 1. – С. 73–80.
158. *Махортов С. Д.* Теория LP-структур и возможности ее применения в интеллектуальных системах / С. Д. Махортов // Материалы 7-й международной конференции «Информатика: проблемы, методология, технологии». – Воронеж : ВГУ, 2007. – С. 269–272.
159. *Махортов С. Д.* Об алгебраической интерпретации продукционной логики нулевого порядка / С. Д. Махортов // Вестник ВГУ. Серия Системный анализ и информационные технологии. – Воронеж. – 2007. – № 1. – С. 56–63.
160. *Махортов С. Д.* Продукционная логика первого порядка и ее алгебраическая интерпретация / С. Д. Махортов // Системы управления и информационные технологии. – 2007. – № 3(29). – С. 21–26.
161. *Махортов С. Д.* Алгебраический подход к исследованию и оптимизации продукционных баз знаний / С. Д. Махортов // Сб. трудов международной школы-семинара «Современные проблемы механики и прикладной математики». – Воронеж : ВГУ, 2007. – С. 237–241.
162. *Махортов С. Д.* О приложениях LP-структур в теории программирования / С. Д. Махортов // Вестник ВГУ. Серия Системный анализ и информационные технологии. – Воронеж. – 2007. – № 2. – С. 40–49.

163. *Махортов С. Д.* О логической редукции условных систем переписывания термов / С. Д. Махортов // Сб. трудов XLIV Всероссийской конференции по проблемам математики, информатики, физики. – Москва : РУДН, 2008. – С. 50–51.
164. *Махортов С. Д.* LP-структуры и возможности их применения для эквивалентных преобразований условных систем переписывания термов / С. Д. Махортов // Обозрение прикладной и промышленной математики. – 2008. – Т. 15, вып. 3. – С. 504–505.
165. *Махортов С. Д.* Продукционно-логические отношения на полных решетках / С. Д. Махортов // Системы управления и информационные технологии. – 2008. – № 3(33). – С. 44–48.
166. *Махортов С. Д.* LP-структуры на полных решетках и возможности их применения в системах продукционного типа / С. Д. Махортов // Обозрение прикладной и промышленной математики. – 2008. – Т. 15, вып. 4. – С. 671–672.
167. *Махортов С. Д.* О редукции продукционно-логических отношений на полных решетках / С. Д. Махортов // Современные проблемы информатизации в анализе и синтезе технологических и программно-телекоммуникационных систем : сб. трудов / под ред. О. Я. Кравца. – Воронеж : Научная книга, 2009. – Вып. 14. – С. 322–325.
168. *Махортов С. Д.* Модель немонотонной продукционной логики для систем компьютерной алгебры / С. Д. Махортов // Вестник факультета ПММ. – Воронеж : ВГУ, 2009. – Вып. 7. – С. 127–141.
169. *Махортов С. Д.* Продукционно-логические уравнения на полных решетках / С. Д. Махортов // Обозрение прикладной и промышленной математики. – 2009. – Т. 16, вып. 2. – С. 368–369.
170. *Махортов С. Д.* Кластерно-релевантный обратный вывод на основе решения логических уравнений / С. Д. Махортов // Сб. трудов Международной конференции «Актуальные проблемы прикладной математики, информатики и механики». – Воронеж : ВГУ, 2009. – Ч. 2. – С. 37–41.
171. *Махортов С. Д.* LP-структуры на решетках типов и некоторые задачи рефакторинга / С. Д. Махортов // Программирование. – 2009. – Т. 35, № 4. – С. 5–14.
172. *Махортов С. Д.* Алгоритмы построения минимального нетранзитивного графа / С. Д. Махортов, Э. Ф. Мамедов // Черноземный альманах научных исследований. Серия Прикладная математика и информатика. – Воронеж : Альбион. – 2006. – № 1. – С. 27–36.
173. *Махортов С. Д.* LP-структуры и возможности их применения в некоторых задачах рефакторинга / С. Д. Махортов, В. А. Погореленко // Сб. трудов XLIV Всероссийской конференции по проблемам математики, информатики, физики. – Москва : РУДН, 2008. – С. 52–53.
174. *Махортов С. Д* Решение некоторых задач рефакторинга на основе алгебраических структур / С. Д. Махортов, В. А. Погореленко // Сб. трудов XII Международной научно-практической конференции-вы-

ставки «Актуальные проблемы информатики и информационных технологий». – Тамбов : ТГУ, 2008. – С. 147–149.

175. *Махортов С. Д.* Алгебраический подход к решению некоторых задач рефакторинга / С. Д. Махортов, В. А. Погореленко // Вестник ВГУ. Серия Системный анализ и информационные технологии. – Воронеж. – 2008. – № 2. – С. 28–36.

176. *Махортов С. Д.* Алгебраический подход к исследованию и оптимизации баз знаний продукционного типа / С. Д. Махортов, С. Л. Подвальный // Информационные технологии. – 2008. – № 8. – С. 55–60.

177. *Морозова И. С.* Программы понижения и повышения уровня структурного описания сложной системы / И. С. Морозова, С. Д. Махортов // Математическое обеспечение ЭВМ вузов. – Воронеж : ВГУ, 1980. – С. 115–120.

178. *Минский М.* Фреймы для представления знаний : пер. с англ. / М. Минский. – М. : Энергия, 1979. – 152 с.

179. *Нигиян С. А.* О семантике бестиповых функциональных программ / С. А. Нигиян, С. А. Аветисян // Программирование. – 2002. – № 3. – С. 5–14.

180. *Нильсон Н.* Принципы искусственного интеллекта : пер. с англ. / Н. Нильсон. – М. : Радио и связь, 1985. – 376 с.

181. *Орлик С. В.* Секреты Delphi на примерах / С. В. Орлик. – М. : Бином, 1996. – 316 с.

182. *Палмер С. Р.* Практическое руководство по функционально-ориентированной разработке ПО : пер. с англ. / С. Р. Палмер, Дж. М. Фелсинг. – М. : Вильямс, 2002. – 304 с.

183. *Питерсон Дж.* Теория сетей Петри и моделирование систем : пер. с англ. / Дж. Питерсон. – М. : Мир, 1984. – 264 с.

184. *Подгорный А. И.* О сравнении положения источника микроволнового излучения с положением токового слоя, вычисленным из данных предвспышечного состояния активной области при помощи МГД моделирования / А. И. Подгорный, И. М. Подгорный, Н. С. Мешалкина // Всероссийская конференция «Многоволновые исследования Солнца и современные проблемы солнечной активности», Нижний Архыз, 28 сент. – 2 окт. 2006. – П. Нижний Архыз, 2006. – С. 50.

185. *Подловченко Р. И.* Иерархия моделей программ / Р. И. Подловченко // Программирование. – 1981. – № 2. – С. 3–14.

186. *Попов Э. В.* Статические и динамические экспертные системы : учебное пособие / Э. В. Попов, И. Б. Фоминых, М. Д. Шапот. – М. : Финансы и статистика, 1996.

187. *Расёва Е.* Математика метаматематики : пер. с англ. / Е. Расёва, Р. Сикорский. – М. : Наука, 1972. – 591 с.

188. *Рыбаков В. В.* Базисы допустимых правил логик *S4* и *Int* / В. В. Рыбаков // Алгебра и логика. – 1985. – Т. 24, № 1. – С. 87–107.

189. *Рыбаков В. В.* Базисы допустимых правил модальной системы *Grz* и интуиционистской логики / В. В. Рыбаков // Матем. сборник. – 1985. – Т. 128 (170), № 3. – С. 321–338.

190. *Рыбаков В. В.* Независимые базисы для правил, допустимых в предтабличных логиках / В. В. Рыбаков // Алгебра и логика. – 2000. – Т. 39, № 2. – С. 206–226.
191. *Рыбина Г. В.* Верификация баз знаний в интегрированных экспертных системах / Г. В. Рыбина, В. В. Смирнов // Новости искусственного интеллекта. – 2005. – № 3. – С. 7–19.
192. *Сикорский Р.* Булевы алгебры : пер. с англ. / Р. Сикорский. – М. : Мир, 1969. – 375 с.
193. *Таненбаум Э.* Архитектура компьютера : пер. с англ. / Э. Таненбаум. – СПб. : Питер, 2002. – 704 с.
194. *Тейз А.* Логический подход к искусственному интеллекту: от классической логики к логическому программированию : пер. с франц. / А. Тейз, П. Грибомон и др. – М. : Мир, 1990. – 432 с.
195. *Тыугу Э. Х.* Концептуальное программирование / Э. Х. Тыугу. – М. : Наука, Гл. ред. физ.-мат. лит., 1984 – 256 с.
196. *Уотермен Д.* Руководство по экспертным системам : пер. с англ. / Д. Уотермен – М. : Мир, 1989. – 388 с.
197. *Уэно Х.* Представление и использование знаний : пер. с яп. / Х. Уэно, Т. Кояма, Т. Окамото, Б. Мацуби, М. Исидзука. – М. : Мир, 1989. – 220 с.
198. *Фаулер М.* Рефакторинг: улучшение существующего кода : пер. с англ. / М. Фаулер. – СПб. : Символ-Плюс, 2004. – 432 с.
199. *Фаулер М.* UML. Основы : пер. с англ. / М. Фаулер, К. Скотт. – СПб. : Символ-Плюс, 2002. – 192 с.
200. *Фуксман А. Л.* Технологические аспекты создания программных систем. / А. Л. Фуксман. – М. : Статистика, 1979. – 184 с.
201. *Фурман М. Е.* Применение метода быстрого умножения матриц к задаче нахождения транзитивного замыкания графа / М. Е. Фурман // Докл. АН СССР. – 1970. – Т. 194, № 3. – С. 530.
202. *Частиков А. П.* Разработка экспертных систем. Среда CLIPS / А. П. Частиков, Т. А. Гаврилова, Д. Л Белов. – СПб. : БХВ-Петербург, 2003. – 608 с.
203. *Чень Ч.* Математическая логика и автоматическое доказательство теорем : пер. с англ. / Ч. Чень, Р. Ли. – М. : Наука, Гл. ред. физ.-мат. лит., 1983. – 360 с.
204. *Чечкин А. В.* Математическая информатика / А. В. Чечкин. – М. : Наука, Гл. ред. физ.-мат. лит., 1991. – 416 с.
205. *Щупак Ю. А.* Win32 API. Эффективная разработка приложений / Ю. А. Щупак. – СПб. : Питер, 2007. – 572 с.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

ОГЛАВЛЕНИЕ

Научное издание

Махортов Сергей Дмитриевич

Математические основы искусственного интеллекта: теория LP-структур для построения и исследования моделей знаний продукционного типа

Издательство Московского центра
непрерывного математического образования
119002, Москва, Большой Власьевский пер., 11. Тел. (499) 241-74-83

Подписано в печать 16.11.2009 г. Формат $60\times90\,{}^1/_{16}$. Бумага офсетная.
Печать офсетная. Печ. л. 19. Тираж 500. Заказ .

Отпечатано с готовых диапозитивов в ППП «Типография „Наука“».
121099, Москва, Шубинский пер., д. 6.