РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ

На правах рукописи

Бениаминов Евгений Михайлович

АЛГЕБРАИЧЕСКИЕ МЕТОДЫ В ТЕОРИИ БАЗ ДАННЫХ И ПРЕДСТАВЛЕНИИ ЗНАНИЙ

Специальность — 05.13.17 —
Теоретические основы информатики

Диссертация на соискание ученой степени
доктора физико-математических наук

Москва – 1996

Содержание:

Введение

Актуальность темы Одним из последствий стремительного развития средств вычислительной техники явилось изменение технологии обработки информации во всех сферах человеческой деятельности. Современные ЭВМ позволяют накапливать огромные объемы информации и производить ее эффективную обработку. Одни и те же данные могут многократно использоваться в различных прикладных задачах. Централизованное хранение, накопление и ведение данных при одновременном доступе к данным большого числа пользователей позволило во много раз повысить эффективность функционирования информационных систем. Развитие новой информационной технологии, основанной на широком использовании компьютеров, современных средств передачи данных, баз данных и средств представления знаний потребовало решения ряда задач моделирования процессов организации, хранения, обработки данных и привело к постановкам новых математических задач.

Представление данных в виде алгебраических структур является естественным для многих специалистов в области программирования. Начиная с известной работы Е.Кодда [43] применение алгебраических методов распространилось на моделирование баз данных. Объектом изучения стали алгебраические системы с операциями над отношениями — реляционные алгебры и цилиндрические алгебры. Среди авторов, исследовавших эти объекты можно назвать H. Andreka, I. Nemeti, F. Banchilhon, Б. И. Плоткина и др. (см.[33, 34, 36, 28]). В настоящее время реляционные алгебры хорошо изучены в работах зарубежных и отечественных специалистов, включая работы автора [3, 4, 6].

Следующим шагом явилось изучение алгебраическими методами схем баз данных, расширение систем операций операциями над типами данных и над схемами. Это направление, с одной стороны, восходит к алгебраическим моделям абстрактных типов данных (R. M. Burstall [41], H. Ehrig [46], J. A. Goguen [50], S. N. Zilles [72] и др.), которые легли в основу языков программирования с развитым механизмами определения типов данных, а с другой стороны, привело к построению алгебраических моделей систем представления знаний, в которых существенна неполнота знаний, модульность организации знаний. Системы последнего типа условно можно назвать базами понятий.

Базы понятий — это новая складывающаяся область в программировании, связанная с использованием и построением больших библиотек типов объектов в объектно-ориентированных языках программирования, а также с использованием CASE–технологий и построением специализированных программных систем типа TOOLKIT, элементы которых предназначены для многоразового многоцелевого использования в различных приложениях.

Практика и опыт использования растущих библиотек подобного рода требует некоторой стандартизации в их организации и формах обращения к ним, которая была бы основана на ясной модели отдельных понятий и базы понятий в целом.

Состояние работ и исследований по базам понятий очень напоминает ситуацию с базами данных в конце 60-х начале 70-х годов, когда была осознана необходимость отделения процессов ведения данных от их использования, были созданы файловые системы, но не было ясной концепции баз данных. Для формулировки одной из таких концепций баз данных был успешно использован алгебраический язык в работах Е. Кодда.

Алгебраические методы, с учетом опыта их применения в теории баз данных и абстрактных типов данных, оказываются полезными и в моделировании баз понятий. Среди алгебраических средств в таких моделях особую роль играют понятия и методы теории категорий.

Один из важных вопросов приложения алгебры к информатике — это какого сорта проблемы в области баз данных и понятий могут решаться алгебраическими средствами, какие достижения в алгебре могут быть использованы для приложений в этой области, какие имеются ограничения по применению алгебраических методов в чистом виде.

Основное, что используется из алгебры в приложениях, — это алгебраический язык. Следующее — это алгебраический стиль мышления: использование понятий изоморфизма и гомоморфизма алгебраических систем, задание алгебраических систем с помощью образующих и соотношений. Дальнейшее использование алгебры приводит к использованию специфических понятий и результатов, введенных и исследованных в алгебре математиками. К ним относятся специфические операции и алгебраические системы, теоремы о разрешимости в алгебрах, специфические алгоритмы.

Результатами алгебраического исследования, помимо построения языка, на котором могут ставиться вопросы в области теории баз данных и понятий, являются теоремы классификации построенных алгебр и, в идеале, алгоритмы для распознавания тождеств, изоморфизмов, гомоморфизмов систем и решений уравнений, которые могут быть использованы в системах проектирования, алгоритмах оптимизации запросов, получения ответов на запросы.

Кроме того, как и в базах данных, так и в базах понятий существенный вопрос — это полнота использованной системы операций, который в некоторой постановке тоже может быть сведен к алгебраической задаче.

Цель работы. 1) Разработка алгебраических средств моделирования баз данных реляционного типа. 2) Полное описание и классификация реляционных алгебр. 3) Разработка алгебраических средств представления понятий и моделирования баз понятий. 4) Исследование математических объектов, используемых для представления понятий.

Методы работы. В исследовании использованы методы теории универсальных алгебр, математической логики, теории категорий, теории топосов.

Научная новизна. В диссертации предложены новые алгебраические объекты, служащие для моделирования баз данных и понятий.

Автором решены следующие задачи, определяющие научную новизну работы:

Дана формализация понятия реляционная алгебра.

Доказана теорема о представлении полных реляционных алгебр в виде прямой суммы неприводимых.

Дано полное описание неприводимых реляционных алгебр в духе теории Галуа.

Предложен подход к представлению понятий, основанный на методах теории категорий и средствах теории топосов.

Доказана теорема классификации булевых топосов конечного типа.

Введено понятие рефлексивного топоса, требуемого задачами представления понятий, и доказана теорема существования непротиворечивого рефлексивного топоса.

Рассмотрено приложение методов алгебраических систем с условными операциями и условными соотношениями к задачам представления понятий.

Теоретическая и практическая ценность Полученные в диссертации результаты позволяют глубоко исследовать математические модели, возникающие при проектировании информационных систем и автоматизированных систем представления понятий, дают возможность анализировать структуру получающихся при этом алгебраических объектов. Алгебраические и, в частности, теоретико категорные методы находят все более широкое применение в математической логике и информатике. Теорема о классификации неприводимых алгебр отражает связь между законами предметной области и принципами инвариантности (симметриями). Эта проблема давно интересовала ученых в различных областях исследований. Предложенные в диссертации подходы могут быть использованы при проектирования баз данных и баз понятий. Алгебраические методы, предложенные здесь для представления понятий, легли в основу системы представления понятий, макет которой на языке TurboProlog 2.0. был разработан в Международном центре по информатике и электронике в 1990г.

Апробация работы Результаты работы докладывались:

- на Всесоюзном симпозиуме "Методы логики в проблемах искусственного интеллекта и информатики" (Телави, 1978);

- на Первой Всесоюзной конференции "Банки данных" (Тбилиси, 1980);

- на IV Всесоюзной конференции "Применение методов математической логики" (Таллин, 1986);

- на Всесоюзной конференции "ДИАЛОГ — 87" (Тбилиси, 1987);

- на II Всесоюзной конференции "Искусственный интеллект — 90" (Минск, 1990);

- на Российской конференции по логическому программированию (RCLP'91) (Ленинград, 1991);

- на расширенном семинаре Московской и Киевской секции ACM SIGMOD (Москва, 1993);

- на международной конференции (Япония и страны СНГ) "Проектирование программного обеспечения, основанного на знаниях (JCKBSE'94)" (Переславль - Залесский, 1994);

- на международном симпозиуме "Перспективы развития систем баз данных и информационных систем (ADBIS'94, ADBIS'95)" (Москва, 1994, 1995);

- на научных семинарах в МГУ по алгебре (руководитель проф. А. В. Михалев (1980)) и по математической логике (руководитель проф. А. Л. Семенов (1986)).

Структура диссертации Диссертация состоит из введения, трех глав и списка литературы. Нумерация определений, утверждений и примеров в каждой главе проводится отдельно. При ссылках внутри главы на определения, утверждения или примеры других глав к соответствующему номеру добавляется номер главы, выраженный римскими цифрами. Например, Определение II.1.3 — третье определение в разделе 1 второй главы.

СОДЕРЖАНИЕ РАБОТЫ

Во введении дан обзор литературы, обсуждена актуальность темы, сформулирована цель и определены задачи исследования, описаны основные результаты.

В главе 1 определяются и исследуются реляционные алгебры, служащие для моделирования баз данных реляционного типа.

Реляционный подход к моделированию баз данных на логическом уровне, введенный Е.Ф. Коддом, в настоящее время широко освещен в литературе.

Суть реляционного подхода к моделированию данных состоит в том, что состояние описываемого мира (или рассматриваемой его части) можно представить в виде набора отношений (таблиц), а обработка данных сводится к операциям над отношениями. Поэтому основными понятиями подхода являются понятия схемы отношений, состояние схемы отношений и операций над отношениями.

Обработка информации с точки зрения реляционной модели заключается в получении новых отношений из отношений, хранящихся в ЭВМ, с помощью точно описанных процедур, которые называются операциями над отношениями. В различных информационных системах может использоваться разный класс операций над отношениями. В традиционном подходе Е.Ф. Кодда этот класс порождается (путем композиции) операциями проекции, соединения отношений, переименования атрибутов отношений и теоретико–множественными операциями объединения, пересечения и разности над отношениями с одинаковыми наборами атрибутов.

Операции над отношениями понимаются в рассматриваемом здесь алгебраическом подходе к моделям баз данных как алгебраические операции. Далее, дается точное аксиоматическое определение реляционной алгебры базы данных как многоосновной алгебры .

Грубо говоря, под реляционной алгеброй понимается множество, замкнутое относительно объявленных операций над отношениями. Так как эти операции имеют разные области определения, то алгебраическая система получается многоосновной. Соотношения между операциями, которые выполняются на любых отношениях, называются аксиомами реляционной алгебры.

В разных информационных системах, вообще говоря, могут быть выбраны разные классы операций над отношениями, но, как правило, эти классы замкнуты относительно композиции операций и содержат ряд простых необходимых операций так, что к исследованию возникающих при моделировании таких информационных систем алгебр могут быть с успехом применены методы современной алгебры.

База данных в реляционном подходе задается схемой и состоянием. Схема базы данных — это набор схем отношений $R^1(a_1^1, ..., a_{n_1}^1), ...,$ $R^k(a_1^k, ..., a_{nk}^k)$ (они называются базисными), состояние которых описывает предметную область, и набор соотношений (зависимостей, ограничений целостности базы данных) между этими отношениями вида:

$$\begin{array}{lcr} f_1(R^1, \ldots, R^k) & = & g_1(R^1, \ldots, R^k) \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ f_l(R^1, \ldots, R^k) & = & g_l(R^1, \ldots, R^k), \end{array}$$

где f_i и g_i — операции над отношениями, выраженные в виде композиций основных операций.

В частности, в виде подобных соотношений можно записать условие функциональной зависимости какого–либо атрибута от набора других атрибутов, вхождение одного отношения в другое как подмножества, пустоту пересечения каких–либо отношений и т.д. Соотношения, зафиксированные в схеме базы данных, отражают условия целостности базы данных, которые обязательно должны выполняться. Они отражают смысл представленных данных и могут используются в приложениях.

Состоянием S базы данных называется такое сопоставление схемам отношений с именами $R^1, ..., R^k$ настоящих отношений $S(R^1), ..., S(R^k)$ с теми же наборами атрибутов, что подстановка их в соотношения схемы делает соотношения верными.

Множество всех схем отношений, которые получаются из базисных схем посредством формального применения к ним операций над отношениями из данного класса, как легко заметить, замкнуто относительно операций, т. е. является реляционной алгеброй (алгеброй правильно построенных выражений — термов). Это множество можно также рассматривать как множество всех возможных запросов к базе данных.

Применяя к выделенным соотношениям схемы базы данных операции над отношениями и аксиомы реляционной алгебры, строится отношение эквивалентности на множестве запросов (термов). Множество классов эквивалентностей относительно этого отношения является реляционной алгеброй, которую естественно назвать реляционной алгеброй базы данных.

Пользователь базы данных имеет свое представление о данных, то есть свои базисные отношения и свои определяющие соотношения между ними и, следовательно, свою алгебру запросов и, соответственно свою реляционную алгебру пользователя. Если пользователь "подключен" к базе данных, то его запросы, выраженные через его базисные схемы отношений, интерпретируются запросами к базе данных так, что выполняются соотношения, объявленные пользователем. Эти условия определяют отображение алгебры пользователя в алгебру базы данных, совместимое с рассматриваемыми операциями, то есть определяют гомоморфизм реляционных алгебр. Наоборот, если отображение на множестве запросов определяет гомоморфизм алгебр, то это отображение допускает "подключение" пользователя к базе данных.

Две схемы базы данных имеют одинаковые информационные возможности, если определяемые ими реляционные алгебры изоморфны. Однако эффективность работы базы данных может существенно зависеть от ее схемы, т. е. выбора образующих и определяющих соотношений. Таким образом, после построения реляционной алгебры базы данных перед проектировщиком системы стоит задача оптимального выбора образующих этой алгебры.

Введение точных понятий позволило установить основные структурные теоремы о строении алгебр баз данных.

Обозначим через $\mathcal{K}$ систему Кодда операций над отношениями с бесконечными пересечениями и объединениями отношений.

Пусть $\mathcal{R}$ и $\mathcal{R}'$ — две $\mathcal{K}$-алгебры. Через $\mathcal{R}\oplus\mathcal{R}'$ обозначается $\mathcal{K}$-алгебра, которая называется прямой суммой $\mathcal{K}$-алгебр. Ее элементами с набором атрибутов A являются пары (R, R'), где $R \in \mathcal{R}(A)$, $R' \in \mathcal{R}'(A)$, а все операции выполняются по каждой координате отдельно.

О п р е д е л е н и е I.5.2. $\mathcal{K}$-алгебра $\mathcal{R}$ называется неприводимой, если она не представляется в виде суммы своих нетривиальных подалгебр, то есть $\mathcal{R} \neq \mathcal{R}_1 \oplus \mathcal{R}_2$, где $\mathcal{R}_1 \neq \emptyset$ и $\mathcal{R}_2 \neq \emptyset$.

Т е о р е м а I.5.15. Пусть $\mathcal{R}$ — произвольная $\mathcal{K}$-алгебра с конечным множеством образующих. Тогда $\mathcal{R}$ представляется в виде прямой суммы неприводимых $\mathcal{K}$-алгебр $\mathcal{R} = \bigoplus\limits_{s\in Spec\mathcal{R}} \mathcal{R}_s$ по некоторому множеству $Spec\mathcal{R}$, однозначно определяемому алгеброй $\mathcal{R}$. Если $j : \mathcal{R} \to \mathcal{R}'$ — гомоморфизм

$\mathcal{K}$-алгебр, то он индуцирует естественное частичное отображение

$$j : Spec\mathcal{R}' \to Spec\mathcal{R}$$

и множество гомоморфизмов-вложений

$$j_{s'} : \mathcal{R}_{j(s')} \to \mathcal{R}'_{s'},$$

для всех $s' \in Spec\mathcal{R}'$, для которых определено отображение j так, что $j(R) = j(\bigoplus\limits_{s \in Spec\mathcal{R}} R_s) = \bigoplus\limits_{s' \in Spec\mathcal{R}'} j_{s'}(R_{j(s')})$, где R — произвольный элемент $\mathcal{R}$.

По этой теореме, произвольная реляционная алгебра с конечным множеством образующих представляется в виде канонической прямой суммы неприводимых реляционных алгебр. Поэтому, чтобы завершить описание реляционных алгебр, нужно дать описание неприводимых реляционных алгебр.

Пусть $\mathcal{D}$ — $\mathcal{K}$-алгебра действительных отношений на множестве D, где D — дизъюнктное объединение областей значений всех типов атрибутов базы данных.

Т е о р е м а I.6.1. Если $\mathcal{R}$ — неприводимая $\mathcal{K}$-алгебра с конечным множеством образующих, тогда существует гомоморфизм - вложение $\mathcal{R} \to \mathcal{D}$.

Так как, согласно следствиям I.5.13 и I.5.14, каждая подалгебра алгебры реальных отношений $\mathcal{D}$ неприводима, то, отсюда следует, что для того, чтобы описать все неприводимые реляционные алгебры, нужно описать все реляционные подалгебры алгебры $\mathcal{D}$.

Пусть A и B — наборы атрибутов. Через $\mathcal{A}(A, B)$ обозначается множество всех отображений (согласований) из A в B, сохраняющих тип атрибутов. Тогда каждое отношение R с набором атрибутов A можно преставить в виде подмножества $R \subset \mathcal{A}(A, D)$, где каждая строка отношения представляется в виде соответствующего отображения набора атрибутов A в объединение доменов D, сохраняющего типы атрибутов.

Обозначим через $S(D)$ множество всех взаимно однозначных отображений D на себя, сохраняющих типы атрибутов. Очевидно, что $S(D)$ образует группу относительно композиции отображений. Будем говорить, что отношение R симметрично относительно $h \in S(H)$, если отношение R

не изменяется после замены значений атрибутов под действием отображения h, то есть $h \circ R = R$, где через $\circ$, обозначена операция композиции отображений из $R \subset \mathcal{A}(A, D)$ и отображения $h \in \mathcal{D}$.

Пусть $\{R_i\}_{i \in I}$ — произвольное множество отношений. Обозначим через H множество всех отображений $h \in S(D)$, относительно которых симметричны отношения R_i, при $i \in I$. Очевидно, что множество отображений H образуют подгруппу в $S(D)$, которая называется группой симметрии множества отношений $\{R_i\}_{i \in I}$.

Т е о р е м а I.6.7 Полные реляционные подалгебры $\mathcal{R} \subset \mathcal{D}$ алгебры отношений $\mathcal{D}$ над категорией наборов атрибутов $\bar{\mathcal{A}}$ (без ограничения их конечности) находятся во взаимно однозначном соответствии с подгруппами $H \subset S(D)$ группы всех перестановок элементов области значений атрибутов D, сохраняющих тип элементов. Подалгебре $\mathcal{R}$ сопоставляется подгруппа H всех перестановок, относительно которых симметрично каждое отношение подалгебры $\mathcal{R}$. Подгруппе H сопоставляется подалгебра R, состоящая из всех отношений симметричных относительно элементов группы H.

В случае конечного множества $\mathcal{D}$, эта теорема I.6.7. впервые была получена М. Краснером (см. [62]), а также независимо получена в работе F. Banchilhon (см.[36]).

Т е о р е м а I.6.8. Пусть $\mathcal{R} \subset \mathcal{D}$ и $\mathcal{R}' \subset \mathcal{D}'$ — полные подалгебры алгебр отношений с областями значений атрибутов, соответственно, D и D', и $j : \mathcal{R} \to \mathcal{R}'$ — изоморфизм реляционных алгебр. Тогда найдется взаимно однозначное отображение $g : D \to D'$ области D на область D', сохраняющее типы элементов, такое, что $H = g^{-1} \circ H' \circ g$, где H и H' — подгруппы, соответствующие подалгебрам $\mathcal{R}$ и $\mathcal{R}'$. В этом случае для любого отношения $R \in \mathcal{R}$ соответствующее ему отношение $j(R) \in \mathcal{R}'$ имеет вид: $j(R) = g \circ R$.

Для описания реляционных подалгебр алгебры отношений с конечными множествами атрибутов введем в множестве всех обратимых отображений $S(D)$ области значений атрибутов на себя следующую топологию:

для любого подмножества $K \subset S(D)$ обратимое отображение $\bar{h} \in S(D)$ принадлежит замыканию $\bar{K}$ тогда и только тогда, когда ограничение отображения $\bar{h}$ на любое конечное подмножество $e_L : L \subset D$ принадлежит множеству отображений $K \circ e_L$.

Т е о р е м а I.6.14. Реляционные подалгебры $\mathcal{R} \subset \mathcal{D}$ алгебры отношений с конечными наборами атрибутов и счетной или конечной областью значений атрибутов D находятся во взаимно однозначном соответствии с замкнутыми подгруппами $\bar{H} \subset S(D)$ группы $S(D)$ в топологии, определенной выше. Подгруппа $\bar{H}$ является группой симметрии отношений подалгебры $\mathcal{R}$.

Эти теоремы дают полную классификацию неприводимых полных реляционных алгебр и являются обобщением соответствующих результатов М. Краснера и Ф. Банчилона (F. Banchilhon) для алгебр конечных отношений.

Глава II посвящена алгебраическому подходу к моделированию понятий и их представлению для автоматизированной работы с ними. В качестве примеров представляемых понятий рассматриваются абстрактные типы данных, схемы баз данных и общие понятия предметных областей.

Известно, что алгебраические методы явились теоретической основой для определения, исследования и построения языков спецификаций абстрактных типов данных. В этой главе показывается, что алгебраические методы, дополненные средствами теории категорий являются удобным инструментом представления и более общих понятий.

Категорный подход к представлению знаний является естественным развитием реляционного подхода к базам данных, дополненного идеями и методами абстрактных типов данных в программировании, при этом используются достижения приложений математической теории категорий к математической логике.

В категорном подходе к представлению знаний предполагается, что представляемые знания прикладной области могут быть структурированы в виде системы понятий. С каждым понятием связывается имя понятия, определение и знания, вытекающие из определения. Теория категорий предоставляет математические средства для отражения семантики и логики понятий. Каждому представляемому понятию в этом подходе сопоставляется алгебраическая модель понятия (категория), в которой потенциально отражается полное знание о предмете понятия, вытекающее из определения этого понятия, а также строится конечная аппроксимация категории (фрагмент категории), в которой отражаются уже имеющиеся знания о представляемом понятии, отображенные в системе представления и непосредственно доступные пользователю.

Средства категорного подхода к представлению знаний условно можно разделить на три основные части:

– средства построения алгебраической модели (категории);

– средства построения конечной аппроксимации категории;

– язык представления знаний.

Все три составляющие подхода сильно связаны между собой:

– категория и ее аппроксимация задаются языковыми средствами;

– корректность новых языковых выражений определяется по уже построенной аппроксимации;

– категория, отражающая полные знания о представляемом понятии, — это идеальный объект, который является пределом своих конечных аппроксимаций.

В основе категорного подхода к представлению знаний также, как и в основе реляционного подхода к моделированию баз данных, лежит некоторый набор операций. Они называются категорными операциями, так как они были введены в математической теории категорий.

Категорные операции позволяют выразить представляемые понятия и определить требуемую их обработку. Особенностью категорных операций является возможность построения любых теоретико множественных конструкций из областей, не все элементы которых известны.

Единая категорная интерпретация понятий, основанная на общем наборе операций, позволяет достичь высокой степени интеграции понятий в систему знаний.

Глава III посвящена изучению алгебраических объектов, используемых для задач представления понятий.

Требование представления сложных понятий с использованием декартовых произведений, выделения элементов в областях с использованием финального объекта, выделением подобъектов и отношений для областей, в которых не все элементы известны, приводит к необходимости использования категорных конструкций топоса.

В приложениях топосов к моделированию баз данных особую роль играют булевы топосы, в которых реализуется классическая логика, и которые могут быть конструктивно заданы. В связи с этим рассматривается понятие алгебраического булевого топоса, заданного конечным множеством образующих и соотношений, в котором образующие-области (объекты) удовлетворяют условию конечности их мощности, выраженном на языке теории категорий. Такие топосы называются алгебраическими топосами конечного типа.

Топос называется непротиворечивым, если элементы его классификатора подобъектов (области истин) $true : I \to \Omega$ и $false : I \to \Omega$ не совпадают.

Топос называется полным или двузначным, если множество элементов его классификатора подобъектов Ω состоит лишь из двух элементов $true : I \to \Omega$ и $false : I \to \Omega$. Условие полноты топоса соответствует тому, что любое утверждение в полном топосе либо истинно, либо ложно, то есть полнота топоса соответствует полноте теории в исчислении предикатов.

Примерами полных булевых топосов являются категории $GFinset$, где G — произвольная конечная группа. Объектами категории $GFinset$ являются конечные множества M вместе с выделенным действием группы G на них перестановками элементов множества M. Морфизмами категории $GFinset$ являются все отображения G-множеств $f : M \to N$, сохраняющие действия группы G.

Оказывается, категории вида $GFinset$ исчерпывают примеры полных топосов конечного типа.

Т е о р е м а III.1.3. Если $\mathcal{C}$ — полный топос конечного типа, то существует такая конечная группа G, что категория $\mathcal{C}$ эквивалентна категории $GFinset$.

В частности, отсюда следует следующий результат:

Т е о р е м а III.1.19. Любой подтопос $\mathcal{C}$ топоса всех конечных множеств $Finset$, порожденный конечным множеством несущих множеств $D_1, ..., D_n$, и конечными множествами функций и отношений $f_1, ..., f_k$; $r_1, ..., r_m$ на множествах, построенных из несущих множеств операциями топоса, совпадает с подтопосом G-множеств, где $G = S(\mathcal{C})$ — группа симметрии этих образующих топоса $\mathcal{C}$.

Известно, что категории $G_1Finset$ и $G_2Finset$ конечных групп G_1 и G_2 эквивалентны тогда и только тогда когда эти группы изоморфны.

Эти результаты позволяет описать произвольные топосы конечного типа.

Т е о р е м а III.1.20. Пусть $\mathcal{C}$ — произвольный топос конечного типа. Тогда существует такой конечный набор конечных групп $G_1, ..., G_k$, что категория $\mathcal{C}$ эквивалентна конечному произведению топосов $G_iFinset$, для $i = 1, ..., k$, то есть $\mathcal{C} \approx G_1Finset \times ... \times G_kFinset$.

Если категорными средствами нужно отражать знания не только о внешнем мире, но и о самих средствах представления знаний, и о всей системе представления знаний в целом, то требуется расширение операций топоса, приводящее к понятию рефлексивного топоса, определяемого ниже.

Пусть в топосе $\mathcal{E}$ выделены два объекта NOb и $NMor$, которые называются область имен объектов и область имен морфизмов, соответственно. Множества элементов объектов NOb и $NMor$, то есть множества $\mathcal{E}(I, NOb)$ и $\mathcal{E}(I, NMor)$, где I — финальный объект категории $\mathcal{E}$, называются множеством имен объектов и множеством имен морфизмов, соответственно.

Предполагается, что между объектами NOb и $NMor$ заданы такие морфизмы, что они определяют внутреннюю категорию в $\mathcal{E}$ и индуцируют на множестве элементов объектов NOb и $NMor$ структуру категории, которая называется категорией имен и обозначается $NCat$.

О п р е д е л е н и е. Топос $\mathcal{E}$ вместе с внутренней категорией и категорией имен $NCat$ называется рефлексивным топосом, если существуют функторы $Name : \mathcal{E} \to NCat$ и $Denote : NCat \to \mathcal{E}$ такие, что их композиция $Denote \circ Name = 1\mathcal{E}$ является тождественным функтором на $\mathcal{E}$. Функтор $Name$ называется функтором именования, а функтор $Denote$ — денотатом.

Т е о р е м а III.2.1 Существует непротиворечивый рефлексивный топос.

В отличие от абстрактных типов данных, которые моделируются многоосновными алгебраическими системами, общие понятия и схемы баз данных, которые моделируются категориями с категорными операциями, требуют привлечения алгебраических средств, в которых операции действуют при выполнении некоторых условий. Примером такой операции является операция композиции отображений: композиция двух отображений f_1 и f_2 определена, если только область значений f_1 совпадает с областью определения f_2, то есть выполняется равенство $codom(f_1) = dom(f_2)$. Многие другие категорные операции также являются условными. Более того, условные операции требуются и для представления понятий, возникающих в предметных областях.

В диссертации исследуется понятие алгебры с условными операциями, порожденной словарем терминов и множеством условных операций и соотношений E. Частными случаями таких алгебр являются многоосновные алгебры, алгебраические топосы, рефлексивные топосы, алгебраические топосы с выделенными подобъектами и наследованием морфизмов. Рассмотрены понятие линейно излагаемой спецификации алгебры с условными операциями и понятие конечной аппроксимации таких алгебр.

Публикации Результаты диссертации опубликованы в 14 работах [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 37, 38, 39].

Благодарности. Я благодарю своих близких - мою жену и детей, за терпение, своих коллег и друзей за постоянное многолетнее сотрудничество, Российский Государственный Гуманитарный Университет за предоставленную возможность работать над диссертацией и Российский Фонд Фундаментальных Исследований за частичную материальную поддержку во время работы над диссертацией (грант номер 94-01-01479).

Литература

[1] Агафонов В. Н. *Спецификация программ: понятийные средства и их организация.* Новосибирск: Наука, Сиб.отд-ние, 1990.

[2] Агафонов В. Н. *Типы и абстракция данных в языках программирования.* В кн. Данные в языках программирования. М.:Мир,1982, с.263–327.

[3] Бениаминов Е. М. *Алгебраический подход к моделям баз данных реляционного типа.* В кн.:Семиотика и информатика, 1980, вып.14, с.44-80.

[4] Бениаминов Е. М. *Алгебраическая стуктура реляционных моделей баз данных.* НТИ, сер.2, 1980, N9, с. 23-25.

[5] Бениаминов Е. М., Березина Н. А. *Об алгебраическом подходе к описанию схем баз данных.*// В сб. Вопросы создания Автоматизированной системы НТИ по документам ГАФ СССР, Москва: ГАУ при Совете Министров СССР, ВНИИДАД, 1981, с.69-77.

[6] Бениаминов Е. М. *О роли симметрии в реляционных моделях баз данных и логических структурах.* НТИ, сер.2, 1984, N5, с.17-25.

[7] Бениаминов Е. М., Березина Н. А., Дунская М. В. *Разработка методов моделирования автоматизированной обработки, поиска и размещения данных в больших информационных системах.* СИФ ОЦНТИ, ВНИИДАД, депонированная рукопись, инв. N031–85,М.,1985.

[8] Бениаминов Е. М. *О некотором подходе к представлению знаний.* Тезисы докладов 4 Всесоюзн.конференции "Применение методов мат. логики" г. Таллин, 1986, с.34-36.

[9] Бениаминов Е. М. *Основания категорного подхода к представлению знаний. Категорные средства.* Изв. АН СССР Техн. кибернет.,N 2, 1988 , с.21–33.

[10] Бениаминов Е. М. *Рефлексивные топосы в категорном подходе к представлению знаний.* Тезисы докладов Всесоюзн. школы-семинара

"Семиотические аспекты формализации интеллектуальной деятельности" г. Боржоми, 1988, с.111-113.

[11] Бениаминов Е.М., Вайнтроб А. Ю. *Основные принципы диалогового языка для представления знаний средствами категорного подхода.* Материалы конференции ДИАЛОГ-87, г. Тбилиси, 1988, с.174-177.

[12] Бениаминов Е. М. *Алгебраические системы и типы данных.* //В кн.:Системное и теоретическое программирование, Ростов-н-Д: РГУ, 1988, с.83-92.

[13] Бениаминов Е. М. *Система представления и обработки понятий, основанная на алгебраическом (категорном) подходе.* Труды II Всесоюзной конференции "Искусственный интеллект- 90", т.2, 1990, с.8-11.

[14] Вигнер П. *Программирование на языке АДА.* М.:Мир, 1983.

[15] Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л. *Алгебра, логика, языки, программирование.* Киев:Наукова думка, 1974.

[16] Голдблатт Р. *Топосы. Категорный анализ логики.* М.:Мир, 1983.

[17] Гончаров С. С., Ершов Ю. Л., Свириденко Д. И. *Семантические основы логического программирования*// В сб. Концептуализация и смысл, под ред. Полякова И. В., Новосибирск:Наука, 1990, с.6-20.

[18] Джонстон П. Т. *Теория топосов.* М.:Наука, 1986.

[19] Жожикашвили А.В., Стефанюк В.Л. *Теория категорий в задачах представления знаний и обучения.* Изв. АН СССР. Техн. кибернет., N 2, 1986.

[20] Замулин А. В. *Системы программирования баз данных и знаний.* Новосибирск:Наука, 1990.

[21] Калиниченко Л. А. *Методы и средства интеграции неоднородных баз данных.* М.:Наука, 1983.

[22] Калиниченко Л. А., Рывкин В. М. *Машины баз данных и знаний.* М.:Наука, 1990.

[23] Капитонова Ю. В., Летичевский А. А. *Математическая теория проектирования вычислительных систем.* М.:Наука, 1988.

[24] Кондрашина Е. Ю., Литвинцева Л. В., Поспелов Д. А. *Представление знаний о времени и пространстве в интеллектуальных системах.* М.:Наука, 1989.

[25] Кучеров Г. А. *Системы подстановок термов.* Препринт 601, ВЦ АН СССР Сиб. отделение, Новосибирск, 1985.

[26] Мальцев А. И. *К общей теории алгебраических систем.* Мат. сборник, 1954, т.35, вып.1.

[27] Массер Д. *Спецификация абстрактных типов данных в системе AFFIRM.*// В сб.: Требования и спецификации в разработке программ, М.:Мир, 1984, с.199-222.

[28] Плоткин Б. И. *Универсальная алгебра, алгебраическая логика и базы данных.* М.:Наука, 1991.

[29] Поспелов Д. А. *Логико-лингвистические модели в системах управления.* М.:Энергоиздат, 1981.

[30] Свириденко Д. И. *Проект Сигма. Цели и задачи.* // В сб. Логические методы в программировании под ред. Ершова Ю. Л. (Вычислительные системы, вып. 133), РАН, Сиб. отд.-ние, Ин.-т математики, Новосибирск, 1990, с.68-94.

[31] Цаленко М. Ш. *Моделирование семантики в базах данных.* М.:Наука, 1989.

[32] Шенфилд Дж. *Математическая логика.* М.:Наука, 1975.

[33] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 1)* Comput. Lingust. Comput. Lang., 13, 1979, p. 152-282.

[34] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 2)* Comput. Lingust. Comput. Lang., 14, 1980, p. 43-65.

[35] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 3)* Lect. Notes in Comp. Sci., V.117, Springer-Verlag, Berlin, 1981, p. 281-290.

[36] Banchilhon F. *On the Compliteness of Query Language for Relational Data Bases.* Lect. Notes in Comp. Sci., V.64, Springer-Verlag, 1978, pp.76-98.

[37] Beniaminov E. M. *Concept Bases and Algebraic Modeling Methods.* Proceedings of the International Workshop on Advances in Databases and Information Systems (ADBIS'94), 1994, p.133-135.

[38] Beniaminov E. M. *A Categorical Approach to Knowledge Representation.* Japan-CIS Symposium on Knowledge Based Software Engineering'94 (JCKBSE'94), 1994, p.181-182.

[39] Beniaminov E. M. *Algebraic Invariants of Database Schemes.* Proceedings of the Second International Workshop on Advances in Databases and Information Systems (ADBIS'95), V.1, 1995, p.259-263.

[40] Breu R. *Algebraic Specification Techniques in OOP Environments.* Lect. Notes in Comp. Sci., V.562, Springer-Verlag, 1991.

[41] Burstall R. M., Goguen J. A. *The Semantics of Clear, a Specification Language* In: Proceedings, of the 1979 Copenhagen Winter Shcool on Abstrakt Software Specification, Springer-Verlad, Berlin, 1980, 292-332.

[42] Chandra A. K., Harel D. *Computable queries for relational data bases.* J. Comput. & Syst. Sci., V.21, N2, 1980, p.156-172.

[43] Codd E. F. *A relational model for large shared data banks.* Comm. of ACM 13, 6, 1970, p.377-387.

[44] Dershowitz N. *Orderings for term-rewriting systems.* Theor. Comput. Sci., 1982, V.17, 3, pp.279-302.

[45] Elmasri R., Weeldrever J., Hevner A. *The category concept: an extension to the entity-relatconship model* Data & Knowledge Engineering, V.1, N1,1985.

[46] Ehrig H., Mahr B. *Fundamentals of Algebraic Specification.* V.1, Springer-Verlag, 1985.

[47] Ehrig H., Mahr B. *Fundamentals of Algebraic Specification.* V.2, Springer-Verlag, 1990.

[48] *First International Symposium on Category Theory Applied to Computation and Control* Lect. Notes in Com. Sci, V.25, Springer - Verlad, Berlin, 1975.

[49] Georgescu I. *A Categorial approach to knowledge-based systems.* Computers and Artifical Intelligence, V.3, N2, 1984, pp.105-113.

[50] Goguen J. A., Thatchar J. W., Wagner E. G. *An inititial algebra approach to the specification, correctness and implementation of abstract data types.* In Current Trends in Programming Methodology IV: Data Structuring, Prentice Hall, 1978, pp.80-144.

[51] Goguen J. A. *Some design principles and theory for OBJ-O, a language to express and execute algebraic specifications of problems.* In: Lect. Notes Comput. Sci.,V.75, 1979, pp.425-473.

[52] Goguen J. A., Burstall R. M. *Some fundamental algebraic tools for the semantics of computation.* Theoretical Computer Science, V.31, N2, 3, 1984.

[53] Goguen J. A., Burstall R. M. *Introducing Institution.* Lect. Not. Comp.Sci., V.164, 1984.

[54] Goguen J. A., Meseguer J. *Equality, types, modules, and (why not?) generics for logic programming.* Conf. of Logical Prog., Uppsala, Sweden, 1984, 179-210.

[55] Goguen J. A., Meseguer J. *Eqlog: Equality, types, and generic modules for logic programming.* In Douglas SeGroot and Gary Lindstrom, eds. "Logic Programming: Functions, Relations and Equations", Prentice-Hall, Englewood Cliffs, N.J., 1986, pp.295-363.

[56] Goguen J. A., Meseguer J. *Order-sorted algebra solves the constructor selector, multiple representation and coercion problems.* Symposium on Logic in Comp. Sci., IEEE Comp. Society Press, 1987, pp.18-29.

[57] Goguen J. A., Burstall R. M. *Institutions: Abstract Model Theory for Specification and Programming.* Jorn. of ACM, V.39, 1, 1992, pp.95-146.

[58] Grothendieck A., Verdier J. L. *Théorie des Topos.* (SGA 4, exposés I-VI).–Second edition.–Berlin; Heidelberg; N. Y.:Springer, 1972.

[59] Huet G. *Confluent reductions: abstract properties and applications to term rewriting systems.* Journ. of ACM, 1980, v.27, N4, pp.797-821.

[60] Kaplan S. *Symplifying conditional term rewriting systems: unification, termination and confluence.* Jorn. Symbolic Computation 4(3), 1987, pp.295-334.

[61] Knuth D., Bendix P. *Simple word problems in universal algebras.* In: Conputational Problems in Abstract Algebra, Pergamon Press, Elmsford, New York, 1970, pp.263-297.

[62] Krasner M. I. *Generalization et analogues de la theorie de Galois.* Comptes Rendus de Congress de la Victorie de I'Ass. Franc. pour I'Avancem. Sci., 1945, pp. 54-58.

[63] Lawvere F. W. *Functorial semantics of algebraic theories.* Proc. Nat. Acad. Sci., 1963, V.50, N5, pp.869-872.

[64] Lawvere F. W. *Introduction* In: Toposes, Algebraic Geometry and Logic, Lect. Notes in Math., V.174, 1972.

[65] Maltsev A. *Algebraic Systems.* Springer-Verlag, 1973.

[66] Melton A., Schmidt D., and Strecher G. *Galois connections and computer science applications.* Lect. Notes in Comp. Sci., V. 240, Springer-Verlag, 1986.

[67] *Proceedings, Category Theory and Computer Programming,* Lect. Notes in Comp. Sci., V. 240, Springer-Verlag, 1986.

[68] *Proceedings, Category Theory and Computer Science,* Lect. Notes in Comp. Sci. V. 283, Springer-Verlag, 1987.

[69] Rydeheard D.F., Burstall R.M. *Computational category theory.* Prentice Hall, 1988.

[70] Tuijn C., Gyssens M., Paredaens J. *A Categorical Approach to Object-Oriented Data Modelling.* Proceedings of Third Workshop on Foundation of Models and Languages for Data and Objects, Aigen, 1991, pp.187–196.

[71] Rusinowitch M., Rémy J. L. (Eds.) *Conditional Term Rewriting Systems.* Lect. Notes in Comp. Sci., V. 656, Springer-Verlag, 1993.

[72] Zilles S.N. *Introduction to data algebras.* Lect. Notes Comput. Sci., V.86, Springer-Verlag, 1980.

Глава 1. Алгебраические методы теории баз данных

1 Некоторые определения

Пусть T — некоторое множество, которое будет называться множеством типов атрибутов. В общем случае это множество может быть и бесконечным.

О п р е д е л е н и е 1.1. Набором атрибутов называется конечное множество, типизированное элементами множества T, то есть конечное множество A, заданное вместе с отображением $t_A : A \to T$. Элементы множества A называются атрибутами. Типом атрибута $a \in A$ называется элемент $t_A(a)$ множества T. Согласованием наборов атрибутов $\varphi : B \to A$ называется отображение, сохраняющее типы атрибутов, то есть отображение конечных множеств, для которого коммутативна диаграмма:

$$\begin{array}{ccc} B & \xrightarrow{\varphi} & A \\ t_B \downarrow & & \downarrow t_A \\ T & = & T \end{array}$$

Категорию всех таких наборов атрибутов и их согласований обозначим через $\mathcal{A}_T$. Множество всех согласований из B в A будет обозначаться через $\mathcal{A}_T(B, A)$ или $\mathcal{A}(B, A)$, когда множество типов ясно из контекста. Категория наборов атрибутов $\mathcal{A}_T$ является полной подкатегорией категории $\bar{\mathcal{A}}_T$ всех множеств без ограничения конечности, типизированных множеством T, с отображениями, сохраняющими типы элементов. Множество морфизмов в этой расширенной категории будет также обозначаться через $\mathcal{A}(B, A)$ или через $\bar{\mathcal{A}}(B, A)$, если в тексте нужно подчеркнуть, что множества A или B могут быть бесконечными.

Сопоставим каждому элементу $t \in T$ множество D_t, которое называется областью значений атрибута типа t.

О п р е д е л е н и е 1.2. Отношением с набором атрибутов $A = \{a_1, ..., a_n\}$ называется подмножество $r(A) \subset D_{t_A(a1)} \times ... \times D_{t_A(an)}$ в декартовом произведении областей значений атрибутов набора A.

Множество всех отношений с набором атрибутов A обозначим через $\mathcal{D}(A)$. Для любого множества I (конечного или бесконечного) и любых отношений $r_i(A) \in \mathcal{D}(A)$, где $i \in I$, определены теоретико-множественные операции объединения $\bigcup\limits_{i \in I} r_i(A)$ и пересечения $\bigcap\limits_{i \in I} r_i(A)$. Для любых двух элементов $r_1(A), r_2(A) \in \mathcal{D}(A)$ определена теоретико множественная операция разности $r_1(A) \setminus r_2(A)$. Кроме того, в $\mathcal{D}(A)$ есть наименьшее пустое отношение $\emptyset$ и наибольшее полное отношение $E(A) = D_{t_A(a_1)} \times ... \times D_{t_A(an)}$. По отношению к этим операциям $\mathcal{D}(A)$ представляет собой полную булеву алгебру.

Определение 1.3. Операцией $f(A_1, ..., A_n; B)$ типа $(A_1, ..., A_n; B)$, где $A_1, ...A_n, B$ — наборы атрибутов, а f — имя операции, называется способ, сопоставляющий каждому набору отношений $r_1(A_1), ..., r_n(A_n)$ отношение $r(B)$, которое будет обозначаться через $f(r_1, ..., r_n)$.

Другими словами, операция $f(A_1, ..., A_n; B)$ задает отображение

$$f(A_1, ..., A_n; B) : \mathcal{D}(A_1) \times ... \times \mathcal{D}(A_n) \to \mathcal{D}(B).$$

Определение 1.4. Две операции $f_1(A_1, ..., A_n; B)$ и $f_2(A_1, ..., A_n; B)$ будем называть равными и писать $f_1 = f_2$, если они совпадают как отображения.

Замечание 1.1. Условие равенства двух операций, заданное определением 1.4, является наиболее сильным и не всегда может быть эффективно проверено.

В разных информационных системах класс всех применяемых операций над отношениями может быть разным. Однако используемые классы операций, как правило, обладают свойствами $\mathcal{K}1$ — $\mathcal{K}4$, которые перечислены ниже.

Обозначим через $\mathcal{K}$ некоторый класс операций над отношениями, а через $\mathcal{K}(A_1, ..., A_n; B)$ — множество всех операций этого класса вида $f(A_1, ..., A_n; B)$. Предполагается, что класс операций $\mathcal{K}$ обладает следующими свойствами.

$\mathcal{K}1$). Операция $\emptyset(A_1, ..., A_n; B)$, которая сопоставляет каждому набору отношений $r_1(A_1), ..., r_n(A_n)$ пустое отношение $\emptyset(B)$, принадлежит $\mathcal{K}(A_1, ..., A_n; B)$.

$\mathcal{K}2$). Операции $x_i(A_1, ..., A_n; A_i)$, которые каждому набору отношений $r(A_1), ..., r(A_n)$ сопоставляют отношения $r_i(A_i)$, то есть являются проекциями $\mathcal{D}(A_1) \times ... \times \mathcal{D}(A_n)$ на $\mathcal{D}(A_i)$, принадлежат множеству операций $\mathcal{K}(A_1, ..., A_n; B)$, для $i = 1, ..., n$.

Пусть $\{f_i(A_1, ..., A_n; B)\}_{i \in I}$ — произвольное множество операций. Определим операции $\bigcup\limits_{i \in I} f_i(A_1, ..., A_n; B)$; $\bigcap\limits_{i \in I} f_i(A_1, ..., A_n; B)$; $f_i(A_1, ..., A_n; B) \setminus f_j(A_1, ..., A_n; B)$ как операции, которые каждому набору отношений $r_1(A_1), ..., r_n(A_n)$ сопоставляют, соответственно, $\bigcup\limits_{i \in I} f_i(r_1, ..., r_n)$; $\bigcap\limits_{i \in I} f_i(r_1, ..., r_n)$; $f_i(r_1, ..., r_n) \setminus f_j(r_1, ..., r_n)$.

$\mathcal{K}3$). Для любого множества операций $f_i \in \mathcal{K}(A_1, ...A_n; B)$, $i \in I$, определенные выше операции $\bigcup\limits_{i \in I} f_i$, $\bigcap\limits_{i \in I} f_i$, и $f_i \setminus f_j$ принадлежат множеству $\mathcal{K}(A_1, ...A_n; B)$.

Будем говорить, что отображение

$$g : \mathcal{D}(A_1) \times ... \times \mathcal{D}(A_n) \to \mathcal{D}(B_1) \times ... \times \mathcal{D}(B_k)$$

принадлежит $\mathcal{K}(A_1, ...A_n; B_1, ..., B_k)$, если каждая композиция отображений

$$x_i(B_1, ..., B_k; B_i) \circ g : \mathcal{D}(A_1) \times ... \times \mathcal{D}(A_n) \to \mathcal{D}(B_i)$$

принадлежит $\mathcal{K}(A_1, ...A_n; B_i)$, для $i = 1, ..., k$.

$\mathcal{K}4$). Пусть отображение $g \in \mathcal{K}(A_1, ...A_n; B_1, ..., B_n)$ и операция $f \in \mathcal{K}(B_1, ...B_k; C)$, тогда существует операция $f \circ g \in \mathcal{K}(A_1, ...A_n; C)$, которая как отображение является композицией отображений f и g.

Основной пример класса операций $\mathcal{K}$, который мы будем подробнее изучать в этой работе, — это операции реляционной алгебры отношений [43].

Для того чтобы ввести эти операции, дадим несколько определений.

Обозначим через $D = \coprod\limits_{t \in T} D_t$ разъединенное объединение областей значений атрибутов всех типов. Определим тип $t_D : D \to T$ элементов множества D следующим образом: элемент $d \in D$ имеет тип $t = t_D(d)$, если $d \in D_t$.

Нетрудно видеть, что для любого набора атрибутов $A = \{a_1, ..., a_n\}$ множество $D_{t_A(a1)} \times ... \times D_{t_A(an)}$ находится в биективном соответствии с множеством $\mathcal{A}(A, D)$ всех отображений из A в D, сохраняющих тип элементов. Эта биекция задается следующим образом:

строке $(d_1, ..., d_n) \in D_{t_A(a1)} \times ... \times D_{t_A(an)}$ сопоставляется отображение $d : A \to D$ такое, что $d(a_k) = d_k$, для $k = 1, ..., n$;

наоборот, отображению $d : A \to D$ сопоставляется строка $(d(a_1), ..., d(a_n))$, где элемент $d(a_i) \in D_{t_A(ai)}$, так как отображение d сохраняет типы элементов.

Поэтому отношение $r(A)$ с набором атрибутов A можно рассматривать как подмножество $r(A) \subset \mathcal{A}(A, D)$ в множестве всех отображений из A в D, сохраняющих тип элементов.

Пусть $\varphi : B \to A$ — произвольное согласование наборов атрибутов. Морфизм φ индуцирует отображение $h_\varphi : \mathcal{A}(A, D) \to \mathcal{A}(B, D)$, которое морфизму $d : A \to D$ сопоставляет композицию морфизмов $h_\varphi(d) = d \circ \varphi$.

О п р е д е л е н и е 1.5. Если $r \subset \mathcal{A}(A, D)$ — произвольное отношение с набором атрибутов A, то через $\varphi^*(r)$ обозначается отношение с набором атрибутов B, которое является образом множества r при отображении h_φ. То есть

$$\varphi^*(r) = \{d \circ \varphi \in \mathcal{A}(B, D) : \ d \in r \subset \mathcal{A}(A, D)\}.$$

О п р е д е л е н и е 1.6. Если $r' \subset \mathcal{A}(B, D)$ — произвольное отношение с набором атрибутов B, то через $\varphi_*(r')$ обозначается отношение с набором атрибутом A, которое является прообразом множества r' относительно отображения h_φ. То есть

$$\varphi_*(r') = \{d \in \mathcal{A}(A, D) : \ d' = d \circ \varphi, d' \in r' \subset \mathcal{A}(B, D)\}.$$

Нетрудно видеть, что, когда $\varphi : B \to A$ — вложение, то операция $\varphi^* : \mathcal{D}(A) \to \mathcal{D}(B)$ является обычной операцией проекции отношений с набором атрибутов A на набор атрибутов B.

О п р е д е л е н и е 1.7. Если $\varphi_1 : A \to C$ и $\varphi_1 : B \to C$ — произвольные согласования наборов атрибутов и $r_1 \in \mathcal{D}(A)$, $r_2 \in \mathcal{D}(B)$, то обычная операция соединения отношений r_1 и r_2 относительно согласований φ_1 и φ_2, обозначаемая через $r_1 \underset{\varphi_1, \varphi_2}{\times} r_2$, выражается через определенные выше операции следующим образом:

$$r_1 \underset{\varphi_1, \varphi_2}{\times} r_2 = ((\varphi_1)_*(r_1)) \cap ((\varphi_2)_*(r_2)).$$

Если $\varphi : B \to A$ — взаимно однозначное согласование атрибутов, то операции φ^* и φ_* представляют собой обычные операции переименования атрибутов в отношениях в соответствии с отображением согласования атрибутов.

Все остальные операции Кодда получаются из уже определенных операций путем пополнения класса операций согласно свойствам $\mathcal{K}1$ — $\mathcal{K}4$.

З а м е ч а н и е 1.2. Введенные операции позволяют выразить не только обычные операции над отношениями, определенные Коддом, но и написать условие функциональной зависимости одного набора атрибутов от другого набора атрибутов для данного отношения в виде уравнения.

Приведем еще несколько примеров операций, которые не входят в класс операций, рассмотренный Коддом, но могут представлять интерес для приложений.

Постоянная операция $(const)r(B)$ типа $(A_1, ..., A_n; B)$ сопоставляет любому набору отношений $r_1(A_1), ..., r_n(A_n)$ одно и то же отношение $r(B)$.

Операция скобки $< >$ сопоставляет набору отношений $r_1(A_1), ..., r_n(A_n)$ одноэлементное отношение (строку) $< r_1(A_1), ..., r_n(A_n) >$. Эта операция позволяет рассматривать набор отношений как строку в отношении со сложными доменами атрибутов.

Следующие операции строятся с использованием произвольного отображения $f : D_{a1} \times ... \times D_{an} \to D_{b1} \times ... \times D_{bk}$, заданного на множествах возможных значений атрибутов набора $A = (a_1, ..., a_n)$ в множество возможных значений атрибутов набора $B = (b_1, ..., b_k)$. Это отображение индуцирует операцию f типа $(A; B)$, которая отношению $r(A) \in D_{a1} \times ... \times D_{an}$ сопоставляет его образ $f(r(A))$ в $D_{b1} \times ... \times D_{bk}$.

В общем случае в классе $\mathcal{K}$ могут быть операции, заданные произвольными алгоритмами [42].

З а м е ч а н и е 1.3. Нетрудно видеть, что условия $\mathcal{K}1$ — $\mathcal{K}4$ позволяют рассмотреть класс операций $\mathcal{K}$ как категорию, объектами которой являются наборы вида $(A_1, ..., A_n)$, а множества морфизмов — это множество операций $\mathcal{K}(A_1, ..., A_n; B_1, ..., B_k)$ типа $(A_1, ..., A_n; B_1, ..., B_k)$. Условие $\mathcal{K}4$ соответствует тому, что объекты $(A_1, ..., A_n)$ представляются в категории $\mathcal{K}$ в виде произведения $\prod_{i=1}^{n} A_i$.

З а м е ч а н и е 1.4. В данной работе предполагается, что все операции над отношениями из класса $\mathcal{K}$ имеют конечную арность (так операция f типа $(A_1, ..., A_n; B)$ имеет арность n), кроме теоретико множественных операций, которые имеют арность I произвольной мощности. Типы этих теоретико множественных операций имеют вид $(\underbrace{A, ..., A}_{I}; A)$. Существование бесконечных теоретико множественных операций по любому множеству является техническим требованием данной работы.

2 Определение $\mathcal{K}$ алгебр

Пусть задан класс операций над отношениями $\mathcal{K}$, обладающий свойствами $\mathcal{K}1$—$\mathcal{K}4$.

О п р е д е л е н и е 2.1. Алгеброй $\mathcal{R}$ над $\mathcal{K}$ или $\mathcal{K}$-алгеброй называется следующее сопоставление:

каждому набору атрибутов A сопоставляется булева алгебра $\mathcal{R}(A)$ с бесконечными объединениями, пересечениями и разностью между элементами;

каждой операции $f \in \mathcal{K}(A_1, ..., A_n; B)$ сопоставляется отображение

$$\bar{f} : \mathcal{R}(A_1) \times ... \times \mathcal{R}(A_n) \to \mathcal{R}(B).$$

Причем сопоставление удовлетворяет следующим условиям,

$\mathcal{R}1$). Нулевой (пустой) операции $\emptyset(A_1, ..., A_n; B)$ сопоставляется постоянное отображение

$$\bar{\emptyset} : \mathcal{R}(A_1) \times ... \times \mathcal{R}(A_n) \to \mathcal{R}(B).$$

в нулевой элемент булевой алгебры $\mathcal{R}(B)$.

$\mathcal{R}2$). Операциям $x_i(A_1, ..., A_n; A_i)$ сопоставляются проекции на i-ые координаты, то есть

$$\bar{x}_i = pr_i : \mathcal{R}(A_1) \times ... \times \mathcal{R}(A_n) \to \mathcal{R}(A_i).$$

$\mathcal{R}3$). Для любого множества операций

$$f_i(A_1, ..., A_n; B) \in \mathcal{K}(A_1, ..., A_n; B), \quad i \in I,$$

имеются равенства отображений

$$\overline{\bigcup_{i \in I} f_i(A_1, ..., A_n; B)} = \bigcup_{i \in I} \overline{f_i(A_1, ..., A_n; B)},$$

$$\overline{\bigcap_{i \in I} f_i(A_1, ..., A_n; B)} = \bigcap_{i \in I} \overline{f_i(A_1, ..., A_n; B)},$$

$$\overline{f_1(A_1, ..., A_n; B) \setminus f_2(A_1, ..., A_n; B)} = \overline{f_1(A_1, ..., A_n; B)} \setminus \overline{f_2(A_1, ..., A_n; B)}.$$

$\mathcal{R}4$). Для любой операции $g \in \mathcal{K}(A_1, ..., A_n; B_1, ..., B_k)$ через $\bar{g}$ обозначим отображение

$$\bar{g} : \mathcal{R}(A_1) \times ... \times \mathcal{R}(A_n) \to \mathcal{R}(B_1) \times ... \times \mathcal{R}(B_k),$$

заданное соотношениями

$$pr_i \circ \bar{g} = \bar{x}_i(B_1, ..., B_k; B_i) \circ \bar{g}, \text{ для } i = 1, ..., k.$$

Тогда, если $g \in \mathcal{K}(A_1, ..., A_n; B_1, ..., B_k)$ и $f \in \mathcal{K}(B_1, ..., B_k; C)$, то $\overline{f \circ g} = \bar{f} \circ \bar{g}$.

П р и м е р 2.1. Через $\mathcal{D}$ обозначим $\mathcal{K}$-алгебру, в которой набору атрибутов A сопоставляется булева алгебра всех отношений $\mathcal{D}(A)$ с набором атрибутов A, а операциям $f \in \mathcal{K}$ сопоставляются они сами.

Проверка условий $\mathcal{R}1$—$\mathcal{R}4$ очевидна и следует из определения и свойств операций класса $\mathcal{K}$.

В дальнейшем $\mathcal{K}$-алгебру $\mathcal{D}$ мы будем называть $\mathcal{K}$-алгеброй отношений или $\mathcal{K}$-алгеброй действительных отношений.

П р и м е р 2.2. Обозначим через $\mathcal{K}[C_1, ..., C_l]$ для фиксированного множества наборов атрибутов $\{C_1, ..., C_l\}$ следующий объект. Для каждого набора атрибутов A положим $\mathcal{K}[C_1, ..., C_l](A) = \mathcal{K}(C_1, ..., C_l; A)$. Для любой операции $f(A_1, ..., A_n; B)$ и любых $k_i \in \mathcal{K}[C_1, ..., C_l](A_i) = \mathcal{K}(C_1, ..., C_l; A_i)$, для $i = 1, ..., n$, положим

$$\bar{f}(k_1, ..., k_n) \stackrel{def}{=} f \circ k \in \mathcal{K}(C_1, ..., C_l; B) = \mathcal{K}[C_1, ..., C_l](B),$$

где k — операция из $\mathcal{K}(C_1, ..., C_l; A_1, ..., A_n)$ такая, что $x_i \circ k = k_i$ для $i = 1, ..., n$.

Покажем, что $\mathcal{K}(C_1, ..., C_l)$ является $\mathcal{K}$-алгеброй. Для этого нужно проверить выполнение условий $\mathcal{R}1$— $\mathcal{R}4$. Условия $\mathcal{R}1$, $\mathcal{R}2$, $\mathcal{R}3$ очевидным образом следуют из определений и свойств класса операций $\mathcal{K}$. Условие $\mathcal{R}4$ также следует из определений и свойства ассоциативности композиции операций.

Пусть $\mathcal{R}$ — произвольная $\mathcal{K}$-алгебра. В дальнейшем значение отображения

$$\bar{f} : \mathcal{R}(A_1) \times ... \times \mathcal{R}(A_n) \to \mathcal{R}(B),$$

сопоставленного операции $f \in \mathcal{K}(A_1, ..., A_n; B)$ на элементе $(R_1, ..., R_n) \in \mathcal{R}(A_1) \times ... \times \mathcal{R}(A_n)$, мы будем обозначать через $f(R_1, ..., R_n)$ без черточки сверху.

З а м е ч а н и е 2.1. В информационных системах реляционного типа в каждый момент времени хранится в том или ином виде некоторое множество отношений $r_1(A_1), ..., r_n(A_n)$, имена которых $R_1, ..., R_n$ известны пользователю системы. Пользователь формулирует запрос с помощью операций из принятого в данной информационной системе класса $\mathcal{K}$, применяемых к именам $R_1, ..., R_n$ в виде $f(R_1, ..., R_n)$, где $f \in \mathcal{K}(A_1, ..., A_n; B)$. Ответом на данный запрос в данный момент является отношение $r(B) = f(r_1, ..., r_n)$. Множество всех запросов $\{f(R_1, ..., R_n)\}$ с точностью до объявленных соотношений между ними образует $\mathcal{K}$-алгебру, которая называется $\mathcal{K}$-алгеброй данной информационной системы.

З а м е ч а н и е 2.2. На языке теории категорий определение $\mathcal{K}$-алгебры можно перефразировать следующим образом. Рассмотрим категорию операций над отношениями $\mathcal{K}$ (в том числе, и теоретико множественных операций). Алгеброй $\mathcal{R}$ над $\mathcal{K}$ называется функтор $\mathcal{R} : \mathcal{K} \to Set$ из категории $\mathcal{K}$ в категорию множеств Set, который прямое произведение объектов переводит в декартово произведение множеств, а постоянные операции (типа $\emptyset$) — в постоянные отображения множеств. Это определение соответствует определению Ловера [63] алгебры над произвольной категорией с произведениями. Изучаемые в данной работе алгебры можно рассматривать как многоосновные (о многоосновных алгебрах на русском языке см. [15, 28]), где каждое $\mathcal{R}(A)$ для набора атрибутов A является основанием многоосновной алгебры $\mathcal{R}$.

3 Гомоморфизмы и идеалы $\mathcal{K}$-алгебр

Пусть $\mathcal{R}_1$ и $\mathcal{R}_2$ — $\mathcal{K}$-алгебры. Тогда гомоморфизмы булевых алгебр $j_A : \mathcal{R}_1(A) \to \mathcal{R}_2(A)$, заданные для каждого набора атрибутов A, называются $\mathcal{K}$-гомоморфизмом алгебр $j : \mathcal{R}_1 \to \mathcal{R}_2$, если для любой операции $f \in \mathcal{K}(A_1, ..., A_n; B)$ и любых $R_i \in \mathcal{R}_1(A_i)$, для $i = 1, ..., n$, выполняется соотношение

$$j_B(f(R_1, ..., R_n)) = f(j_{A_1}(R_1), ..., j_{An}(R_n)).$$

В частности, $\mathcal{K}$-гомоморфизм переводит пустой элемент $\mathcal{R}_1(A)$ в пустой элемент $\mathcal{R}_2(A)$.

Подмножества $j_A^{-1}(\emptyset) \subset \mathcal{R}_1(A)$ для всех наборов атрибутов A называется ядром гомоморфизма j и обозначается через $Ker(j)$.

Семейство идеалов $\mathcal{J}(A) \subset \mathcal{R}(A)$ булевых алгебр $\mathcal{R}(A)$, для всех наборов атрибутов A, называется идеалом $\mathcal{K}$-алгебры и обозначается через $\mathcal{J}$, если для любой операции $f \in \mathcal{K}(A_1, ..., A_n; B)$ и любых элементов $R_i \in \mathcal{R}(A_i)$ и $J_i \in \mathcal{J}(A_i)$, где $i = 1, ..., n$, выполняется соотношение

$$f(R_1 + J_1, ...R_n + J_n) + f(R_1, ..., R_n) \in \mathcal{J}(B),$$

где знаком плюс $(+)$ обозначена операция симметрической разности между элементами в соответствующей булевой алгебры, то есть $a + b \stackrel{def}{=} (a \setminus b) \cup (b \setminus a)$.

У т в е р ж д е н и е 3.1. Пусть $\mathcal{J}$ — идеал $\mathcal{K}$-алгебры $\mathcal{R}$, и f — операция из $\mathcal{K}(A_1, ..., A_{n+k};\ B)$ такая, что

$$\begin{aligned} f(x_1(A_1, ..., A_n; A_1), ..., x_n(A_1, ..., A_n; A_n),\\ \emptyset(A_1, ..., A_n; A_{n+1}), ..., \emptyset(A_1, ..., A_n; A_{n+k})) =\\ = \emptyset(A_1, ..., A_n; B). \end{aligned}$$

Тогда для любых $R_i \in \mathcal{R}(A_i)$, $i = 1, ..., n$ и $J_l \in \mathcal{J}(A_{n+l})$, где $l = 1, ..., k$, элемент $f(R_1, ...R_n;\ J_1, ..., J_k)$ принадлежит $\mathcal{J}(B)$.

Д о к а з а т е л ь с т в о. Положим $R'_1 = R_1, ..., R'_n = R_n, R'_{n+1} = \emptyset, ..., R'_{n+k} = \emptyset$ и $J'_1 = \emptyset, ..., J'_n = \emptyset, J'_{n+1} = J_1, ..., J'_{n+k} = J_k$. Тогда, по определению идеала $\mathcal{K}$-алгебры, имеем:

$$f(R'_1 + J'_1, ..., R'_{n+k} + J'_{n+k}) + f(R'_1, ..., R'_{n+k}) \in \mathcal{J}(B).$$

С другой стороны, имеем:

$$
\begin{aligned}
& f(R_1' + J_1', ... R_{n+k}' + J_{n+k}') + f(R_1', ..., R_{n+k}') = \\
& = f(R_1, ..., R_n, J_1, ..., J_k) + f(R_1, ..., R_n, \emptyset, ..., \emptyset) = \\
& = f(R_1, ..., R_n, J_1, ..., J_k),
\end{aligned}
$$

так как, согласно условию, второе слагаемое $f(R_1, ..., R_n, \emptyset, ..., \emptyset) = \emptyset$.

У т в е р ж д е н и е 3.2. Пусть $\mathcal{J}$ — идеал в $\mathcal{K}$-алгебре $\mathcal{R}$. Тогда в объекте, сопоставляющем каждому набору атрибутов A булеву алгебру $\mathcal{R}(A)/\mathcal{J}(A)$, естественным образом вводится структура $\mathcal{K}$-алгебры так, что отображение $\mathcal{R} \to \mathcal{R}(A)/\mathcal{J}(A)$ является гомоморфизмом $\mathcal{K}$-алгебр.

Д о к а з а т е л ь с т в о. Действительно, пусть $[R_i] \in \mathcal{R}(A_i)/\mathcal{J}(A_i)$, $i = 1, ..., n$, — смежные классы, определенные элементами $R_i \in \mathcal{R}(A_i)$. Пусть $f(A_1, ..., A_n;\ B)$ — операция. Определим $f([R_1], ..., [R_n])$ — как смежный класс в $\mathcal{R}(A_i)/\mathcal{J}(A_i)$, определенный элементом $f(R_1, ..., R_n)$. Докажем корректность этого определения. Пусть $R_1 + J_1, ..., R_n + J_n$ — другой набор представителей тех же смежных классов. Тогда, по определению идеала,

$$f(R_1 + J_1, ..., R_n + J_n) + f(R_1, ..., R_n) \in \mathcal{J}(B),$$

то есть элементы $f(R_1 + J_1, ..., R_n + J_n)$ и $f(R_1, ..., R_n)$ определяют один и тот же элемент в $\mathcal{R}(B)/\mathcal{J}(B)$. Легко видеть, кроме того, что так определенное действие операций на $\mathcal{R}/\mathcal{J}$ удовлетворяет условиям $\mathcal{R}1$ — $\mathcal{R}4$, то есть $\mathcal{R}/\mathcal{J}$ является $\mathcal{K}$-алгеброй.

У т в е р ж д е н и е 3.3. Ядро $Ker(j)$ гомоморфизма $\mathcal{K}$-алгебр $j : \mathcal{R}_1 \to \mathcal{R}_2$ является идеалом.

Д о к а з а т е л ь с т в о. Пусть $R_i \in \mathcal{R}(A_i)$, $J_i \in j_{Ai}^{-1}(\emptyset)$, для $i = 1, ..., n$, и $f(A_1, ..., A_n;\ B)$ — произвольная операция класса $\mathcal{K}$. Рассмотрим

$$
\begin{aligned}
& j_B(f(R_1 + J_1, ..., R_n + J_n) + f(R_1, ..., R_n)) = \\
& = j_B(f(R_1 + J_1, ..., R_n + J_n)) + j_B(f(R_1, ..., R_n)) = \\
& = f(j_{A_1}(R_1 + J_1), ..., j_{An}(R_n + J_n)) + f(j_{A_1}(R_1), ..., j_{An}(R_n)) = \\
& = f(j_{A_1}(R_1), ..., j_{An}(R_n)) + f(j_{A_1}(R_1), ..., j_{An}(R_n)) = \emptyset,
\end{aligned}
$$

где все равенства были получены на основании свойства гомоморфизма $\mathcal{K}$-алгебр. Отсюда,

$$f(R_1 + J_1, ..., R_n + J_n) + f(R_1, ..., R_n) \in j_B^{-1}(\emptyset).$$

У т в е р ж д е н и е 3.4. Если $j : \mathcal{R}_1 \to \mathcal{R}_2$ — гомоморфизм и $\mathcal{J}_1 \subset \mathcal{R}_1$, $\mathcal{J}_2 \subset \mathcal{R}_2$ — идеалы $\mathcal{K}$-алгебр такие, что $j(\mathcal{J}_1) \subset \mathcal{J}_2$, то существует единственный гомоморфизм $j' : \mathcal{R}_1/\mathcal{J}_1 \to \mathcal{R}_2/\mathcal{J}_2$, делающий следующую диаграмму коммутативной:

$$\begin{array}{ccc} \mathcal{R}_1 & \longrightarrow & \mathcal{R}_1/\mathcal{J}_1 \\ j\downarrow & & \downarrow j' \\ \mathcal{R}_2 & \longrightarrow & \mathcal{R}_1/\mathcal{J}_2 \end{array}$$

Доказательство очевидно.

Подалгеброй $\mathcal{R}_1 \subset \mathcal{R}_2$ алгебры $\mathcal{R}_2$ называется семейство подалгебр $\mathcal{R}_1(A) \subset \mathcal{R}_2(A)$ булевых алгебр $\mathcal{R}_2(A)$ для всех наборов атрибутов A, замкнутое относительно операций класса $\mathcal{K}$.

Пусть $S = \{R_i\}_{i \in I}$ — множество элементов $\mathcal{K}$-алгебры $\mathcal{R}$, то есть $R_i \in \mathcal{R}(A_i)$, для $i \in I$. Через $\mathcal{R}\{S\}$ обозначим пересечение всех подалгебр в $\mathcal{R}$, содержащих S. Очевидно, что $\mathcal{R}\{S\}$ — подалгебра в $\mathcal{R}$. Эту подалгебру называют подалгеброй, порожденной множеством элементов S.

П р и м е р 3.1. Рассмотрим алгебру отношений $\mathcal{D}$ примера 2.1. Обозначим через $\mathcal{E}$ подалгебру алгебры $\mathcal{D}$, порожденную наибольшими отношениями — элементами вида $D_{a_1} \times ... \times D_{an} \in \mathcal{D}(A)$ для каждого набора атрибутов $A = \{a_1, ..., a_n\}$, где D_{ai} — множество возможных значений атрибута a_i. Алгебра $\mathcal{E}$ называется $\mathcal{K}$-алгеброй единиц.

Если $\mathcal{R} = \mathcal{R}\{S\}$, то говорят, что S является множеством образующих $\mathcal{K}$-алгебры $\mathcal{R}$. Если S — конечно, то говорят, что $\mathcal{R}$ конечно порождена.

Заметим, что, как и в случае колец, гомоморфизм $\mathcal{K}$-алгебр однозначно определяется своим действием на образующие. Именно, пусть $j : \mathcal{R}_1 \to \mathcal{R}_2$ — гомоморфизм $\mathcal{K}$-алгебр, и пусть S — множество образующих $\mathcal{K}$-алгебры $\mathcal{R}_1$. Тогда не существует гомоморфизма из $\mathcal{R}_1$ в $\mathcal{R}_2$, отличного от j и совпадающего с j на множестве S.

У т в е р ж д е н и е 3.5. Пусть $j : \mathcal{R}_1 \to \mathcal{R}_2$ — гомоморфизм $\mathcal{K}$-алгебр. Тогда образ $j(\mathcal{R}_1)$ — подалгебра в $\mathcal{R}_2$.

Доказательство очевидно.

Перейдем, теперь, к построению свободных $\mathcal{K}$-алгебр. Рассмотрим $\mathcal{K}$-алгебру $\mathcal{K}[C_1, ..., C_l]$ (пример 2.2.).

У т в е р ж д е н и е 3.6. Элементы $X_i \overset{def}{=} x_i(C_1, ..., C_l;\ C_i)$, где $x_i(C_1, ..., C_l;\ C_i) \in \mathcal{K}[C_1, ..., C_l](C_i)$, для $i = 1, ..., l$, являются образующими $\mathcal{K}$-алгебры $\mathcal{K}[C_1, ..., C_l]$.

Доказательство. Пусть $R \in \mathcal{K}[C_1, ..., C_l](A)$. Тогда, по определению (см. пример 2.2), $R = g$ для некоторой операции $g \in \mathcal{K}(C_1, ..., C_l;\ A)$. С другой стороны, имеем:

$$R = g = g(x_1(C_1, ..., C_l;\ C_1), ..., x_l(C_1, ..., C_l;\ C_l)) = g(X_1, ..., X_l).$$

По определению подалгебры, если подалгебра алгебры $\mathcal{K}[C_1, ..., C_l]$ содержит элементы $X_1, ..., X_n$, то она содержит элемент $R = g(X_1, ..., X_l)$. Так как R — произвольный элемент, то $X_1, ..., X_n$ — образующие алгебры $\mathcal{K}[C_1, ..., C_l]$.

Утверждение 3.7. Пусть $\mathcal{R}$ — произвольная $\mathcal{K}$-алгебра и $R_i \in \mathcal{R}(C_i)$, для $i = 1, ..., l$ — ее любые элементы. Тогда существует единственный $\mathcal{K}$-гомоморфизм

$$j : \mathcal{K}[C_1, ..., C_l] \to \mathcal{R}$$

такой, что $j(X_i) = R_i$, для $i = 1, ..., l$.

Доказательство. Для $g \in \mathcal{K}[C_1, ..., C_l](A) = \mathcal{K}(C_1, ..., C_l;\ A)$ положим $j(g) = g(R_1, ..., R_l)$. Покажем, что построенное отображение есть гомоморфизм булевых алгебр. Действительно, пусть $\{g_i(C_1, ..., C_l;\ A)\}_{i\in I}$ — множество элементов $\mathcal{K}[C_1, ..., C_l](A)$. Тогда

$$j(\bigcup_{i\in I} g_i) = (\bigcup_{i\in I} g_i)(R_1, ..., R_l) = \bigcup_{i\in I} g_i(R_1, ..., R_l).$$

Последнее равенство следует из того, что $\mathcal{R}$ — это $\mathcal{K}$-алгебра и поэтому удовлетворяет условию $\mathcal{R}3$. Аналогично доказывается, что пересечение переходит в пересечение и разность в разность. Пусть, теперь, $f \in \mathcal{K}(A_1, .., A_n;\ B)$ и g_i — произвольные элементы $\mathcal{K}(C_1, ..., C_l;\ A_i)$, для $i = 1, ..., n$. Тогда,

$$j(f(g_1, ..., g_n)) = f(g_1, ..., g_n)(R_1, ..., R_l) = f \circ g(R_1, ..., R_l).$$

Так как $\mathcal{R}$ — $\mathcal{K}$-алгебра и удовлетворяет свойству $\mathcal{R}4$, то последнее выражение представляется в виде:

$$f(g_1(R_1, ..., R_l), ..., g_n(R_1, ..., R_l)) = f(j(g_1), ..., j(g_n)).$$

Тем самым доказано, что j — гомоморфизм. Единственность такого гомоморфизма следует из того, что $X_1, ..., X_l$ — образующие $\mathcal{K}$-алгебры $\mathcal{K}[C_1, ..., C_l]$.

Алгебра $\mathcal{K}[C_1, ..., C_l]$ будет в дальнейшем называться свободной $\mathcal{K}$-алгеброй с образующими $X_1, ..., X_l$.

Утверждение 3.8. Пусть $R_i \in \mathcal{R}(A_i)$, для $i = 1, ..., n$, — образующие $\mathcal{K}$-алгебры $\mathcal{R}$. Тогда для любого элемента $R \in \mathcal{R}(B)$ найдется такая операция $f \in \mathcal{K}(A_1, ..., A_n;\ B)$, что $R = f(R_1, ..., R_n)$.

Это утверждение непосредственно следует из утверждений 3.7 и 3.6.

Пусть $M = \{J_i\}_{i \in I}$ — множество элементов $\mathcal{K}$-алгебры $\mathcal{R}$, то есть $J_i \in \mathcal{R}(A_i)$, для $i \in I$. Обозначим через $\mathcal{J}\{M\}$ пересечение всех идеалов $\mathcal{R}$, содержащих M. Очевидно, что $\mathcal{J}\{M\}$ — идеал $\mathcal{K}$-алгебры $\mathcal{R}$. Идеал $\mathcal{J}\{M\}$ называется идеалом, порожденным элементами множества M, а само множество M называется множеством образующих идеала $\mathcal{J}\{M\}$.

Утверждение 3.9. Пусть $\mathcal{R}$ — $\mathcal{K}$-алгебра, порожденная элементами $R_i \in \mathcal{R}(A_i)$, для $i = 1, ..., n$, и $\mathcal{J}$ — идеал, порожденный конечным множеством $M = \{J_s\}_{s \in I}$ элементов $J_s \in \mathcal{R}(A_{n+s})$, $s = 1, ..., k$. Тогда для каждого набора атрибутов B множество элементов вида $\mathcal{J}\{M\}(B)$ в точности совпадает с множеством $\mathcal{J}'(B)$ элементов вида $f(R_1, ..., R_n, J_1, ..., J_k)$, где $f \in \mathcal{K}(A_1, ..., A_{n+k};\ B)$ — произвольная операция, обладающая следующим свойством:

$$f(x_1(A_1, ...A_n;\ A_1), ..., x_n(A_1, ...A_n;\ A_n),$$
$$\emptyset(A_1, ..., A_n;\ A_{n+1}), ..., \emptyset(A_1, ...A_n;\ A_{n+k})) = \emptyset(A_1, ...A_n;\ B).$$

Доказательство. Согласно утверждению 3.1, элементы вида $f(R_1, ..., R_n, J_1, ..., J_k)$, где операция f удовлетворяет условию утверждения 3.9, принадлежат любому идеалу, содержащему элементы $J_1, ..., J_k$. Осталось показать, что $\mathcal{J}'$ — идеал. Для сокращения записи будем писать просто x_i вместо $x_i(A_1, ..., A_n;\ A_i)$ и $\emptyset$ вместо $\emptyset(A_1, ..., A_n;\ B)$, если тип операции известен из контекста. Покажем, сначала, что $\mathcal{J}'(B)$ — идеал булевой алгебры $\mathcal{R}'(B)$. Пусть $\{J'_i = f(R_1, ..., R_n, J_1, ..., J_k)\}_{i \in I}$ — произвольное подмножество в $\mathcal{J}'(B)$. Имеем

$\bigcup\limits_{i \in I} J'_i = \bigcup\limits_{i \in I} f_i(R_1, ..., R_n, J_1, ..., J_k) =$ (так как $\mathcal{R}$ — $\mathcal{K}$-алгебра и выполняется условие $\mathcal{R}3$) $= (\bigcup\limits_{i \in I} f_i)(R_1, ..., R_n, J_1, ..., J_k)$. С другой стороны,

$$(\bigcup_{i \in I} f_i)(x_1, ..., x_n, \emptyset, ..., \emptyset) = \bigcup_{i \in I} f_i(x_1, ..., x_n, \emptyset, ..., \emptyset) = \bigcup_{i \in I} \emptyset = \emptyset.$$

Отсюда, $\bigcup\limits_{i \in I} J'_i \in \mathcal{J}'(B)$. Пусть, теперь, $J' = f(R_1, ..., R_n, J_1, ..., J_k) \in$

$\mathcal{J}'(B)$ и R' — произвольный элемент $\mathcal{R}(B)$. Покажем, что $J' \bigcap R' \in \mathcal{J}'(B)$. Так как $R_1, ..., R_n$ — образующие $\mathcal{K}$-алгебры $\mathcal{R}$, то, согласно утверждению 3.8, существует операция $g \in \mathcal{K}(A_1, ...A_n\ B)$ такая, что $R' = g(R_1, ..., R_n)$. Тогда

$$J' \cap R' = f(R_1, ..., R_n, J_1, ..., J_k) \cap g(R_1, ..., R_n) =$$
$$= (f \cap g)(R_1, ..., R_n, J_1, ..., J_k).$$

С другой стороны, имеем:

$$(f \cap g)(x_1, ..., x_n, \emptyset, ..., \emptyset) = f(x_1, ..., x_n, \emptyset, ..., \emptyset) \bigcap g(x_1, ..., x_n) =$$
$$= \emptyset \bigcap g(x_1, ..., x_n) = \emptyset.$$

Отсюда, $J' \bigcap R' \in \mathcal{J}'(B)$. Итак, доказано, что $\mathcal{J}'(B)$ идеал булевой алгебры $\mathcal{R}(B)$.

Для завершения доказательства утверждения рассмотрим для произвольной операции $F \in \mathcal{K}(B_1, ..., B_s;\ C)$ и произвольных элементов $R'_j \in \mathcal{R}(B_j)$, $J'_j \in \mathcal{J}'(B_j)$ выражение $F(R'_1 + J'_1, ..., R'_s + J'_s) + F(R'_1, ..., R'_s)$. Так как $R_1, ..., R_n$ — образующие $\mathcal{K}$-алгебры $\mathcal{R}$, то, согласно утверждению 3.8, найдутся операции $g_j \in \mathcal{K}(A_1, ..., A_n;\ B_j)$ такие,что $R'_j = g_j(R_1, ..., R_n)$, для $j = 1, ..., s$. По определению $\mathcal{J}'$, элементы $J'_j \in \mathcal{J}'(B)$ имеют вид $J'_j = f_j(R_1, ..., R_n, J_1, ..., J_k)$, где операции $f_j \in \mathcal{K}(A_1, ..., A_{n+k}; B)$, для $j = 1, ..., s$, такие, что $f_j(x_1, ..., x_n, \emptyset, ..., \emptyset) = \emptyset$, для $j = 1, ..., s$.

Итак, имеем:

$$F(R'_1 + J'_1, ..., R'_s + J'_s) + F(R'_1, ..., R'_s) =$$
$$= F(g_1(R_1, ..., R_n) + f_1(R_1, ..., R_n, J_1, ..., J_k), ...$$
$$..., g_s(R_1, ..., R_n) + f_s(R_1, ..., R_n, J_1, ..., J_k)) +$$
$$+F(g_1(R_1, ..., R_n), ..., g_s(R_1, ..., R_n)).$$

Обозначим операцию от $R_1, ..., R_n, J_1, ..., J_k$, запись которой есть вся правая часть последнего равенства, через $\Phi \in \mathcal{K}(A_1, ..., A_{n+k};\ C)$.

С другой стороны, имеем:

$$\Phi(x_1, ..., x_n, \emptyset, ..., \emptyset) =$$
$$= F(g_1(x_1, ..., x_n) + f_1(x_1, ..., x_n, \emptyset, ..., \emptyset), ...,$$

$$
\begin{aligned}
& g_s(x_1, ..., x_n) + f_s(x_1, ..., x_n, \emptyset, ..., \emptyset)) + \\
& +F(g_1(x_1, ..., x_n), ..., g_s(x_1, ..., x_n)) = \\
= \ & F(g_1(x_1, ..., x_n) + \emptyset, ..., g_s(x_1, ..., x_n) + \emptyset) + \\
& +F(g_1(x_1, ..., x_n), ..., g_s(x_1, ..., x_n)) = \\
= \ & F(g_1(x_1, ..., x_n), ..., g_s(x_1, ..., x_n)) + \\
& +F(g_1(x_1, ..., x_n), ..., g_s(x_1, ..., x_n)) = \emptyset.
\end{aligned}
$$

Отсюда,

$$
F(R_1' + J_1', ..., R_s' + J_s') + F(R_1', ..., R's) = \Phi(R_1, ..., R_n, J_1, ..., J_k) \in \mathcal{J}'(C),
$$

то есть $\mathcal{J}'$ — идеал.

4 Категория $\mathcal{K}$-алгебр

В дальнейшем будем рассматривать только такие $\mathcal{K}$-алгебры, которые порождаются множествами образующих.

Легко видеть, что такие $\mathcal{K}$-алгебры и их гомоморфизмы образуют категорию. Множество гомоморфизмов из $\mathcal{K}$-алгебры $\mathcal{R}_1$ в $\mathcal{R}_2$ будет обозначаться через $\mathcal{K}(\mathcal{R}_1, \mathcal{R}_2)$.

Особо выделяется множество гомоморфизмов $\mathbf{S}(\mathcal{R}) = \mathcal{K}(\mathcal{R}, \mathcal{D})$ из $\mathcal{K}$-алгебры $\mathcal{R}$ в $\mathcal{K}$-алгебру действительных отношений $\mathcal{D}$ примера 2.1. Множество $\mathbf{S}(\mathcal{R})$ называется множеством или пространством возможных состояний алгебры $\mathcal{R}$. Это название связано с тем, что, если $\mathcal{R}$ — алгебра запросов некоторой информационной системы, и информационная система находится в некотором состоянии S, то по каждому запросу $R \in \mathcal{R}(A)$ выдается ответ $S(R) \in \mathcal{D}(A)$, причем отображение $S : \mathcal{R} \to \mathcal{D}$ является $\mathcal{K}$-гомоморфизмом.

Будем говорить, что $\mathcal{R}_1$ изоморфна $\mathcal{R}_2$, если существуют гомоморфизмы $j_1 : \mathcal{R}_1 \to \mathcal{R}_2$, $j_2 : \mathcal{R}_2 \to \mathcal{R}_1$ такие, что их композиции $j_2 \circ j_1 = id(\mathcal{R}_1)$, $j_1 \circ j_2 = id(\mathcal{R}_2)$ есть тождественные гомоморфизмы. Изоморфизм $\mathcal{K}$-алгебр $\mathcal{R}_1$ и $\mathcal{R}_2$ будем обозначать знаком $\approx$ и писать $\mathcal{R}_1 \approx \mathcal{R}_2$.

Тривиальным упражнением является доказательство изоморфизма множеств состояний $\mathbf{S}(\mathcal{R}_1) \approx \mathbf{S}(\mathcal{R}_2)$ для изоморфных $\mathcal{K}$-алгебр $\mathcal{R}_1 \approx \mathcal{R}_2$.

Пусть $\mathcal{R}$ и $\mathcal{R}'$ — две $\mathcal{K}$-алгебры. Через $\mathcal{R} \oplus \mathcal{R}'$ обозначим следующую $\mathcal{K}$-алгебру, которую будем называть суммой $\mathcal{K}$-алгебр.

Каждому набору атрибутов A сопоставляется булева алгебра $(\mathcal{R} \oplus \mathcal{R}')(A) = \mathcal{R}(A) \oplus \mathcal{R}'(A)$ — прямая сумма булевых алгебр $\mathcal{R}(A)$ и $\mathcal{R}'(A)$, то есть элементами $(\mathcal{R} \oplus \mathcal{R}')(A)$ являются пары (R, R'), где $R \in \mathcal{R}(A)$, $R' \in \mathcal{R}'(A)$, а все операции выполняются по каждой координате отдельно. Выполнение условий $\mathcal{R}1$ — $\mathcal{R}4$ при этом тривиально проверяется.

Из определения суммы $\mathcal{R} \oplus \mathcal{R}'$ следует, что отображения $j : \mathcal{R} \oplus \mathcal{R}' \to \mathcal{R}$ и $j' : \mathcal{R} \oplus \mathcal{R}' \to \mathcal{R}'$, сопоставляющие элементу (R, R') элементы $j(R, R') = R$ и $j'(R, R') = R'$, являются гомоморфизмами $\mathcal{K}$-алгебр.

У т в е р ж д е н и е 4.1. Пусть $i_1 : \mathcal{R}'' \to \mathcal{R}$ и $i_2 : \mathcal{R}'' \to \mathcal{R}'$ — произвольные гомоморфизмы $\mathcal{K}$-алгебр. Тогда существует единственный гомоморфизм $(i_1 \oplus i_2) : \mathcal{R}'' \to \mathcal{R} \oplus \mathcal{R}'$ такой, что выполняются следующие равенства для композиций гомоморфизмов $j \circ (i_1 \oplus i_2) = i_1$ и $j' \circ (i_1 \oplus i_2) = i_2$.

Д о к а з а т е л ь с т в о. Пусть $R'' \in \mathcal{R}''(A)$. Так как, согласно условию утверждения $j \circ (i_1 \oplus i_2)(R'') = i_1(R'')$ и $j' \circ (i_1 \oplus i_2)(R'') = i_2(R'')$, то из определения гомоморфизмов j и j' следует, что $(i_1 \oplus i_2)(R'') = (i_1(R''),\ i_2(R''))$. Осталось проверить, что так определенное отображение является гомоморфизмом, а это очевидно.

Перейдем к определению тензорного произведения $\mathcal{K}$-алгебр. Пусть $\mathcal{R}$ и $\mathcal{R}$ — алгебры с конечными множествами образующих S и S', соответственно. Рассмотрим свободную алгебру $\mathcal{K}[S \cup S']$ с множеством образующих $S \cup S'$ (см. пример 2.2). Согласно утверждению 3.7, чтобы задать гомоморфизм из свободной алгебры, достаточно только задать отображение на образующих.

Положим $\bar{p} : \mathcal{K}[S] \to \mathcal{K}[S \cup S']$ и $\bar{p}' : \mathcal{K}[S'] \to \mathcal{K}[S \cup S']$ — гомоморфизмы, заданные равенствами $\bar{p}(s) = s$, для $s \in S$ и $\bar{p}'(s') = s'$, для $s' \in S'$.

Пусть $j : \mathcal{K}[S] \to \mathcal{R}$ и $j : \mathcal{K}[S'] \to \mathcal{R}'$ — гомоморфизмы, тождественные на образующих.

Обозначим через $\mathcal{J} = \mathcal{J}\{\bar{p}(Ker(j)) \cup \bar{p}'(Ker(j'))\}$ идеал в $\mathcal{K}[S \cup S']$, порожденный элементами вида $\bar{p}(R)$ и $\bar{p}'(R')$, где элементы R и R' принадлежат ядрам гомоморфизмов j и j', то есть $j(R) = \emptyset$ и $j'(R') = \emptyset$.

Тензорным произведением $\mathcal{K}$-алгебр $\mathcal{R}$ и $\mathcal{R}'$ с множествами образующих S и S' называется $\mathcal{K}$-алгебра $\mathcal{R} \otimes \mathcal{R}' = \mathcal{K}[S \cup S']/\mathcal{J}$.

Согласно утверждению 3.4, имеем коммутативную диаграмму $\mathcal{K}$-алгебр и идеалов:

$$\begin{array}{ccccc}
Ker(j) & \to & \mathcal{J} = \mathcal{J}\{\bar{p}(Ker(j)) \cup \bar{p}'(Ker(j'))\} & \leftarrow & Ker(j') \\
\downarrow & & \downarrow & & \downarrow \\
\mathcal{K}[S] & \xrightarrow{\bar{p}} & \mathcal{K}[S \cup S'] & \xleftarrow{\bar{p}'} & \mathcal{K}[S'] \\
j \downarrow & & \downarrow j \otimes j' & & \downarrow j' \\
\mathcal{R} & \xrightarrow{p} & \mathcal{R} \otimes \mathcal{R}' & \xleftarrow{p'} & \mathcal{R}'
\end{array}$$

У т в е р ж д е н и е 4.2. Пусть $i_1 : \mathcal{R} \to \mathcal{R}''$ и $i_2 : \mathcal{R}' \to \mathcal{R}''$ — произвольные гомоморфизмы $\mathcal{K}$-алгебр. Тогда существует единственный гомоморфизм $\mathcal{K}$-алгебр $i_1 \otimes i_2 : \mathcal{R} \otimes \mathcal{R}' \to \mathcal{R}''$ такой, что композиции гомоморфизмов $(i_1 \otimes i_2) \circ p = i_1$ и $(i_1 \otimes i_2) \circ p' = i_2$.

Д о к а з а т е л ь с т в о. Если гомоморфизм $i_1 \otimes i_2$ существует, то сквозной гомоморфизм

$$\overline{i_1 \otimes i_2} : \mathcal{K}[S \cup S'] \overset{j \otimes j'}{\to} \mathcal{R} \otimes \mathcal{R}' \overset{i_1 \otimes i_2}{\to} \mathcal{R}''$$

на образующих, согласно условию утверждения и коммутативности приведенной выше диаграммы, имеет вид $\overline{i_1 \otimes i_2}(s) = i_1(s)$, для $s \in S$ и

$\overline{i_1 \otimes i_2}(s') = i_2(s')$, для $s' \in S'$. Отсюда следует единственность гомоморфизма $i_1 \otimes i_2$. Для доказательства существования рассмотрим гомоморфизм $\overline{i_1 \otimes i_2} : \mathcal{K}[S \cup S'] \to \mathcal{R}''$, определенный на образующих соотношениями, выписанными выше. Так как $i_1 \circ j = \overline{i_1 \otimes i_2} \circ \bar{p}$, что легко проверяется на образующих алгебры $\mathcal{K}[S]$, то $\bar{p}(Ker(j)) \subset Ker(\overline{i_1 \otimes i_2})$. Действительно, пусть $R \in \mathcal{K}[S]$ и $j(R) = \emptyset$. Тогда

$$(\overline{i_1 \otimes i_2})(\bar{p}(R)) = i_1 \circ j(R) = \emptyset.$$

Аналогично показывается, что $\bar{p}'(Ker(j')) \subset Ker(\overline{i_1 \otimes i_2})$. Отсюда, $\mathcal{J} \subset Ker(\overline{i_1 \otimes i_2})$, так как $\mathcal{J}$ – наименьший идеал, содержащий $\bar{p}(Ker(j))$ и $\bar{p}'(Ker(j'))$. Из утверждения 3.4 следует существование гомоморфизма

$$i_1 \otimes i_2 : \mathcal{R} \otimes \mathcal{R}' = \mathcal{K}[S \cup S']/\mathcal{J} \to \mathcal{K}[S \cup S']/Ker(\overline{i_1 \otimes i_2}) = \mathcal{R}''.$$

Проверка условий утверждения 3.4 тривиальна, так как эти условия выполняются на образующих.

С л е д с т в и е 4.3. Определение тензорного произведения не зависит от выбора образующих в $\mathcal{K}$-алгебрах.

С л е д с т в и е 4.4. Пространство состояний $\mathbf{S}(\mathcal{R} \otimes \mathcal{R}') = \mathbf{S}(\mathcal{R}) \times \mathbf{S}(\mathcal{R}')$.

Пусть $\mathcal{C}$ — некоторая $\mathcal{K}$-алгебра, которую мы будем называть алгеброй констант.

$\mathcal{C}$-алгеброй $\mathcal{R}$ называется $\mathcal{K}$-алгебра с выделенным гомоморфизмом $\mathcal{K}$-алгебр $i_{\mathcal{R}} : \mathcal{C} \to \mathcal{R}$.

Гомоморфизмом $\mathcal{C}$-алгебр $j : \mathcal{R}_1 \to \mathcal{R}_2$ называется гомоморфизм $\mathcal{K}$-алгебр, обладающий свойством $i_{\mathcal{R}_2} = j \circ i_{\mathcal{R}_1}$.

Идеал $\mathcal{J} \subset \mathcal{R}$ $\mathcal{C}$-алгебры $\mathcal{R}$ называется $\mathcal{C}$-идеалом, если выполняется равенство $\mathcal{J} \cap i_{\mathcal{R}}(\mathcal{C}) = \emptyset$.

Очевидно, что $\mathcal{C}$-алгебры и их гомоморфизмы образуют категорию. Множество гомоморфизмов из $\mathcal{C}$-алгебры $\mathcal{R}_1$ в $\mathcal{C}$-алгебру $\mathcal{R}_2$ будем обозначать через $\mathcal{C}(\mathcal{R}_1, \mathcal{R}_2)$.

Если в качестве алгебры констант $\mathcal{C}$ берется $\mathcal{K}$-алгебра единиц $\mathcal{E}$, введенная в примере 3.1 и для любого набора атрибутов A элемент $i_{\mathcal{R}}(E(A))$ является единицей в булевой алгебре $\mathcal{R}(A)$, то такую алгебру $\mathcal{R}$ называют $\mathcal{K}$-алгеброй с единицей.

Пусть, теперь, $\mathcal{K}'$ — другой класс операций над отношениями, удовлетворяющий условиям $\mathcal{K}1$—$\mathcal{K}4$, и $\mathcal{K} \subset \mathcal{K}'$. Такой класс операций $\mathcal{K}'$ называется расширением класса $\mathcal{K}$. Пусть $\mathcal{R}$ — произвольная $\mathcal{K}$-алгебра с конечным множеством образующих $S = \{R_1, ..., R_n\}$ и конечным числом соотношений $f_1(R_1, ..., R_n) = \emptyset, ..., f_k(R_1, ..., R_n) = \emptyset$, то есть

$$\mathcal{R} = \mathcal{K}[S]/\mathcal{J}\{f_1, ..., f_k\},$$

где $\mathcal{K}[S]$ — свободная $\mathcal{K}$-алгебра S, а $\mathcal{J}\{f_1, ..., f_k\}$ — идеал, порожденный операциями $f_1, ..., f_k$, рассматриваемыми как элементы алгебры $\mathcal{K}[S]$.

Каждой такой $\mathcal{K}$-алгебре $\mathcal{R}$ сопоставим $\mathcal{K}'$-алгебру $\mathcal{R}'$, по определению равную

$$\mathcal{R}' = \mathcal{K}'[S]/\mathcal{J}'\{f_1, ..., f_k\},$$

где $\mathcal{J}'\{f_1, ..., f_k\}$ — идеал, порожденный операциями $f_1, ..., f_k$, которые рассматриваются как элементы свободной $\mathcal{K}'$-алгебры $\mathcal{K}'[S]$. Стандартным образом показывается, что сопоставление $\mathcal{R} \mapsto \mathcal{R}'$ не зависит от выбора образующих в $\mathcal{R}$ и является ковариантным функтором. Этот функтор называется функтором замены класса операций.

З а м е ч а н и е 4.1. Так как $\mathcal{K}$-алгебры имеют идеалы, то можно определить спектр $Spec(\mathcal{R})$ $\mathcal{K}$-алгебры $\mathcal{R}$ как множество максимальных идеалов и ввести на множестве $Spec(\mathcal{R})$ топологию Зариского, где замкнутыми подмножествами в $Spec\ \mathcal{R}$ являются подмножества вида $N(\mathcal{J})$ =(множество максимальных идеалов, содержащих идеал $\mathcal{J}$). Далее можно было бы попытаться распространить методы и результаты алгебраической геометрии на случай $\mathcal{K}$-алгебр. Однако в данной работе в таком общем случае мы этим заниматься не будем.

5 Теорема о разложении $\mathcal{K}$-алгебр

Начиная с этого момента, под $\mathcal{K}$ понимается класс операций реляционной алгебры (пример раздела 1). Операции над отношениями в этом классе порождаются согласованиями между наборами атрибутов.

В различных системах могут использоваться не все возможные согласования (см. определение 1.1) между наборами атрибутов, а только некоторые, совокупность которых обозначается через $\mathcal{A}$. Множество согласований между наборами атрибутов A и B из класса $\mathcal{A}$ обозначается, как обычно, через $\mathcal{A}(A, B)$.

Будем предполагать, что класс $\mathcal{A}$ обладает следующими свойствами:

$\mathcal{A}1$). Если набор атрибутов A принадлежит классу $\mathcal{A}$, то тождественное согласование $id(A) : A \to A$ принадлежит $\mathcal{A}(A, A)$.

$\mathcal{A}2$). Если $\varphi_1 \in \mathcal{A}(B, A)$, $\varphi_2 \in \mathcal{A}(C, B)$, то их композиция $\varphi_1 \circ \varphi_2 \in \mathcal{A}(C, A)$.

Другими словами, свойства $\mathcal{A}1$, $\mathcal{A}2$ означают, что $\mathcal{A}$ — это подкатегория категории конечных множеств атрибутов и их согласований.

Для определения операций декартова произведения нам понадобится также следующее свойство класса $\mathcal{A}$:

$\mathcal{A}3$). Для любых наборов атрибутов A и B из класса $\mathcal{A}$ существует такой набор атрибутов $A \sqcup B$, который называется непересекающимся объединением или копроизведением наборов A и B, и такие согласования-вложения $i_1 : A \to A \sqcup B$, $i_2 : B \to A \sqcup B$ из класса $\mathcal{A}$, что $i_1(A) \cup i_2(B) = A \sqcup B$ и $i_1(A) \cap i_2(B) = \emptyset$. Если $\varphi_1 \in \mathcal{A}(A', A)$, $\varphi_2 \in \mathcal{A}(B', B)$, то согласование $\varphi_1 \sqcup \varphi_2 : A' \sqcup B' \to A \sqcup B$ принадлежит $\mathcal{A}$, где через $\varphi_1 \sqcup \varphi_2$ обозначено непересекающееся объединение согласований.

В этом разделе под классом $\mathcal{K}$ операций над отношениями понимается класс операций, удовлетворяющий условиям $\mathcal{K}1$ — $\mathcal{K}4$, порожденный операциями вида φ^* и $\underset{\varphi_1,\varphi_2}{\times}$ (см. определения 1.4, 1.5), где $\varphi, \varphi_1, \varphi_2$ — согласования из класса $\mathcal{A}$.

Выпишем некоторые соотношения в классе операций $\mathcal{K}$, которые нам потребуются в дальнейшем.

$C1$). Пусть $\varphi \in \mathcal{A}(A, B)$. Тогда

$$\varphi^*(\emptyset(A_1, ..., A_n; A)) = \emptyset(A_1, ..., A_n; B).$$

Это соотношение следует из определения операции φ^*.

$C2$). Пусть $id(A) : A \to A$ — тождественное согласование наборов атрибутов и $f \in \mathcal{K}(A_1, ..., A_n; A)$. Тогда $id(A)^*(f) = f$.

$C3$). Если $\varphi_1 \in \mathcal{A}(A, B)$ и $\varphi_2 \in \mathcal{A}(B, C)$, то $(\varphi_1 \circ \varphi_2)^* = \varphi_2^* \circ \varphi_1^*$.

$C4$). Если $f_i \in \mathcal{K}(A_1, ..., A_n; B)$, для $i \in I$, — произвольное множество операций класса $\mathcal{K}$, и $\varphi \in \mathcal{A}(A, B)$, то

$$\varphi^*(\bigcup_{i \in I} f_i) = \bigcup_{i \in I} \varphi^*(f_i).$$

Пусть $f_1, f_2 \in \mathcal{K}(A_1, ..., A_n; B)$. Будем писать, что $f_1 \subset f_2$, если $f_1 \cap f_2 = f_1$ или $f_1 \cup f_2 = f_2$.

Из предыдущего соотношения вытекает следующее свойство:

$C4'$). Если $f_1, f_2 \in \mathcal{K}(A_1, ..., A_n; B)$, где $f_1 \subset f_2$ и $\varphi \in \mathcal{A}(A, B)$, то $\varphi^*(f_1) \subset \varphi^*(f_2)$.

Действительно, так как $f_1 \subset f_2$, то $f_2 = f_1 \cup (f_2 \setminus f_1)$. Согласно соотношению $C4$, имеем $\varphi_*(f_2) = \varphi^*(f_1 \cup (f_2 \setminus f_1)) = \varphi^*(f_1) \cup \varphi^*(f_2 \setminus f_1)$, то есть $\varphi^*(f_1) \subset \varphi^*(f_2)$.

Следующие соотношения содержат операцию $\underset{\varphi_1, \varphi_2}{\times} \in \mathcal{K}(A, B; C)$ (см. определение 1.7), где $\varphi_1 \in \mathcal{A}(A, C)$ и $\varphi_2 \in \mathcal{A}(B, C)$ и образ отображения $(\varphi_1 \cup \varphi_2) : A \cup B \to C$ совпадает со всем C.

Из определения 1.7 вытекают следующие два соотношения.

$C5$). Если $f_1 \in \mathcal{K}(A_1, ..., A_n; A)$, $\quad f_2 \in \mathcal{K}(A_1, ..., A_n; B)$, $\varphi_1 \in \mathcal{A}(A, C)$ и $\varphi_2 \in \mathcal{A}(B, C)$, то $\varphi_1^*(f_1 \underset{\varphi_1, \varphi_2}{\times} f_2) \subset f_1$ и $\varphi_2^*(f_1 \underset{\varphi_1, \varphi_2}{\times} f_2) \subset f_2$.

$C6$). Пусть $f_1, f_2, \varphi_1, \varphi_2$ такие же, как в соотношении $C5$ и $f \in \mathcal{K}(A_1, ..., A_n;)$. Тогда, если $\varphi_1^*(f) \subset f_1$ и $\varphi_2^*(f) \subset f_2$, то $f \subset f_1 \underset{\varphi_1, \varphi_2}{\times} f_2$.

$C7$). Пусть $\varphi_1 \in \mathcal{A}(A, C)$, $\quad \varphi_2 \in \mathcal{A}(B, C)$ и $f \in \mathcal{K}(A_1, ..., A_n; B)$. Тогда

$$\emptyset(A_1, ..., A_n; A) \underset{\varphi_1, \varphi_2}{\times} f = \emptyset(A_1, ..., A_n; C).$$

Это равенство проверяется непосредственно, по определению 1.4 равенства между операциями, подстановкой любых отношений.

$C8$). Пусть $f_1, f_2 \in \mathcal{K}(A_1, ..., A_n; A)$, $f_3 \in \mathcal{K}(A_1, ..., A_n; B)$, φ_1, φ_2 — те же, что и прежде. Тогда

$$(f_1 \cup f_2) \underset{\varphi_1, \varphi_2}{\times} f_3 = (f_1 \underset{\varphi_1, \varphi_2}{\times} f_3) \cup (f_2 \underset{\varphi_1, \varphi_2}{\times} f_3).$$

Это равенство проверяется так же, как и равенство $C7$.

$C9$). Пусть $f \in \mathcal{K}(A_1, ..., A_n; B)$, $\varphi \in \mathcal{A}(A, B)$ и $id(B) : B \to B$ — тождественное отображение. Тогда

$$\varphi^*(f) \underset{\varphi, id(B)}{\times} f = f.$$

Действительно, согласно соотношениям $C2$ и $C5$, имеем:

$$\varphi^*(f) \underset{\varphi, id(B)}{\times} f = id_B^*(\varphi^*(f) \underset{\varphi, id(B)}{\times} f) \subset f.$$

С другой стороны, так как $\varphi_*(f) \subset \varphi_*(f)$ и $id_B^*(f) \subset f$, то по свойству $C6$ имеем $f \subset \varphi^*(f) \underset{\varphi, id(B)}{\times} f$, что доказывает равенство $C9$.

Пусть $i_1 : A \to A \sqcup B$, $i_2 : B \to A \sqcup B$ — естественные вложения в копроизведение наборов атрибутов. В этом случае, если $f_1 \in \mathcal{K}(A_1, ..., A_n; A)$ и $f_2 \in \mathcal{K}(A_1, ..., A_n; B)$, то операцию $f_1 \underset{i_1, i_2}{\times} f_2$ будем обозначать через $f_1 \times f_2$ и называть декартовым произведением операций f_1 и f_2.

$C10$). Пусть $f_1 \in \mathcal{K}(A_1, ..., A_n; A)$, $f_2 \in \mathcal{K}(A_1, ..., A_n; B)$ и $\varphi_1 \in \mathcal{A}(A', A)$, $\varphi_2 \in \mathcal{A}(B', B)$. Через $\varphi_1 \sqcup \varphi_2 : A' \sqcup B' \to A \sqcup B$ обозначается копроизведение отображений φ_1 и φ_2. Тогда

$$(\varphi_1 \sqcup \varphi_2)^*(f_1 \times f_2) = \varphi_1^*(f_1) \times \varphi_2^*(f_2).$$

$C11$). Для любой операции f выполняется равенство $i_1^*(f \times f) = f$, где $i_1 : A \to A \sqcup A$ — вложение одного слагаемого в копроизведение $A \sqcup A$.

Эти равенства непосредственно проверяются по определению 1.2 равенства между операциями подстановкой произвольных отношений.

Перейдем к изучению $\mathcal{K}$-алгебр $\mathcal{R}$ с конечным числом образующих. В таких алгебрах каждый элемент представляется значений некоторой операции от образующих (см. утверждение 3.8). Поэтому равенства между операциями (например $C1$ — $C11$) влекут соответствующие равенства между элементами $\mathcal{K}$-алгебр $\mathcal{R}$, где вместо произвольных операций берутся произвольные элементы $\mathcal{K}$-алгебры $\mathcal{R}$.

Рассмотрим $\mathcal{R}(A)$ для некоторого набора атрибутов A. Согласно определению $\mathcal{K}$-алгебры, $\mathcal{R}$ — булева алгебра с бесконечными пересечениями и объединениями.

Обозначим через $At\mathcal{R}(A)$ множество атомов булевой алгебры $\mathcal{R}(A)$, то есть множество минимальных непустых элементов $\mathcal{R}(A)$. Если $R \in \mathcal{R}(A)$, через $At(R)$ обозначим множество атомов, содержащихся в R.

Известно, что $R = \bigcup\limits_{t \in At(R)} t$, и булева алгебра $\mathcal{R}(A)$ изоморфна алгебре всех подмножеств множества $At\mathcal{R}(A)$. Если $j : \mathcal{R}(A) \to \mathcal{R}'(A)$ — гомоморфизм булевых алгебр, (напомним, что здесь рассматриваются булевы алгебры без единиц, у которых вместо унарной операции дополнения рассматривается бинарная операция разности) то ему соответствует частичное отображение множеств

$$At(j) : At\mathcal{R}'(A) \to At\mathcal{R}(A),$$

заданное формулой: $At(j)(t') \overset{def}{=} t$, если $t' \in At(j(t))$. И, наоборот, если $At(j) : At\mathcal{R}'(A) \to \mathcal{R}(A)$ — произвольное отображение множеств, то оно индуцирует гомоморфизм булевых алгебр

$$j : \mathcal{R}(A) \to \mathcal{R}'(A),$$

который задается соотношением $j(R) \overset{def}{=} (At(j))^{-1}(At(R))$, для любого $R \in \mathcal{R}(A)$.

Отображение $At(j)$ всюду определено тогда и только тогда, когда отображение j единицу булевой алгебры переводит в единицу.

Отображение $At(j)$ — вложение тогда и только тогда, когда отображение j является отображением на все $\mathcal{R}'(A)$

У т в е р ж д е н и е 5.1. Пусть $\varphi : A \to B$ — согласование наборов атрибутов, $\mathcal{R}$ — $\mathcal{K}$-алгебра и $t \in At\mathcal{R}(B)$. Тогда $\varphi^*(t)$ — атом булевой алгебры $\mathcal{R}(A)$.

Д о к а з а т е л ь с т в о. Покажем, что $\varphi^*(t) \neq \emptyset$. Если бы $\varphi^*(t) = \emptyset$, то $\varphi^*(t) \underset{\varphi, id(B)}{\times} t = \emptyset \underset{\varphi, id(B)}{\times} t = \emptyset$, согласно соотношению $C7$. С другой стороны, соотношение $C9$ дает: $\varphi^*(t) \underset{\varphi, id(B)}{\times} t = t$. Отсюда, $t = \emptyset$, что противоречит условию атомарности t. Покажем, теперь, что $\varphi^*(t)$ — минимальный элемент булевой алгебры. Пусть $\varphi^*(t) = r_1 \cup r_2$ и $r_1 \cap r_2 = \emptyset$. Тогда, по свойству $C9$, имеем $t = \varphi^*(t) \underset{\varphi, id(B)}{\times} t = (r_1 \cup r_2) \underset{\varphi, id(B)}{\times} t$. Далее, свойство $C8$ дает равенство $t = (r_1 \cup r_2) \underset{\varphi, id(B)}{\times} t = (r_1 \underset{\varphi, id(B)}{\times} t) \cup (r_2 \underset{\varphi, id(B)}{\times} t)$. Так как t — атомарный элемент в $\mathcal{R}(B)$, то t совпадает с одним из элементов правой части последнего выражения. Пусть, для определенности, $t = r_1 \underset{\varphi, id(B)}{\times} t$. Тогда $\varphi^*(t) = \varphi^*(r_1 \underset{\varphi, id(B)}{\times} t)$, но, согласно свойству $C5$, $\varphi^*(r_1 \underset{\varphi, id(B)}{\times} t) \subset r_1$. Отсюда, $\varphi^*(t) \subset r_1$ и $r_2 = \emptyset$.

Итак, для любого $\varphi \in \mathcal{A}(A, B)$ операция φ^* определяет отображение:

$$\varphi^* : At\mathcal{R}(B) \to At\mathcal{R}(A).$$

Если $R \in \mathcal{R}(B)$ и $R \neq \emptyset$, то $R = \bigcup\limits_{t \in At(R)} t$ и, согласно свойству $C4$,

$$\varphi^*(R) = \bigcup_{t \in At(R)} \varphi^*(t),$$

а по свойству $C1$, $\varphi^*(\emptyset) = \emptyset$. То есть $\varphi^*(R)$ — это образ подмножества $At(R) \subset At\mathcal{R}(B)$ при отображении $\varphi^* : At\mathcal{R}(B) \to At\mathcal{R}(A)$.

З а м е ч а н и е 5.1. Так как операция φ^* обладает свойствами $C2$ и $C3$, то $At\mathcal{R}$ представляет собой диаграмму множеств над категорией $\mathcal{A}$, в которой каждому объекту A сопоставляется множество $At\mathcal{R}(A)$, а каждому морфизму $\varphi : A \to B$ — отображение множеств $\varphi^* : At\mathcal{R}(B) \to At\mathcal{R}(A)$. Причем, $id(A)^* : At\mathcal{R}(A) \to At\mathcal{R}(A)$ — тождественное отображение множеств, и, если $\varphi_1 \in \mathcal{A}(A, B)$, $\varphi_2 \in \mathcal{A}(B, C)$, то $(\varphi_2 \circ \varphi_1)^* = \varphi_1^* \circ \varphi_2^*$.

У т в е р ж д е н и е 5.2. Пусть $\varphi_1 : A \to C$, $\varphi_2 : B \to C$ — согласования наборов атрибутов такие, что $\varphi_1 \sqcup \varphi_2 : A \sqcup B \to C$ — отображение на все C. Пусть $R_1 \in \mathcal{R}(A)$, $R_2 \in \mathcal{R}(B)$. Тогда $At(R_1 \underset{\varphi_1, \varphi_2}{\times} R_2)$ состоит в точности из тех атомов $t \in At\mathcal{R}(C)$, для которых $\varphi_1^*(t) \in At(R_1)$ и $\varphi_2^*(t) \in At(R_2)$.

Д о к а з а т е л ь с т в о. Если $t \in At(R_1 \underset{\varphi_1, \varphi_2}{\times} R_2)$, то $t \subset R_1 \underset{\varphi_1, \varphi_2}{\times} R_2$. Из свойств $C4'$ и $C5$ имеем:

$$\varphi_1^*(t) \subset \varphi_1^*(R_1 \underset{\varphi_1, \varphi_2}{\times} R_2) \subset R_1, \qquad \varphi_2^*(t) \subset \varphi_2^*(R_1 \underset{\varphi_1, \varphi_2}{\times} R_2) \subset R_2.$$

Так как t — атом, то, согласно утверждению 5.1, $\varphi_1^*(t)$ и $\varphi_2^*(t)$ — также атомы. Отсюда, $\varphi_1^*(t) \in At(R_1)$ и $\varphi_2^*(t) \in At(R_2)$. Обратно, пусть $t \in At\mathcal{R}(C)$ и $\varphi_1^*(t) \in At(R_1)$, $\varphi_2^*(t) \in At(R_2)$. Тогда $\varphi_1^*(t) \subset R_1$ и $\varphi_2^*(t) \subset R_2$, и по свойству $C6$ получим: $t \subset R_1 \underset{\varphi_1, \varphi_2}{\times} R_2$. Отсюда, $t \in At(R_1 \underset{\varphi_1, \varphi_2}{\times} R_2)$.

Из утверждения 5.2 следует, что

$$At(R_1 \underset{\varphi_1,\varphi_2}{\times} R_2) = (\varphi_1^*)^{-1}(At(R_1)) \cap (\varphi_2^*)^{-1}(At(R_2)).$$

О п р е д е л е н и е 5.1. Семейство подмножеств $M(A) \subset At\mathcal{R}(A)$ для всех наборов атрибутов $A \in \mathcal{A}$ называется идеальной поддиаграммой диаграммы $At\mathcal{R}$, если выполняются следующие условия:

1). если $t \in M(B)$ и $\varphi \in \mathcal{A}(A,B)$, то $\varphi^*(t) \in M(A)$;

2). если $t \in At\mathcal{R}(B)$ и $\varphi^*(t) \in M(A)$, то $t \in M(B)$.

У т в е р ж д е н и е 5.3. Пусть $At\mathcal{R}$ — диаграмма множеств, соответствующая $\mathcal{K}$-алгебре $\mathcal{R}$, и M — идеальная поддиаграмма диаграммы $At\mathcal{R}$. Тогда существует подалгебра $\mathcal{R}M \subset \mathcal{R}$ такая, что $At\mathcal{R}M(A) = M(A)$ и $\mathcal{R}M(A)$ изоморфна булевой алгебре всех подмножеств множества $M(A)$.

Д о к а з а т е л ь с т в о. Положим $\mathcal{R}M(A) = \{R : R \in \mathcal{R}(A), At(R) \subset M(A)\}$. Очевидно, что $\mathcal{R}M(A)$ — подалгебра булевой алгебры $\mathcal{R}$. Из утверждения 5.1 и определения 5.1 идеальности M следует, что, если $R \in \mathcal{R}M(B)$, $\varphi \in \mathcal{A}(A,B)$, то $\varphi^*(R) \in \mathcal{R}M(A)$. Аналогично, из утверждения 5.2 следует, что, если $\varphi_1 \in \mathcal{A}(A,C)$, $\varphi_2 \in \mathcal{A}(B,C)$ и $R_1 \in \mathcal{R}M(A)$, $R_2 \in \mathcal{R}M(B)$, то $R_1 \underset{\varphi_1,\varphi_2}{\times} R_2 \in \mathcal{R}M(C)$. Так как все операции являются композициями операций φ^*, $\underset{\varphi_1,\varphi_2}{\times}$ и теоретико множественных операций, то $\mathcal{R}M$ — подалгебра $\mathcal{R}$.

У т в е р ж д е н и е 5.4. Пусть $\mathcal{R}M$ — алгебра предыдущего утверждения. Тогда отображение $j : \mathcal{R} \to \mathcal{R}M$, сопоставляющее элементу $R \in \mathcal{R}(A)$ элемент $R \cap M(A) \in \mathcal{R}M$, является $\mathcal{K}$-гомоморфизмом.

Д о к а з а т е л ь с т в о. Очевидно, что $j : \mathcal{R}(A) \to \mathcal{R}M(A)$ — гомоморфизм булевых алгебр. Так как любая операция из класса $\mathcal{K}$ является композицией операций φ^*, $\underset{\varphi_1,\varphi_2}{\times}$ и теоретико множественных операций, то осталось показать, что

$$j(\varphi^*(R)) = \varphi^*(j(R)) \text{ и } j(R_1 \underset{\varphi_1,\varphi_2}{\times} R_2) = j(R_1) \underset{\varphi_1,\varphi_2}{\times} j(R_2).$$

Для доказательства первого равенства рассмотрим $At(j(\varphi^*(R))$. Так как, согласно равенству $C4$, $\varphi^*(R) = \bigcup_{t \in At(R)} \varphi^*(t)$ и j — гомоморфизм булевых алгебр, то $j(\varphi^*(R)) = \bigcup_{t \in At(R)} j(\varphi^*(t))$. По определению отображения j имеем: $j(\varphi^*(t)) = \varphi^*(t) \cap M(A)$.

Так как $\varphi^*(t)$, согласно утверждению 5.1, это атом, то

$$At(j(\varphi^*(R)) = \{\varphi^*(t) : t \in At(R) \text{ и } \varphi^*(t) \in M(A)\}.$$

Но, если $\varphi^*(t) \in M(A)$, то из определения идеальности M следует, что $t \in M(B)$. Отсюда,

$$At(j(\varphi^*(R)) = \{\varphi^*(t) : t \in At(R) \text{ и } t \in M(B)\} = \{\varphi^*(t) : t \in At(j(R))\}.$$

Окончательно имеем,

$$j(\varphi^*(R)) = \bigcup_{t' \in At(j(\varphi^*(R)))} t' = \bigcup_{t \in At(j(R))} \varphi^*(t) = \varphi^*(\bigcup_{t \in At(j(R))} t) = \varphi^*(j(R)).$$

Для доказательства коммутативности j с операцией $\underset{\varphi_1,\varphi_2}{\times}$ рассмотрим множество атомов $At(j(R_1 \underset{\varphi_1,\varphi_2}{\times} R_2))$. По определению отображения j и из утверждения 5.2 следует, что

$$At(j(R_1 \underset{\varphi_1,\varphi_2}{\times} R_2)) = \{t : t \in M(C),\ \varphi_1^*(t) \in At(R_1), \varphi_2^*(t) \in At(R_2)\}.$$

Так как M — идеальная поддиаграмма, то условия $t \in M(C)$ и $\varphi_1^*(t) \in M(A), \varphi_2^*(t) \in M(B)$ эквивалентны. Отсюда,

$$At(j(R_1 \underset{\varphi_1,\varphi_2}{\times} R_2)) = \{t : t \in At\mathcal{R}(C), \varphi_1^*(t) \in At(j(R_1)), \varphi_2^*(t) \in At(j(R_2))\}$$

и из утверждения 5.2 имеем:

$$j(R_1 \underset{\varphi_1,\varphi_2}{\times} R_2) = j(R_1) \underset{\varphi_1,\varphi_2}{\times} j(R_2),$$

что завершает доказательство утверждения 5.4.

Положим $M'(A) = At\mathcal{R}(A) \setminus M(A)$. Очевидно, что, если M — идеальная поддиаграмма, то и M' — идеальная поддиаграмма.

Утверждение 5.5. Гомоморфизмы $j_M : \mathcal{R} \to \mathcal{R}M$ и $j_{M'} : \mathcal{R} \to \mathcal{R}M'$ представляют $\mathcal{K}$-алгебру $\mathcal{R}$ в виде суммы $\mathcal{R} = \mathcal{R}M \oplus \mathcal{R}M'$.

Действительно, любой элемент $R \in \mathcal{R}(A)$ представляется в виде пары $(R \cap M(A), R \cap M'(A)) = (j_M(R), j_{M'}(R))$ так, что $R = j_M(R) \cup j_{M'}(R)$ и $j_M(R) \cap j_{M'}(R) = \emptyset$. Наоборот, паре $(R_M, R_{M'}) \in (\mathcal{R}M \oplus \mathcal{R}M')(A)$ сопоставляется элемент $R = R_M \cup R_{M'}$, так как $R_M \in \mathcal{R}M(A) \subset \mathcal{R}$ и $R_{M'} \in$

$\mathcal{R}M'(A) \subset \mathcal{R}$. Легко видеть, что эта пара отображений представляют собой взаимно обратные изоморфизмы $\mathcal{K}$-алгебр $\mathcal{R}$ и $\mathcal{R}M \oplus \mathcal{R}M'$.

С л е д с т в и е 5.6. Если M — идеальная поддиаграмма диаграммы $At\mathcal{R}$, то $\mathcal{R}M \subset \mathcal{R}$ — идеал алгебры $\mathcal{R}$.

Пусть $\mathcal{J} \subset \mathcal{R}$ — идеал $\mathcal{K}$-алгебры $\mathcal{R}$. Обозначим через $At\mathcal{J}(A)$ подмножество в $At\mathcal{R}(A)$, состоящее из всех атомов, содержащихся в $\mathcal{J}(A)$.

У т в е р ж д е н и е 5.7. Для произвольного идеала $\mathcal{J} \subset \mathcal{R}$ поддиаграмма $At\mathcal{J}$ идеальна в $At\mathcal{R}$.

Д о к а з а т е л ь с т в о. Пусть $t \in At\mathcal{J}(B)$, то есть t — атом и $t \in \mathcal{J}(B)$. Тогда, согласно утверждению 3.1, $\varphi^*(t) \in \mathcal{J}(A)$, так как равенство $C1$ дает $\varphi^*(\emptyset) = \emptyset$. Отсюда, если $t \in At\mathcal{J}(B)$, то $\varphi^*(t) \in At\mathcal{J}(A)$. Пусть, теперь, $\varphi^*(t) \in At\mathcal{J}(A)$, то есть $\varphi^*(t) \in \mathcal{J}(A)$ и $\varphi^*(t)$ — атом. Из равенства $C9$ следует, что $\varphi^*(t) \underset{\varphi,id(B)}{\times} t = t$. Так как согласно равенству $C7$ выполняется соотношение $\emptyset \underset{\varphi,id(B)}{\times} t = \emptyset$, то из утверждения 3.1 следует, что $t = \varphi^*(t) \underset{\varphi,id(B)}{\times} t \in \mathcal{J}(B)$.

Следствие 5.6 и утверждение 5.7 дают взаимно однозначное соответствие между идеалами $\mathcal{K}$-алгебры $\mathcal{R}$ и идеальными поддиаграммами диаграммы $At\mathcal{R}$.

Пусть $\mathcal{R}$ — произвольная $\mathcal{K}$-алгебра. Из определения 5.1 легко выводится, что пересечение любого множества идеальных поддиаграмм диаграммы $At\mathcal{R}$ — идеальная поддиаграмма, объединение любого множества идеальных поддиаграмм — идеальная поддиаграмма и разность двух идеальных поддиаграмм — идеальная поддиаграмма. То есть совокупность всех идеальных поддиаграмм диаграммы $At\mathcal{R}$ образуют полную булеву алгебру.

У т в е р ж д е н и е 5.8. Пусть $R_1 \in \mathcal{R}(A_1), ..., R_n \in \mathcal{R}(A_n)$ — множество образующих $\mathcal{K}$-алгебры $\mathcal{R}$; M — идеальная поддиаграмма диаграммы $At\mathcal{R}$ такая, что $M(A_i) \cap At(R_i) = \emptyset$, для $i = 1, ..., n$. Тогда $M = \emptyset$.

Д о к а з а т е л ь с т в о. Согласно утверждению 5.5, $\mathcal{K}$-алгебра $\mathcal{R}$ представляется в виде $\mathcal{R} = \mathcal{R}M \oplus \mathcal{R}M'$, где M' — дополнительная к M диаграмма. Из следствия 5.6 вытекает, что $\mathcal{R}M' \subset \mathcal{R}$ — идеал $\mathcal{K}$-алгебры $\mathcal{R}$. По условию $\mathcal{R}M'$ содержит образующие R_i, для $i = 1, ..., n$. Отсюда, $\mathcal{R}M' = \mathcal{R}$ и $M = \emptyset$.

С л е д с т в и е 5.9. Пусть $R_1 \in \mathcal{R}(A_1), ..., R_n \in \mathcal{R}(A_n)$ — множество образующих $\mathcal{K}$-алгебры $\mathcal{R}$; M_1 и M_2 идеальные поддиаграммы $At\mathcal{R}$.

Тогда, если $M_1(A_i) \cap R_i = M_2(A_i) \cap R_i$, для $i = 1, ..., n$, то $M_1 = M_2$.

Для доказательства нужно рассмотреть симметрическую разность M_1 и M_2 и применить утверждение 5.8

С л е д с т в и е 5.10. Любой идеал $\mathcal{K}$-алгебры $\mathcal{R}$ с n образующими порождается n элементами.

Действительно, пусть $\mathcal{J} \subset \mathcal{R}$ — идеал и $R_i \in \mathcal{R}(A_i)$, $i = 1, ..., n$ — образующие $\mathcal{K}$-алгебры $\mathcal{R}$. Согласно утверждению 5.7 поддиаграмма $At\mathcal{J}$ идеальна в $At\mathcal{R}$. Рассмотрим идеал $\mathcal{J}[J_1, ...J_n]$, порожденный элементами $J_i = At\mathcal{J}(A_i) \cap R_i$, для $i = 1, ..., n$. Так как $J_i \in \mathcal{J}$, то $\mathcal{J}[J_1, ...J_n] \subset \mathcal{J}$. С другой стороны, применяя следствие 5.9, получим $At\mathcal{J}[J_1, ...J_n] = At\mathcal{J}$. Отсюда, $\mathcal{J}[J_1, ...J_n] = \mathcal{J}$.

Рассмотрим булеву алгебру $\mathcal{JR}$ всех идеальных поддиаграмм диаграммы $At\mathcal{R}$. Через $Spec\mathcal{R}$ обозначим множество атомов булевой $\mathcal{JR}$.

О п р е д е л е н и е 5.2. $\mathcal{K}$-алгебра $\mathcal{R}$ называется неприводимой, если она не представляется в виде суммы своих нетривиальных подалгебр, то есть $\mathcal{R} \neq \mathcal{R}_1 \oplus \mathcal{R}_2$, где $\mathcal{R}_1 \neq \emptyset$ и $\mathcal{R}_2 \neq \emptyset$.

Из утверждений 5.7 и 5.5 следует, что, если $\mathcal{R}$ — неприводимая $\mathcal{K}$-алгебра, то в ней нет собственных идеалов.

Для изучения строения неприводимых $\mathcal{K}$-алгебр нам понадобится следующее понятие.

Пусть $R \in \mathcal{R}$ — произвольный элемент $\mathcal{K}$-алгебры $\mathcal{R}$. Для каждого набора атрибутов $A \in \mathcal{A}$ рассмотрим множество $N_R(A) = \{R_1 : R_1 \in \mathcal{R}(A)$ и $R \times R_1 = \emptyset\}$.

У т в е р ж д е н и е 5.11. Для любого элемента $R \in \mathcal{R}(C)$ подмножества $N_R(A)$, где $A \in \mathcal{A}$, образуют идеал N_R в $\mathcal{K}$-алгебре $\mathcal{R}$.

Д о к а з а т е л ь с т в о. Пусть $R_i \in N_R(A)$, для $i \in I$. Рассмотрим множество атомов элемента $R \times (\bigcup\limits_{i \in I} R_i)$. Согласно утверждению 5.2, t — атом $R \times (\bigcup\limits_{i \in I} R_i)$ тогда и только тогда, когда $i_1^*(t) \in At(R)$ и $i_2^*(t) \in At(\bigcup\limits_{i \in I} R_i)$, где $i_1 : C \to C \sqcup A$, $\quad i_2 : A \to C \sqcup A$. Так как $At(\bigcup\limits_{i \in I} R_i) = \bigcup\limits_{i \in I} At(R_i)$, то $i_2^*(t) \in At(R_i)$, для некоторого $i \in I$. Тогда из утверждения 5.2 следует, что $t \in At(R \times R_i)$, что противоречит условию $R \times R_i = \emptyset$. Отсюда следует равенство $R \times (\bigcup\limits_{i \in I} R_i) = \emptyset$ или $\bigcup\limits_{i \in I} R_i \in N_R(A)$. Аналогично показывается, что $\bigcap\limits_{i \in I} R_i \in N_R(A)$ и

$R_1 \setminus R_2 \in N_R(A)$, если $R_1, R_2 \in N_R(A)$. Кроме того, если $R_1 \in N_R(A)$ и R' — произвольный элемент $\mathcal{R}(A)$, то $R_1 \cap R' \in N_R(A)$. Таким образом доказывается, что $N_R(A)$ — идеал булевой алгебры $\mathcal{R}(A)$. Для завершения доказательства нужно показать, что $At(N_R(A))$ — идеальная поддиаграмма диаграммы $At\mathcal{R}$ и воспользоваться следствием 5.6. Пусть $\varphi \in \mathcal{A}(A; B)$ и $t \in At(N_R(B))$. Рассмотрим $R \times \varphi^*(t)$. Согласно равенствам $C10$ и $C1$ имеем, $R \times \varphi^*(t) = (id(C) \cup \varphi)^*(R \times t) = (id(C) \cup \varphi)^*(\emptyset) = \emptyset$. То есть $\varphi^*(t) \in At(N_R(A))$. Пусть, теперь, $t \in At\mathcal{R}(B)$ и $\varphi^*(t) \in At(N_R(A))$. Рассмотрим $(id(C) \cup \varphi)^*(R \times t)$. Применяя равенство $C10$, получим $(id(C) \cup \varphi)^*(R \times t) = R \times \varphi^*(t) = \emptyset$. Отсюда, $R \times t = \emptyset$, так как, если бы $R \times t$ было непусто, то и $(id(C) \cup \varphi)^*(R \times t) = \bigcup\limits_{t' \in At(R \times t)} (id(C) \cup \varphi)^*(t')$ было бы непусто. Таким образом, $At(N_R)$ удовлетворяет определению 5.1 идеальной поддиаграммы.

У т в е р ж д е н и е 5.12. $\mathcal{K}$-алгебра $\mathcal{R}$ неприводима тогда и только тогда, когда не существует непустых элементов R_1 и $R_2 \in \mathcal{R}$ таких, что декартово произведение $R_1 \times R_2 = \emptyset$.

Д о к а з а т е л ь с т в о. Пусть $\mathcal{R}$ — неприводимая $\mathcal{K}$-алгебра и R_1 — произвольный непустой ее элемент. Рассмотрим идеал $N_{R_1} \subset \mathcal{R}$. Так как согласно равенству $C11$ $i_1^*(R_1 \times R_1) = R_1$ непусто, то $R_1 \times R_1 \neq \emptyset$. Отсюда, $R_1 \notin N_{R_1}$. В неприводимой $\mathcal{K}$-алгебре $\mathcal{R}$ нет собственных идеалов и $N_{R_1} \neq \mathcal{R}$, поэтому $N_{R_1} = \emptyset$. То есть $R_1 \times R_2 \neq \emptyset$ для любых непустых элементов R_1 и R_2.

Обратно, пусть $\mathcal{R}$ — приводима, то есть $\mathcal{R} = \mathcal{R}' \oplus \mathcal{R}''$ и $\mathcal{R}' \neq \emptyset$ и $\mathcal{R}'' \neq \emptyset$. Тогда существуют непустые элементы $R' \in \mathcal{R}'$ и $R'' \in \mathcal{R}''$. Рассмотрим элементы $\mathcal{R} = \mathcal{R}' \oplus \mathcal{R}''$ вида $R_1 = (R', \emptyset)$ и $R_2 = (\emptyset, R'')$. Они непусты и

$$R_1 \times R_2 = (R_1, \emptyset) \times R_2(\emptyset, R_2) = (R_1 \times \emptyset,\ \emptyset \times R_2) = \emptyset.$$

С л е д с т в и е 5.13. Подалгебра $\mathcal{R}_1 \subset \mathcal{R}$ неприводимой $\mathcal{K}$-алгебры $\mathcal{R}$ — неприводима.

С л е д с т в и е 5.14. $\mathcal{K}$-алгебра неприводима тогда и только тогда, когда для любых непустых элементов $R_1, R_2 \in \mathcal{R}$ имеется равенство $i_1^*(R_1 \times R_2) = R_1$, где i_1 — естественное вложение, определяющее декартово произведение.

Д о к а з а т е л ь с т в о. Действительно, если для любых непустых элементов $R_1, R_2 \in \mathcal{R}$ имеется равенство $i_1^*(R_1 \times R_2) = R_1 \neq \emptyset$, то и $R_1 \times R_2 \neq \emptyset$, то есть в $\mathcal{R}$ нет делителей пустого элемента. Тогда из утверждения 5.12 вытекает, что $\mathcal{R}$ — неприводима. Обратно, пусть $\mathcal{R}$ неприводима и $R_1, R_2 \in \mathcal{R}$ — непустые ее элементы. Покажем, что элемент $(R_1 \setminus i_1^*(R_1 \times R_2)) \times R_2 = \emptyset$. Действительно, так как $(R_1 \setminus i_1^*(R_1 \times R_2)) \times R_2 \subset R_1 \times R_2$, то, согласно свойству $C4'$, имеем:

$$i_1^*[(R_1 \setminus i_1^*(R_1 \times R_2)) \times R_2] \subseteq i_1^*(R_1 \times R_2).$$

С другой стороны, из свойства $C5$ следует, что

$$i_1^*[(R_1 \setminus i_1^*(R_1 \times R_2)) \times R_2] \subseteq R_1 \setminus i_1^*(R_1 \times R_2).$$

Отсюда $i_1^*[(R_1 \setminus i_1^*(R_1 \times R_2)) \times R_2] = \emptyset$ и, следовательно, $(R_1 \setminus i_1^*(R_1 \times R_2)) \times R_2 = \emptyset$. Так как $\mathcal{R}$ — неприводима, то согласно утверждению 5.12, в ней нет делителей пустого элемента, поэтому $R_1 \setminus i_1^*(R_1 \times R_2) = \emptyset$. Так как $i_1^*(R_1 \times R_2) \subset R_1$, то $i_1^*(R_1 \times R_2) = R_1$.

Т е о р е м а 5.15. Пусть $\mathcal{R}$ — произвольная $\mathcal{K}$-алгебра с конечным множеством образующих. Тогда $\mathcal{R}$ представляется в виде прямой суммы неприводимых $\mathcal{K}$-алгебр $\mathcal{R} = \bigoplus\limits_{s \in Spec\mathcal{R}} \mathcal{R}_s$ по множеству $Spec\mathcal{R}$ — атомов булевой алгебры идеальных поддиаграмм диаграммы $At\mathcal{R}$. Если $j : \mathcal{R} \to \mathcal{R}'$ — гомоморфизм $\mathcal{K}$-алгебр, то он индуцирует естественное частичное отображение

$$j : Spec\mathcal{R}' \to Spec\mathcal{R}$$

и множество гомоморфизмов-вложений

$$j_{s'} : \mathcal{R}_{j(s')} \to \mathcal{R}'_{s'},$$

для всех $s' \in Spec\mathcal{R}'$, для которых определено отображение j так, что $j(R) = j(\bigoplus\limits_{s \in Spec\mathcal{R}} R_s) = \bigoplus\limits_{s' \in Spec\mathcal{R}'} j_{s'}(R_{j(s')})$, где R — произвольный элемент $\mathcal{R}$.

Д о к а з а т е л ь с т в о. Рассмотрим булеву алгебру идеальных поддиаграмм $\mathcal{J}(\mathcal{R})$ диаграммы $At\mathcal{R}$. Пусть $Spec\mathcal{R}$ — множество атомов булевой алгебры $\mathcal{J}(\mathcal{R})$. Тогда для каждой поддиаграммы $s \in Spec\mathcal{R}$ алгебра $\mathcal{R}_s$ неприводима, так как в противном случае, согласно утверждениям 5.7 и 5.5, поддиаграмма s не являлась бы атомом булевой алгебры $\mathcal{J}(\mathcal{R})$.

Так как $At\mathcal{R} = \bigcup\limits_{s \in Spec\mathcal{R}} s$, то любой элемент $R \in \mathcal{R}(A)$ единственным образом представляется в виде $R = \bigcup\limits_{s \in Spec\mathcal{R}} R_s$, где $R_s \in \mathcal{R}_s(A)$ и определяются по формуле $R_s = R \cap s(A)$. Очевидно, что операции над элементами $R = \bigcup_{s \in Spec\mathcal{R}} R_s$ выполняются покоординатно. Отсюда $\mathcal{R} = \bigoplus_{s \in Spec\mathcal{R}} \mathcal{R}_s$

Пусть $j : \mathcal{R} \to \mathcal{R}'$ — гомоморфизм $\mathcal{K}$-алгебр. Тогда $\mathcal{R} = \bigoplus_{s \in Spec\mathcal{R}} \mathcal{R}_s$ и $\mathcal{R}' = \bigoplus_{s' \in Spec\mathcal{R}'} \mathcal{R}'_{s'}$. Для каждой пары элементов $s \in Spec\mathcal{R}$ и $s' \in Spec\mathcal{R}'$ рассмотрим сквозной гомоморфизм

$$i_{s,s'} : \mathcal{R}_s \subset \mathcal{R} \xrightarrow{j} \mathcal{R}' \to \mathcal{R}'_{s'},$$

где крайние гомоморфизмы — это естественные вложения и проекции прямой суммы. Так как $\mathcal{R}_s$ — неприводимая $\mathcal{K}$-алгебра и в ней нет идеалов, то $i_{s,s'}$ — либо вложение, либо отображение в пустой элемент $\mathcal{R}'_s$.

Пусть $s_1 \neq s_2 \in Spec\mathcal{R}$. Покажем, что для любого элемента $s' \in Spec\mathcal{R}'$ оба гомоморфизма $i_{s_1,s'}$ и $i_{s_2,s'}$ не могут быть вложениями. Для этого рассмотрим сквозной гомоморфизм

$$i_{s_1 s_2,s'} : \mathcal{R}_{s_1} \oplus \mathcal{R}_{s_2} \subset \mathcal{R} \xrightarrow{j} \mathcal{R}' \to \mathcal{R}'_{s'},$$

где крайние гомоморфизмы — естественное вложение и проекция прямой суммы. Очевидно, что ограничение гомоморфизма $i_{s_1 s_2,s'}$ на $\mathcal{R}_{s_1}$ и $\mathcal{R}_{s_2}$ совпадает, соответственно, с $i_{s_1 s'}$ и $i_{s_2 s'}$. Возьмем непустые элементы вида $(R_1, \emptyset)$ и $(\emptyset, R_2)$ из $\mathcal{R}_{s_1} \oplus \mathcal{R}_{s_2}$, где $R_1 \in \mathcal{R}_{s_1}$ $R_2 \in \mathcal{R}_{s_2}$. Так как декартово произведение $(R_1, \emptyset) \times (\emptyset, R_2) = (R_1 \times \emptyset, \emptyset \times R_2) = \emptyset$, то произведение образов элементов $(R_1, \emptyset)$, $(\emptyset, R_2)$ при гомоморфизме $i_{s_1 s_2,s'}$ пусто, то есть

$$i_{s_1 s_2,s'}(R_1, \emptyset) \times i_{s_1 s_2,s'}(\emptyset, R_2) = i_{s_1 s_2,s'}((R_1, \emptyset) \times (\emptyset, R_2)) = \emptyset.$$

Из неприводимости $\mathcal{R}'_{s'}$ следует (утверждение 5.12), что один из элементов $i_{s_1 s_2,s'}(R_1, \emptyset)$ или $i_{s_1 s_2,s'}(\emptyset, R_2)$ пуст. Для определенности предположим, что $i_{s_1 s_2,s'}(R_1, \emptyset) = \emptyset$. Отсюда, $i_{s_1,s'}(R_1, \emptyset) = \emptyset$, то есть $i_{s_1,s'}$ — не вложение.

Положим $j(s') = s$, если $i_{s_1,s'}$ — вложение. Таким образом, j — частичное отображение. Кроме того, положим $j_{s'} = i_{s_1,s'}$, для $s = j(s')$. Это завершает доказательство теоремы.

6 Описание и классификация неприводимых реляционных алгебр

По теореме предыдущего раздела произвольная реляционная алгебра с конечным множеством образующих представляется в виде канонической прямой суммы неприводимых реляционных алгебр. Поэтому, чтобы завершить описание реляционных алгебр, нужно дать описание неприводимых реляционных алгебр.

Т е о р е м а 6.1. Если $\mathcal{R}$ — неприводимая $\mathcal{K}$-алгебра с конечным множеством образующих, тогда существует гомоморфизм - вложение $\mathcal{R} \to \mathcal{D}$, где $\mathcal{D}$ — $\mathcal{K}$-алгебра действительных отношений (см. пример 2.1).

Д о к а з а т е л ь с т в о. Пусть $R_i \in \mathcal{R}(A_i)$, для $i = 1, ..., n$ — образующие $\mathcal{K}$-алгебры $\mathcal{R}$. Рассмотрим свободную $\mathcal{K}$-алгебру $\mathcal{K}[A_1, ..., A_n]$ и гомоморфизм $j : \mathcal{K}[A_1, ..., A_n] \to \mathcal{R}$, который переводит образующие A_i в R_i, то есть $j(A_i) = R_i$, для $i = 1, ..., n$. Тогда, так как j — гомоморфизм на всю алгебру, то, согласно утверждениям 5.7 и 5.5, имеем равенство:

$$\mathcal{K}[A_1, ..., A_n] = j^{-1}(\emptyset) \oplus \mathcal{R}.$$

Отсюда следует, что $\mathcal{R}$ вкладывается в $\mathcal{K}[A_1, ..., A_n]$.

Пусть $R \in \mathcal{R}(A) \subset \mathcal{K}[A_1, ..., A_n]$ — непустой элемент $\mathcal{R}$. Согласно построению свободной $\mathcal{K}$-алгебры $\mathcal{K}[A_1, ..., A_n]$ ее элемент R представляется некоторой операцией над отношениями $f \in \mathcal{K}(A_1, ..., A_n; A)$. Так как по построению f — непустая операция, то по определению равенства операций существуют такие действительные отношения, что $r_i \in \mathcal{D}(A_i)$, что $f(r_1, ..., r_n) \neq \emptyset$.

Рассмотрим гомоморфизм $(r_1, ..., r_n) : \mathcal{K}[A_1, ..., A_n] \to \mathcal{D}$, переводящий образующие A_i алгебры $\mathcal{K}[A_1, ..., A_n]$ в $r_i, \quad i = 1, ..., n$. Этот гомоморфизм переводит элемент $R \in \mathcal{R} \subset \mathcal{K}[A_1, ..., A_n]$, соответствующий f, в непустой элемент $\mathcal{K}$-алгебры $\mathcal{D}$. Поэтому ограничение гомоморфизма $(r_1, ..., r_n)$ на $\mathcal{R}$ — вложение, так как $\mathcal{R}$ неприводима и не имеет собственных идеалов. Это завершает доказательство теоремы.

Заметим, что, согласно следствиям 5.13 и 5.14, каждая подалгебра алгебры реальных отношений $\mathcal{D}$ неприводима. Поэтому по теореме 5.1, чтобы описать все неприводимые реляционные алгебры, нужно описать все реляционные подалгебры алгебры $\mathcal{D}$.

Далее будет предполагаться, что подалгебра $\mathcal{R}$ содержит единицы алгебры $\mathcal{D}$. Это предположение не сильно ограничивает общность, так как в противном случае, если в качестве областей значений атрибутов взять элементы, входящие хотя бы в одно из отношений алгебры $\mathcal{R}$, то условие будет выполнено.

Для начала рассмотрим алгебру реальных отношений, введенную в разделе 1, над категорией наборов атрибутов $\bar{\mathcal{A}}$ без условия конечности элементов в наборах. Через D, как и раньше, обозначается разъединенное объединение областей значений всех типов атрибутов, рассматриваемое как набор атрибутов. Для любого набора атрибута A, конечного или бесконечного, множество всех отношений $\mathcal{D}(A)$ с набором атрибутов A по определению совпадает с множеством всех подмножества в $\mathcal{A}(A, D)$, где $\mathcal{A}(A, D)$ — множество всех отображений (согласований) из набора атрибутов A в D, сохраняющие типы атрибутов.

Пусть $\mathcal{R}$ — реляционная подалгебра в $\mathcal{D}$. Это означает, что для каждого набора атрибутов A подмножество $\mathcal{R}(A) \subset \mathcal{D}(A)$ является полной булевой подалгеброй, и для каждого согласования атрибутов $\varphi : B \to A$ и произвольных отношений $R \in \mathcal{R}(A)$ и $R' \in \mathcal{R}(B)$ отношения $\varphi^*(R) \in \mathcal{R}(B)$ и $\varphi_*(R') \in \mathcal{R}(A)$.

Рассмотрим полную булеву алгебру $\mathcal{R}(D)$. Она является подалгеброй $\mathcal{D}(D)$, которая, в свою очередь, представляет собой булеву алгебру всех подмножеств множества $\mathcal{A}(D, D)$. Так как $\mathcal{R}(D)$ содержит единицу $E(D)$ булевой алгебры $\mathcal{D}(D)$ и в ней определена операция бесконечного пересечения, то существует наименьший элемент $H \in \mathcal{R}(D)$, содержащий тождественное отображение $id(D) \in \mathcal{A}(D, D)$.

У т в е р ж д е н и е 6.2. Подмножество $H \subset \mathcal{A}(D, D)$ является группой относительно операции композиции.

Д о к а з а т е л ь с т в о. Подмножество H по построению содержит единицу $id(D) \in H$. Пусть $h \in H \subset \mathcal{A}(D, D)$ — произвольный элемент множества H. Рассмотрим отношение $h^*(H)$. По определению операции h^*, отношение $h^*(H)$ состоит из всех элементов вида $h_1 \circ h$, где $h_1 \in H$, то есть $h^*(H) = H \circ h$ (знаком $\circ$ обозначена операция композиции отображений). В частности, элемент $h = id(D) \circ h \in h^*(H)$. Таким образом, $h^*(H)$ и H имеют непустое пересечение. Так как H — атом булевой алгебры $\mathcal{R}(D)$, то, согласно утверждению 5.1, $h^*(H)$ — атом и $H = h^*(H)$. Итак доказано, что для любого $h \in H$ выполняется равенство $H = H \circ h$, то есть композиция отображений из $H \subset \mathcal{A}(D, D)$ принадлежит

H и для каждого $h \in H$ найдется $h^{-1} \in H$, что $h^{-1} \circ h = id(D)$.

Утверждение 6.3. Для любого атома R булевой алгебры $\mathcal{R}(A)$ существует согласование наборов атрибутов $\varphi : A \to D$ такое, что $R = \varphi^*(H)$.

Доказательство. Так как $\mathcal{R}$ — подалгебра $\mathcal{D}$, то элемент $R \in \mathcal{R}(A)$ — это некоторое подмножество $R \subset \mathcal{A}(A, D)$. Элемент R непустой, поэтому существует некоторое отображение $\varphi : A \to D$, принадлежащее R. Рассмотрим $\varphi^*(H)$. По определению операции φ^* множество (отношение) $\varphi^*(H) = H \circ \varphi$ состоит из отображений набора A в D, представимых в виде композиции отображения φ и отображений $h \in H$. В частности, элемент $\varphi = id(D) \circ \varphi \in \varphi^*(H)$. Таким образом, R и $\varphi^*(H)$ имеют непустое пересечение. Так как R и $\varphi^*(H)$ — атомы, то $R = \varphi^*(H)$.

Перейдем к другому построению той же группы H.

Обозначим через $S(D)$ множество всех взаимно однозначных отображений D на себя, сохраняющих типы атрибутов. Очевидно, что $S(D)$ образует группу относительно композиции отображений. Будем говорить, что отношение R симметрично относительно $h \in S(D)$, если отношение R не изменяется после замены значений атрибутов под действием отображения h, то есть $h \circ R = R$, где через $\circ$, как и прежде, обозначена операция композиции отображений из $R \subset \mathcal{A}(A, D)$ и отображения $h \in \mathcal{D}$.

Утверждение 6.4. Пусть $R_i \in \mathcal{D}(A)$, для $i \in I$, — некоторое множество отношений, симметричных относительно $h \in S(D)$. Тогда отношения $\bigcup_{i \in I} R_i$, $\bigcap_{i \in I} R_i$, $R_i \setminus R_j$ симметричны относительно h. Если $\varphi : B \to A$ и $\psi : A \to C$ — произвольные согласования наборов атрибутов, то отношения $\varphi^*(R_i)$ и $\psi_*(R_i)$ также симметричны относительно h.

Доказательство очевидно.

Обозначим через $S\{R_i\}_{i \in I}$ множество всех отображений $h \in S(D)$, относительно которых симметричны отношения R_i, при $i \in I$. Очевидно, что множество отображений $S\{R_i\}_{i \in I}$ образует подгруппу в $S(D)$, которая называется группой симметрии данного множества отношений.

Из утверждения 6.4 вытекает следующее утверждение.

Следствие 6.5. Если $\mathcal{R} \subset \mathcal{D}$ — подалгебра алгебры отношений и R_i, при $i \in I$, — ее образующие, то любое отношение $R \in \mathcal{R}$ симметрично относительно всякого $h \in S\{R_i\}_{i \in I}$.

Утверждение 6.6. Для произвольной реляционной подалгебры

$\mathcal{R} \subset \mathcal{D}$ алгебры отношений группа H утверждения 3.2 совпадает с группой симметрии $S(\mathcal{R}) = S\{R_i\}_{i\in I}$, где $R_i,\ i \in I$, — образующие подалгебры $\mathcal{R}$.

Д о к а з а т е л ь с т в о. Докажем, что $S(\mathcal{R}) \subset H$. Так как $H \in \mathcal{R}$, то отношение H, согласно следствию 6.5, симметрично относительно всех $h \in S(\mathcal{R})$, то есть $H = h \circ H$ для всех $h \in S(\mathcal{R})$. Так как H содержит тождественное отображение $id(D)$, то отсюда следует, что $h \in H$ для всех $h \in S(\mathcal{R})$, то есть $S(\mathcal{R}) \subset H$. Перейдем к доказательству включения $H \subset S(\mathcal{R})$. Пусть $h \in H$ — произвольный элемент. Так как H — группа относительно операции композиции, то очевидно, что $H = h \circ H$, то есть отношение H симметрично относительно h. Согласно утверждению 6.3 любое отношение $R \in \mathcal{R}$ получается из H с помощью операций φ^* и объединений. Отсюда, согласно утверждению 6.4, отношение R симметрично относительно $h \in H$. Так как отношение $R \in \mathcal{R}$ — любое, то $h \in S(\mathcal{R})$, что завершает доказательство включения $H \subset S(\mathcal{R})$.

Т е о р е м а 6.7. Реляционные подалгебры $\mathcal{R} \subset \mathcal{D}$ алгебры отношений $\mathcal{D}$ над категорией наборов атрибутов $\bar{\mathcal{A}}$ (без ограничения их конечности) находятся во взаимно однозначном соответствии с подгруппами $H \subset S(D)$ группы всех перестановок элементов области значений атрибутов D, сохраняющих тип элементов. Подалгебре $\mathcal{R}$ сопоставляется подгруппа H всех перестановок, относительно которых симметрично каждое отношение подалгебры $\mathcal{R}$. Подгруппе H сопоставляется подалгебра R, состоящая из всех отношений симметричных относительно элементов группы H.

Д о к а з а т е л ь с т в о. Пусть $\mathcal{R} \subset \mathcal{D}$ — произвольная подалгебра. Согласно утверждению 6.2, ей соответствует подгруппа $H \subset S(D)$. Так как, согласно утверждению 6.3, подгруппа H, рассмотренная как отношение $H \in \mathcal{R}(D)$, является атомарной образующей подалгебры $\mathcal{R}$, то разным подалгебрам соответствуют разные подгруппы. Пусть $H \subset S(D)$ — произвольная подгруппа. Ей соответствует подалгебра $\mathcal{R}$, порожденная отношением $H \in \mathcal{D}(D)$. Очевидно, что у такой подалгебры группа симметрии $S(\mathcal{R}) = S\{H\} = H$ и, согласно утверждению 6.6, H является атомарным элементом построенной подалгебры $\mathcal{R}$. Поэтому разным подгруппам соответствуют разные подалгебры. Второе и третье утверждение теоремы следуют из утверждений 6.6 и 6.5.

З а м е ч а н и е 6.1. Из доказательства теоремы 6.7 видно, что требование существования отношений с наборами атрибутов произвольной мощности можно ослабить требованием существования отношений с наборами атрибутов мощности, не превышающими мощность множества D.

Т е о р е м а 6.8. Пусть $\mathcal{R} \subset \mathcal{D}$ и $\mathcal{R}' \subset \mathcal{D}'$ — полные подалгебры алгебр отношений с областями значений атрибутов, соответственно, D и D', и $j : \mathcal{R} \to \mathcal{R}'$ — изоморфизм реляционных алгебр. Тогда найдется взаимно однозначное отображение $g : D \to D'$ области D на область D', сохраняющее типы элементов, такое, что $H = g^{-1} \circ H' \circ g$, где H и H' — подгруппы, соответствующие подалгебрам $\mathcal{R}$ и $\mathcal{R}'$. В этом случае для любого отношения $R \in \mathcal{R}$ соответствующее ему отношение $j(R) \in \mathcal{R}'$ имеет вид: $j(R) = g \circ R$.

Д о к а з а т е л ь с т в о. Пусть $H \in \mathcal{R}$ и $H' \in \mathcal{R}'$ — группы, определенные утверждением 6.2. Рассмотрим отношение $j(H)$. Так как j — изоморфизм реляционных алгебр, то он переводит атомарный элемент $H \in \mathcal{R}(D)$ в некоторый атомарный элемент $j(H)$ булевой алгебры $\mathcal{R}'(D)$. Согласно утверждению 6.3, тогда найдется такое отображение $g : D \to D$, что $j(H) = g^*(H')$. Аналогичными рассуждениями найдется такое отображение $g_1 : D' \to D$, что $j^{-1}(H') = g_1^*(H)$, где j^{-1} — гомоморфизм алгебры $\mathcal{R}'$ в $\mathcal{R}$, обратный гомоморфизму j. Покажем, сначала, что g — взаимно однозначное отображение. Для этого рассмотрим равенство $H = j^{-1}(j(H)) = j^{-1}(g^*(H'))$. Так как j^{-1} — гомоморфизм реляционной алгебры, то, по определению гомоморфизма, он коммутирует с операцией g^*, и последнее равенство принимает вид $H = g^*(j^{-1}(H'))$. Учитывая, что $j^{-1}(H') = g_1^*(H)$, получим равенство $H = g^*(g_1^*(H))$, из которого, по определению операций g^* и g_1^*, следует, что $H = (H \circ g_1) \circ g$. В частности, имеем равенство: $id(D) = (h \circ g_1) \circ g$, то есть отображение g имеет обратное слева. Если аналогичным образом рассмотреть равенство $H' = j(j^{-1}(H'))$, то получим равенство $H' = (H' \circ g) \circ g_1$, из которого следует равенство $h_1 = (id(D') \circ g) \circ g_1$, где $h_1 \in H'$ и, следовательно, обратим. Умножим обе части последнего равенства на h_1^{-1} справа. Получим, $id(D') = g \circ (g_1 \circ h_1^{-1})$, то есть отображение g имеет обратное справа. Отсюда стандартным образом следует, что g — взаимно однозначное отображение.

Для доказательства включения $g^{-1} \circ H' \circ g \subset H$ рассмотрим выражение $(g^{-1} \circ h_2 \circ g)^*(g^*(H'))$, где h_2 — произвольный элемент группы H'. По определению операций $(g^{-1} \circ h_2 \circ g)^*$ и g^*, имеем: $(g^{-1} \circ h_2 \circ g)^*(g^*(H')) = H' \circ g \circ (g^{-1} \circ h_2 \circ g) = H' \circ g = g^*(H')$, где второе равенство следует из ассоциативности операции композиции и равенств $g \circ g^{-1} = id(D)$, $H' \circ h_2 = H'$. Итак, доказано, что операция $(g^{-1} \circ h_2 \circ g)^*$ оставляет отношение $g^*(H') = j(H)$ неизменным. Так как j — изоморфизм, это значит, что операция $(g^{-1} \circ h_2 \circ g)^*$ оставляет неизменным и отношение

H. То есть $(g^{-1} \circ h_2 \circ g)^*(H) = H$ или, по определению действия операции, $H \circ (g^{-1} \circ h_2 \circ g) = H$. Так как группа H содержит тождественное отображение, то последнее равенство возможно тогда и только тогда, когда $g^{-1} \circ h_2 \circ g \in H$. Учитывая, что h_2 — произвольный элемент группы H', получаем включение $g^{-1} \circ H' \circ g \subset H$.

Для доказательства включения $g^{-1} \circ H' \circ g \supset H$ проводятся аналогичные рассуждения, где вместо отображения g рассматривается отображение $g_1 : D' \to D$. В результате получается включение $g_1^{-1} \circ H \circ g_1 \subset H'$. Ранее мы получили равенство $id(D) = (h \circ g_1) \circ g$, где $h \in H$. Отсюда выражается $g_1 = h^{-1} \circ g^{-1}$. Подставив это выражение в последнее полученное включение, получим соотношение $g \circ h \circ H \circ h^{-1} \circ g^{-1} \subset H'$. Умножим это соотношение слева на g^{-1} и справа на g. Учитывая, что h принадлежит H, получим требуемое включение.

Из доказанных двух соотношений $g^{-1} \circ H' \circ g \subset H$ и $g^{-1} \circ H' \circ g \supset H$ следует требуемое равенство $g^{-1} \circ H' \circ g = H$.

Последнее утверждение теоремы следует из того, что сопоставление $R \mapsto g \circ R$ для любого отношения $R \in \mathcal{R}$ является, очевидным образом, гомоморфизмом реляционных алгебр, когда $g : D \to D'$ — взаимно однозначное отображение. Так как этот гомоморфизм и гомоморфизм j, переводят образующую H алгебры $\mathcal{R}$ в один и тот же элемент $j(H) = g^*(H') = (g \circ H \circ g^{-1}) \circ g = g \circ H$, то они совпадают.

Перейдем к изучению подалгебр $\mathcal{R} \subset \mathcal{D}$ алгебры отношений над категорией $\mathcal{A}_T$ конечных наборов атрибутов. При этом область значений атрибутов $D = \coprod\limits_{t \in T} D_t$ может быть бесконечной.

Так как категория $\mathcal{A}_T$ является подкатегорией категории всех множеств $\bar{\mathcal{A}}_T$, типизированных множеством T, то естественно начать изучение подалгебры $\mathcal{R}$ с изучения наименьшей подалгебры $\bar{\mathcal{R}} \subset \bar{\mathcal{D}}$, содержащей элементы $\mathcal{R}$, где $\bar{\mathcal{D}}$ — алгебра всех отношений с произвольными наборами атрибутов. Алгебра $\bar{\mathcal{R}}$ называется пополнением алгебры $\mathcal{R}$ за счет операций над отношениями с бесконечными наборами атрибутов. Для любой подалгебры $\bar{\mathcal{R}} \subset \bar{\mathcal{D}}$ такое пополнение $\bar{\mathcal{R}}$ существует, как пересечение всех подалгебр в $\bar{\mathcal{D}}$, содержащих $\mathcal{R}$.

Интересным и важным вопросом является вопрос о том, что при таком пополнении происходит с множествами $\mathcal{R}(A)$ для конечных наборов атрибутов. Очевидно, что $\mathcal{R}(A) \subset \bar{\mathcal{R}}(A)$. Более того, кажется естественным, что $\mathcal{R}(A) = \bar{\mathcal{R}}(A)$. Однако, автору удалось доказать это равенство лишь при условии счетности или конечности области значений атрибутов D.

Пусть $\mathcal{R} \subset \mathcal{D}$ — произвольная подалгебра реляционной алгебры отношений с конечными наборами атрибутов. Для любого конечного подмножества $e_L : L \subset D$ области значений атрибутов рассмотрим полную булеву алгебру $\mathcal{R}(L)$, которая по определению является подалгеброй булевой алгебры всех подмножеств множества $\bar{\mathcal{A}}(L, D)$. Обозначим через $H(L)$ наименьший элемент булевой алгебры $\mathcal{R}(L)$, содержащий отображение $e_L : L \to D$.

Система отображений $\mathcal{H} = \bigcup\limits_{L \subset D} H(L)$, где L — любое конечное подмножество D, является аналогом группы H утверждения 6.2 для подалгебр отношений с конечными наборами атрибутов. Если отображения h_1 и h_2 принадлежат $\mathcal{H}$ и образ отображения h_2 содержится в области определения отображения h_1, то через $h_1 \circ h_2$ обозначается композиция отображений.

У т в е р ж д е н и е 6.9. Система отображений $\mathcal{H} = \bigcup\limits_{L \subset D} H(L)$ обладает следующими свойствами:

для любого конечного подмножества $e_L : L \subset D$ множество $H(L)$ является подмножеством множества $\bar{\mathcal{A}}_T(L, D)$, содержащим отображение e_L;

если $h_1 : L_1 \to L_2 \subset D$ — элемент множества $H(L_1)$, то $h_1^*(H(L_2)) = H(L_2) \circ h_1 = H(L_1)$.

Д о к а з а т е л ь с т в о. Первое свойство, сформулированное в утверждении, следует из определения $H(L)$. Для доказательства второго второго свойства заметим, что по построению $H(L_i)$ — это атомы булевых алгебр $\mathcal{R}(L_i)$. С другой стороны, операция h_1^* переводит атомы булевой алгебры $\mathcal{R}(L_2)$ в атомы булевой алгебры $\mathcal{R}(L_1)$. Следовательно, $h_1^*(H(L_2))$ и $H(L_1)$ — атомы булевой алгебры $\mathcal{R}(L_1)$, но они имеют общий элемент h_1, а потому совпадают.

С л е д с т в и е 6.10. Все отображения из системы отображений $\mathcal{H}$ являются вложениями. Более того, если $h \in H(L)$ и $L_1 \subset D$ — образ отображения h, то найдется отображение $h_1 \in H(L_1)$ такое, что композиция $h_1 \circ h = e_L$. Для всякого отображения $h \in H(L)$ и любого конечного подмножества $L_2 \subset D$, содержащего L, существует отображение $h_2 \in H(L_2)$, являющееся продолжением отображения h. Если h' и h'' — произвольные элементы $\mathcal{H}$ такие, что композиция $h'' \circ h'$ определена, то $h'' \circ h'$ принадлежит системе отображений $\mathcal{H}$.

Доказательство следствия вытекает из первого и второго свойств системы отображений $\mathcal{H}$, сформулированных в утверждении 6.9.

Следующее утверждение является аналогом утверждения 6.3 для реляционных алгебр с конечными наборами атрибутов.

Утверждение 6.11. Для любого атомарного элемента $R \in \mathcal{R}(A)$ реляционной подалгебры $\mathcal{R} \subset \mathcal{D}$ найдется такое сюръективное согласование наборов атрибутов $\varphi : A \to L$, что $R = \varphi^*(H(L))$.

Доказательство. Рассмотрим отношение R как подмножество в множестве согласований $\bar{\mathcal{A}}_T(A, D)$. Пусть $\varphi \in R$ и L — образ множества A при отображении φ. Рассмотрим отношение $\varphi^*(H(L))$. По определению операции φ^* это отношение имеет вид $H(L) \circ \varphi$. Так как отношение $H(L)$ содержит элемент e_L, то отсюда следует, что $\varphi \in \varphi^*(H(L))$. По определению, $H(L)$ — атом булевой алгебры $\mathcal{R}(L)$. Так как операция φ^* переводит атомы в атомы и отношения R и $\varphi^*(H(L))$ имеют общий элемент φ, то $R = \varphi^*(H(L))$.

Утверждения 6.9, 6.10, 6.11 показывают в каком смысле система отображений $\mathcal{H} = \bigcup\limits_{L \subset D} H(L)$ является аналогом группы утверждения 6.2. Более того, не составляет труда сформулировать и доказать теорему, аналогичную теореме 6.7 о взаимно однозначном соответствии между полными подалгебрами $\mathcal{R}$ алгебры отношений $\mathcal{D}$ над конечными наборами атрибутов и системами отображений $\mathcal{H}$, обладающими свойствами, сформулированными в следствии 6.10. Очевидно также понятие отношения симметричное относительно системы отображений $\mathcal{H}$.

Перейдем теперь к случаю, в котором удалось доказать более сильное утверждение о соответствии подалгебр $\mathcal{R}$ алгебры отношений $\mathcal{D}$ над конечными наборами атрибутов и их групп симметрий.

Пусть $\mathcal{R} \subset \mathcal{D}$, как и прежде произвольная реляционная подалгебра алгебры отношений с конечными наборами атрибутов; $\mathcal{H} = \bigcup\limits_{L \subset D} H(L)$ — система отображений утверждения 6.9, построенная по алгебре отношений $\mathcal{R}$. Пусть, кроме того, $\bar{\mathcal{R}} \subset \bar{\mathcal{D}}$ — пополнение подалгебры $\mathcal{R}$ за счет операций над отношениями с бесконечными наборами атрибутов, и $\bar{H}$ — группа симметрии подалгебры $\bar{\mathcal{R}}$, определенная в теореме 6.7.

Сформулируем, сначала, утверждение, дающее выражение группы симметрии $\bar{H}$ через систему отображений $\mathcal{H}$.

Утверждение 6.12. Взаимно однозначное отображение $\bar{h} : D \to D$ тогда и только тогда принадлежит группе симметрии $\bar{H}$ подалгебры отношений, когда ограничение отображения $\bar{h}$ на любое конечное подмножество

$L \subset D$ принадлежит $H(L)$.

Д о к а з а т е л ь с т в о. Рассмотрим группу симметрии $\bar{H}$ подалгебры $\bar{\mathcal{R}} \subset \bar{\mathcal{D}}$. Так как, согласно утверждению 6.11, отношения $H(L)$, где L — конечные подмножества D, являются образующими подалгебры отношений $\mathcal{R} \subset \mathcal{D}$, а, следовательно, и пополненной подалгебры $\bar{\mathcal{R}}$, то по утверждению 6.6 взаимно однозначное отображение $\bar{h} : D \to D$ является элементом $\bar{H}$ тогда и только тогда, когда $\bar{h}$ является симметрией отношений $H(L)$ для всех конечных подмножеств $L \subset D$. Рассмотрим действие симметрии $\bar{h}$ на отношение $H(L)$. По определению симметрии $\bar{h} \in \bar{H}$ тогда и только тогда, когда $\bar{h} \circ H(L) = H(L)$. В частности, $\bar{h} \circ e_L \in H(L)$, то есть ограничение отображения $\bar{h}$ на L принадлежит $H(L)$ для любого конечного подмножества $L \subset D$. Наоборот, если ограничение взаимно однозначного отображения $\bar{h} : D \to D$ на любое конечное подмножество принадлежит $\mathcal{H}$, то $\bar{h} \circ H(L) = H(L)$, так как композиция отображений из $\mathcal{H}$ принадлежит $\mathcal{H}$.

В случае счетности области значений атрибутов D выражение системы отображений $\mathcal{H}$ через группу симметрии $\bar{H}$ дается следующим утверждением.

У т в е р ж д е н и е 6.13. Если мощность области значений атрибутов D счетна, то выполняются равенства $H(L) = e_L^*(\bar{H})$ для всех конечных подмножеств $e_L : L \subset D$. То есть отображения из системы $\mathcal{H}$ являются ограничениями элементов группы $\bar{H} \subset \bar{\mathcal{A}}(D, D)$ на конечные подмножества $L \subset D$.

Доказательство утверждения 6.13 основано на следующей лемме.

Л е м м а. Для любого отображения h, принадлежащего $\mathcal{H}$, и любого элемента $d \in D$ существует такое продолжение h' отображения h, также принадлежащее $\mathcal{H}$, что элемент d принадлежит области определения и образу отображения h'.

Д о к а з а т е л ь с т в о л е м м ы. Пусть отображение $h : L \to D$ принадлежит $H(L)$. Согласно следствию 6.10, всякое такое отображение продолжается до отображения $h_1 : L \cup \{d\} \to D$, принадлежащего $H(L \cup \{d\})$. Это же следствие утверждает, что обратное отображение $h_1^{-1} : L_1 \to D$, где L_1 — образ отображения h_1, принадлежит $H(L_1)$. Рассмотрим продолжение $h_2 : L_1 \cup \{d\} \to D$ отображения h_1^{-1} на элемент d, принадлежащее $H(L_1 \cup \{d\})$. Нетрудно видеть, что требуемым в лемме продолжение h' отображения h является отображение h_2^{-1}, обратное к отображению h_2.

Д о к а з а т е л ь с т в о у т в е р ж д е н и я 6.13. Перенумеруем элементы области значений атрибутов D. Пусть d_n — n-ый элемент области D. Для любого отображения $h \in H(L)$ построим последовательность продолжающих друг друга отображений h_n из системы отображений $\mathcal{H}$. При этом потребуем, чтобы $h_0 = h$ и элемент d_n принадлежал области определения и образу отображения h_n. Согласно лемме такая последовательность может быть индуктивно построена. Рассмотрим отображение $\bar{h} : D \to D$, которое на области определения h_n совпадает с h_n для каждого n. Отображение $\bar{h}$ по построению продолжает отображение $h \in H(L)$ и является взаимно однозначным, так как отображения h_n — вложения и для любого элемента $d_n \in D$ определен образ $\bar{h}(d_n) = h_n(d_n)$ и прообраз $\bar{h}^{-1}(d_n) = h_n^{-1}(d_n)$.

Введем в множестве всех обратимых отображений $S(D)$ области значений атрибутов на себя следующую топологию:

для любого подмножества $K \subset S(D)$ обратимое отображение $\bar{h} \in S(D)$ принадлежит замыканию $\bar{K}$ тогда и только тогда, когда ограничение отображения $\bar{h}$ на любое конечное подмножество $e_L : L \subset D$ принадлежит множеству отображений $K \circ e_L$.

Т е о р е м а 6.14. Реляционные подалгебры $\mathcal{R} \subset \mathcal{D}$ алгебры отношений с конечными наборами атрибутов и счетной или конечной областью значений атрибутов D находятся во взаимно однозначном соответствии с замкнутыми подгруппами $\bar{H} \subset S(D)$ группы $S(D)$ в топологии, определенной выше. Подгруппа $\bar{H}$ является группой симметрии отношений подалгебры $\mathcal{R}$.

Д о к а з а т е л ь с т в о. Замкнутость подгруппы симметрии $\bar{H} \subset S(D)$ пополненной подалгебры отношений $\bar{\mathcal{R}}$ следует из утверждения 6.12. Кроме того, из утверждений 6.12 и 6.13 следует взаимно однозначное соответствие между замкнутыми подгруппами $\bar{H}$ и системами отображений $\mathcal{H}$ и, следовательно, между замкнутыми подгруппами $\bar{H}$ и реляционными подалгебрами $\mathcal{R}$ алгебры отношений $\mathcal{D}$.

Литература

[1] Агафонов В. Н. *Спецификация программ: понятийные средства и их организация.* Новосибирск: Наука,Сиб.отд-ние,1990.

[2] Агафонов В. Н. *Типы и абстракция данных в языках программирования.* В кн. Данные в языках программирования. М.:Мир,1982, с.263–327.

[3] Бениаминов Е. М. *Алгебраический подход к моделям баз данных реляционного типа.* В кн.:Семиотика и информатика, 1980, вып.14, с.44-80.

[4] Бениаминов Е. М. *Алгебраическая стуктура реляционных моделей баз данных.* НТИ, сер.2, 1980, N9, с. 23-25.

[5] Бениаминов Е. М., Березина Н. А. *Об алгебраическом подходе к описанию схем баз данных.*// В сб. Вопросы создания Автоматизированной системы НТИ по документам ГАФ СССР, Москва: ГАУ при Совете Министров СССР, ВНИИДАД, 1981, с.69-77.

[6] Бениаминов Е. М. *О роли симметрии в реляционных моделях баз данных и логических структурах.* НТИ, сер.2, 1984, N5, с.17-25.

[7] Бениаминов Е. М., Березина Н. А., Дунская М. В. *Разработка методов моделирования автоматизированной обработки, поиска и размещения данных в больших информационных системах.* СИФ ОЦНТИ, ВНИИДАД, депонированная рукопись, инв. N031–85,М.,1985.

[8] Бениаминов Е. М. *О некотором подходе к представлению знаний.* Тезисы докладов 4 Всесоюзн.конференции "Применение методов мат. логики" г. Таллин, 1986, с.34-36.

[9] Бениаминов Е. М. *Основания категорного подхода к представлению знаний. Категорные средства.* Изв. АН СССР Техн. кибернет.,N 2, 1988 , с.21–33.

[10] Бениаминов Е. М. *Рефлексивные топосы в категорном подходе к представлению знаний.* Тезисы докладов Всесоюзн. школы-семинара

"Семиотические аспекты формализации интеллектуальной деятельности" г. Боржоми, 1988, с.111-113.

[11] Бениаминов Е.М., Вайнтроб А. Ю. *Основные принципы диалогового языка для представления знаний средствами категорного подхода.* Материалы конференции ДИАЛОГ-87, г. Тбилиси, 1988, с.174-177.

[12] Бениаминов Е. М. *Алгебраические системы и типы данных.* //В кн.:Системное и теоретическое программирование, Ростов-н-Д: РГУ, 1988, с.83-92.

[13] Бениаминов Е. М. *Система представления и обработки понятий, основанная на алгебраическом (категорном) подходе.* Труды II Всесоюзной конференции "Искусственный интеллект- 90", т.2, 1990, с.8-11.

[14] Вигнер П. *Программирование на языке АДА.* М.:Мир, 1983.

[15] Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л. *Алгебра, логика, языки, программирование.* Киев:Наукова думка, 1974.

[16] Голдблатт Р. *Топосы. Категорный анализ логики.* М.:Мир, 1983.

[17] Гончаров С. С., Ершов Ю. Л., Свириденко Д. И. *Семантические основы логического программирования*// В сб. Концептуализация и смысл, под ред. Полякова И. В., Новосибирск:Наука, 1990, с.6-20.

[18] Джонстон П. Т. *Теория топосов.* М.:Наука, 1986.

[19] Жожикашвили А.В., Стефанюк В.Л. *Теория категорий в задачах представления знаний и обучения.* Изв. АН СССР. Техн. кибернет., N 2, 1986.

[20] Замулин А. В. *Системы программирования баз данных и знаний.* Новосибирск:Наука, 1990.

[21] Калиниченко Л. А. *Методы и средства интеграции неоднородных баз данных.* М.:Наука, 1983.

[22] Калиниченко Л. А., Рывкин В. М. *Машины баз данных и знаний.* М.:Наука, 1990.

[23] Капитонова Ю. В., Летичевский А. А. *Математическая теория проектирования вычислительных систем.* М.:Наука, 1988.

[24] Кондрашина Е. Ю., Литвинцева Л. В., Поспелов Д. А. *Представление знаний о времени и пространстве в интеллектуальных системах.* М.:Наука, 1989.

[25] Кучеров Г. А. *Системы подстановок термов.* Препринт 601, ВЦ АН СССР Сиб. отделение, Новосибирск, 1985.

[26] Мальцев А. И. *К общей теории алгебраических систем.* Мат. сборник, 1954, т.35, вып.1.

[27] Массер Д. *Спецификация абстрактных типов данных в системе AFFIRM.*// В сб.: Требования и спецификации в разработке программ, М.:Мир, 1984, с.199-222.

[28] Плоткин Б. И. *Универсальная алгебра, алгебраическая логика и базы данных.* М.:Наука, 1991.

[29] Поспелов Д. А. *Логико-лингвистические модели в системах управления.* М.:Энергоиздат, 1981.

[30] Свириденко Д. И. *Проект Сигма. Цели и задачи.* // В сб. Логические методы в программировании под ред. Ершова Ю. Л. (Вычислительные системы, вып. 133), РАН, Сиб. отд.-ние, Ин.-т математики, Новосибирск, 1990, с.68-94.

[31] Цаленко М. Ш. *Моделирование семантики в базах данных.* М.:Наука, 1989.

[32] Шенфилд Дж. *Математическая логика.* М.:Наука, 1975.

[33] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 1)* Comput. Lingust. Comput. Lang., 13, 1979, p. 152-282.

[34] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 2)* Comput. Lingust. Comput. Lang., 14, 1980, p. 43-65.

[35] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 3)* Lect. Notes in Comp. Sci., V.117, Springer-Verlag, Berlin, 1981, p. 281-290.

[36] Banchilhon F. *On the Compliteness of Query Language for Relational Data Bases.* Lect. Notes in Comp. Sci., V.64, Springer-Verlag, 1978, pp.76-98.

[37] Beniaminov E. M. *Concept Bases and Algebraic Modeling Methods.* Proceedings of the International Workshop on Advances in Databases and Information Systems (ADBIS'94), 1994, p.133-135.

[38] Beniaminov E. M. *A Categorical Approach to Knowledge Representation.* Japan-CIS Symposium on Knowledge Based Software Engineering'94 (JCKBSE'94), 1994, p.181-182.

[39] Beniaminov E. M. *Algebraic Invariants of Database Schemes.* Proceedings of the Second International Workshop on Advances in Databases and Information Systems (ADBIS'95), V.1, 1995, p.259-263.

[40] Breu R. *Algebraic Specification Techniques in OOP Environments.* Lect. Notes in Comp. Sci., V.562, Springer-Verlag, 1991.

[41] Burstall R. M., Goguen J. A. *The Semantics of Clear, a Specification Language* In: Proceedings, of the 1979 Copenhagen Winter Shcool on Abstrakt Software Specification, Springer-Verlad, Berlin, 1980, 292-332.

[42] Chandra A. K., Harel D. *Computable queries for relational data bases.* J. Comput. & Syst. Sci., V.21, N2, 1980, p.156-172.

[43] Codd E. F. *A relational model for large shared data banks.* Comm. of ACM 13, 6, 1970, p.377-387.

[44] Dershowitz N. *Orderings for term-rewriting systems.* Theor. Comput. Sci., 1982, V.17, 3, pp.279-302.

[45] Elmasri R., Weeldrever J., Hevner A. *The category concept: an extension to the entity-relatconship model* Data & Knowledge Engineering, V.1, N1,1985.

[46] Ehrig H., Mahr B. *Fundamentals of Algebraic Specification.* V.1, Springer-Verlag, 1985.

[47] Ehrig H., Mahr B. *Fundamentals of Algebraic Specification.* V.2, Springer-Verlag, 1990.

[48] *First International Symposium on Category Theory Applied to Computation and Control* Lect. Notes in Com. Sci, V.25, Springer - Verlad, Berlin, 1975.

[49] Georgescu I. *A Categorial approach to knowledge-based systems.* Computers and Artifical Intelligence, V.3, N2, 1984, pp.105-113.

[50] Goguen J. A., Thatchar J. W., Wagner E. G. *An inititial algebra approach to the specification, correctness and implementation of abstract data types.* In Current Trends in Programming Methodology IV: Data Structuring, Prentice Hall, 1978, pp.80-144.

[51] Goguen J. A. *Some design principles and theory for OBJ-O, a language to express and execute algebraic specifications of problems.* In: Lect. Notes Comput. Sci.,V.75, 1979, pp.425-473.

[52] Goguen J. A., Burstall R. M. *Some fundamental algebraic tools for the semantics of computation.* Theoretical Computer Science, V.31, N2, 3, 1984.

[53] Goguen J. A., Burstall R. M. *Introducing Institution.* Lect. Not. Comp.Sci., V.164, 1984.

[54] Goguen J. A., Meseguer J. *Equality, types, modules, and (why not?) generics for logic programming.* Conf. of Logical Prog., Uppsala, Sweden, 1984, 179-210.

[55] Goguen J. A., Meseguer J. *Eqlog: Equality, types, and generic modules for logic programming.* In Douglas SeGroot and Gary Lindstrom, eds. "Logic Programming: Functions, Relations and Equations", Prentice-Hall, Englewood Cliffs, N.J., 1986, pp.295-363.

[56] Goguen J. A., Meseguer J. *Order-sorted algebra solves the constructor selector, multiple representation and coercion problems.* Symposium on Logic in Comp. Sci., IEEE Comp. Society Press, 1987, pp.18-29.

[57] Goguen J. A., Burstall R. M. *Institutions: Abstract Model Theory for Specification and Programming.* Jorn. of ACM, V.39, 1, 1992, pp.95-146.

[58] Grothendieck A., Verdier J. L. *Théorie des Topos.* (SGA 4, exposés I-VI).–Second edition.–Berlin; Heidelberg; N. Y.:Springer, 1972.

[59] Huet G. *Confluent reductions: abstract properties and applications to term rewriting systems.* Journ. of ACM, 1980, v.27, N4, pp.797-821.

[60] Kaplan S. *Symplifying conditional term rewriting systems: unification, termination and confluence.* Jorn. Symbolic Computation 4(3), 1987, pp.295-334.

[61] Knuth D., Bendix P. *Simple word problems in universal algebras.* In: Conputational Problems in Abstract Algebra, Pergamon Press, Elmsford, New York, 1970, pp.263-297.

[62] Krasner M. I. *Generalization et analogues de la theorie de Galois.* Comptes Rendus de Congress de la Victorie de I'Ass. Franc. pour I'Avancem. Sci., 1945, pp. 54-58.

[63] Lawvere F. W. *Functorial semantics of algebraic theories.* Proc. Nat. Acad. Sci., 1963, V.50, N5, pp.869-872.

[64] Lawvere F. W. *Introduction* In: Toposes, Algebraic Geometry and Logic, Lect. Notes in Math., V.174, 1972.

[65] Maltsev A. *Algebraic Systems.* Springer-Verlag, 1973.

[66] Melton A., Schmidt D., and Strecher G. *Galois connections and computer science applications.* Lect. Notes in Comp. Sci., V. 240, Springer-Verlag, 1986.

[67] *Proceedings, Category Theory and Computer Programming,* Lect. Notes in Comp. Sci., V. 240, Springer-Verlag, 1986.

[68] *Proceedings, Category Theory and Computer Science,* Lect. Notes in Comp. Sci. V. 283, Springer-Verlag, 1987.

[69] Rydeheard D.F., Burstall R.M. *Computational category theory.* Prentice Hall, 1988.

[70] Tuijn C., Gyssens M., Paredaens J. *A Categorical Approach to Object-Oriented Data Modelling.* Proceedings of Third Workshop on Foundation of Models and Languages for Data and Objects, Aigen, 1991, pp.187–196.

[71] Rusinowitch M., Rémy J. L. (Eds.) *Conditional Term Rewriting Systems.* Lect. Notes in Comp. Sci., V. 656, Springer-Verlag, 1993.

[72] Zilles S.N. *Introduction to data algebras.* Lect. Notes Comput. Sci., V.86, Springer-Verlag, 1980.

Глава 2. Алгебраический подход к представлению понятий

1 Понятия как объекты моделирования, примеры

1.1 Основные характеристики баз понятий

Тема этой главы — некоторые математические средства представления понятий для автоматизированной работы с ними.

В этой главе, следуя [40, 46, 54, 70, 9], рассматривается алгебраический подход к моделированию понятий и их представлений.

Алгебраические методы лежат в основе построения реляционных моделей баз данных и абстрактных типов данных в программировании. В диссертации опыт применения алгебры в этих областях используется для приложения к более широкой задаче: моделированию понятий и формированию баз понятий.

Базы понятий — это новая складывающаяся область в программировании, связанная с использованием и построением больших библиотек типов объектов в объектно-ориентированных языках программирования, а также с использованием CASE–технологий и построением специализированных программных систем типа TOOLKIT, элементы которых предназначены для многоразового многоцелевого использования в различных приложениях.

Практика и опыт использования растущих библиотек подобного рода требует некоторой стандартизации в их организации и формах обращения к ним, которая была бы основана на ясной модели отдельных понятий и базы понятий в целом.

Для начала сформулируем несколько свойств, выделяющих базы понятий среди других объектов программирования.

1. База понятий — это большая совокупность (сотни — тысячи) отдельных понятий (типов объектов, абстрактных типов данных, фрагментов схем баз данных, баз данных и т.д.).

2. Понятия из базы понятий предназначены для многоразового многоцелевого использования в приложениях, в постановках задач, в формулировках запросов.

3. Система понятий должна быть открыта для расширения и модификации пользователями (введение новых понятий, версий понятий, уничтожение неиспользуемых понятий).

4. Понятия в базе понятий взаимосвязаны по отношениям:

   (a) использования (одно понятие может использоваться в другом);

   (b) конкретизации (понятие может быть уточнением более абстрактного другого понятия);

   (c) реализации (одно понятие может иметь несколько реализаций).

5. Работа с базой понятий состоит из интерактивного процесса формирования новых понятий (задач) с использованием понятий из базы понятий, постановок вопросов в среде нового понятия и получения ответов с последующим возможным использованием понятия в приложениях.

6. Использование понятий должно обеспечивать открытость языка работы с понятиями и их элементами, обеспечивать пользователю возможность расширения языка языковыми конструкциями прикладной области.

7. Должна быть обеспечена принципиальная совместимость различных баз понятий — возможность использования понятий из нескольких специализированных баз понятий.

Состояние работ и исследований по базам понятий очень напоминает ситуацию с базами данных в конце 60-х начале 70-х годов, когда была осознана необходимость отделения процессов ведения данных от их использования, были созданы файловые системы, но не было ясной концепции баз данных. Для формулировки одной из таких концепций был успешно использован алгебраический язык в работах Е. Кодда.

Цель этой главы показать, что алгебраические методы, с учетом опыта их применения в теории баз данных и абстрактных типов данных, могут оказаться полезными и в моделировании баз понятий.

Представление данных в виде алгебраических структур является естественным для многих специалистов в области программирования. Начиная с известной работы Е.Кодда [43], применение алгебраических методов распространилось на моделирование баз данных. Объектом изучения стали алгебраические системы с операциями над отношениями — реляционные алгебры. В настоящее время реляционные алгебры хорошо изучены в работах зарубежных и отечественных специалистов, включая работы автора.

Следующим шагом явилось изучение алгебраическими методами схем баз данных, расширение систем операций операциями над типами данных и над схемами. Это направление привело к построению алгебраических моделей систем представления знаний, в которых существенна неполнота знаний. Среди алгебраических средств в таких моделях особую роль играют понятия и методы теории категорий [16].

Основное, что используется из алгебры в приложениях,— это алгебраический язык. Следующее — это алгебраический стиль мышления: использование понятий изоморфизма и гомоморфизма алгебраических систем, задание алгебраических систем с помощью образующих и соотношений. Дальнейшее использование алгебры приводит к использованию специфических понятий и результатов, введенных и исследованных в алгебре математиками. К ним относятся специфические операции и алгебраические системы, теоремы о разрешимости в алгебрах, специфические алгоритмы.

Результатами алгебраического исследования, помимо построения языка, на котором могут ставиться вопросы в области теории баз данных и понятий, могут быть теоремы классификации построенных алгебр и, в идеале, алгоритмы для распознавания тождеств, изоморфизмов, гомоморфизмов систем и решений уравнений, которые могут быть использованы в системах проектирования, алгоритмах оптимизации запросов, в алгоритмах получения ответов на запросы.

Кроме того, как и в базах данных, в базах понятий существенный вопрос — это полнота использованной системы операций, который в некоторой постановке тоже может быть сведен к алгебраической задаче.

1.2 Типы данных как примеры понятий

Начнем с примеров понятий, которые обычно представляюся в развитых системах программирования, — это типы данных, абстрактные типы данных, типы объектов.

В настоящее время имеется большое количество статей и монографий по типам данных в программировании, по абстрактным типам данных, по применению современных математических методов к моделированию типов данных в программировании, по языкам спецификаций типов данных [20, 1, 46, 25, 12, 7] . В этой области сложились общее понимание задач, некоторая система понятий, выработался язык и некоторые методы, то есть произошло становление области науки в информатике, которое впитало в себя некоторые достижения опыта, методов и представлений в программировании, алгебре и логике.

Основная задача введения развитого механизма типов данных в программирование — это программная поддержка общения человека с машиной в понятиях, привычных и удобных специалисту во время структурирования программы, программирования, отладки программы и ее модификации.

Понятие типа данных используется в том или ином объёме во всех языках программирования "высокого уровня". Как отмечает В.Н.Агафонов в своем обзоре [2], введение типов данных в программирование, по всей вероятности, связано с человеческим способом умственной деятельности, требующим образования понятий и работы с ними. С другой стороны, так как цель программирования — переложить ряд процессов обработки информации на современные машины со сложившимися к настоящему времени принципами их работы, то машинная деятельность накладывает определенные ограничения на форму описания понятий, степень подробности такого описания и на необходимость уложиться в машинные возможности. Однако, с совершенствованием вычислительных машин и использованием новых более совершенных принципов машинной обработки ряд ограничений преодолеваются, и всё ясней выделяются принципиальные ограничения, связанные с вычислимостью процессов, их сложностью и конечностью памяти в каждый момент времени.

В этих условиях первостепенной проблемой становится проблема проектирования общения человека с машиной, обеспечивающего человеку понимание и управление процессом обработки данных на уровне, что должна сделать машина без подробной информации о том, как она это делает. Большую подробность система представляет по особому запросу и, может быть, специалисту другого класса, специализирующемуся на эффективных реализациях введенных типов данных.

В каждом развитом языке программирования имеются исходные (базовые или основные) типы данных, которые реализуются на уровне машинных команд. В языке АДА [14] такие типы называются предопределенными. К ним обычно относятся следующие типы данных (в разных языках программирования они могут отличаться названиями) INTEGER, FLOAT (REAL), BOOLEAN и CHARACTER (целые числа, числа с плавающей запятой, булевы значения, полный набор символов).

Рассмотрим некоторые примеры распространенных базисных типов.

```
    П р и м е р 1.1. Тип: BOOLEAN     /* булев, логический */
BOOLEAN = { TRUE, FALSE }   /* множество констант */
    Операции:
TRUE: -> BOOLEAN;           /* нульарная операция */
FALSE: -> BOOLEAN;          /* нульарная операция */
AND: BOOLEAN x BOLEAN ->BOOLEAN;
OR: BOOLEAN x BOLEAN ->BOOLEAN;
NOT: BOLEAN ->BOOLEAN;
IF_THEN_ELSE: BOOLEAN x  BOOLEAN x BOLEAN ->BOOLEAN;
    Соотношения:  p,q,r - BOOLEAN;
/* операция IF_THEN_ELSE (p,q,r) - это обозначение */
/* логического выражения IF p THEN q ELSE r,       */
/* она будет изображаться как в том, так и в другом*/
/* виде                                            */
IF_THEN_ELSE (TRUE, q, r) = q;
IF FALSE THEN q ELSE r = r;
AND(p,q) = IF p THEN q ELSE FALSE;
OR(p,q) = IF p THEN TRUE ELSE q;
NOT(p) = IF p THEN FALSE ELSE TRUE;
```

В этом примере TRUE и FALSE являются как логическими константами, обозначающими истину и ложь, так и нульарными операциями, выделяющими эти константы в булевом типе данных.

```
П р и м е р 1.2.  Тип: INTEGER      /* целый */
/* множество  констант  типа  INTEGER  зависит */
/* от  языка  транслятора и типа  ЭВМ ; константы  */
```

```
/* выражаются знаком и набором цифр                */
    Операции:
+ : INTEGER x INTEGER -> INTEGER ;   /* сложение */
- : INTEGER x INTEGER -> INTEGER ;   /* вычитание */
* : INTEGER x INTEGER -> INTEGER ;   /* умножение чисел */
DIV : INTEGER x INTEGER -> INTEGER ; /* частное от деления*/
MOD : INTEGER x INTEGER -> INTEGER ; /* остаток от деления*/
ABS : INTEGER -> INTEGER ; /* абсолютное значение числа */
=:  INTEGER x INTEGER -> BOOLEAN; /*отношение равенства */
>:  INTEGER x INTEGER -> BOOLEAN; /*отношение больше */
<:  INTEGER x INTEGER -> BOOLEAN; /*отношение меньше */
```

Деление нацело, обозначаемое словом DIV, дает целую часть частного от деления первого числа на второе.

В этом примере в определении типа INTEGER используется тип BOOLEAN.

```
П р и м е р 1.3. Тип: CHAR     /* литерный, символьный */
/* множество констант типа CHAR - множество  символов в*/
/* некотором порядке; различные типы машин могут иметь */
/* свой набор символов,  но  существует международный  */
/* стандартизованный   набор   ISO   и   американский  */
/*              стандарт ASCII                         */
    Операции:
ORD: CHAR -> INTEGER;  /* порядковый номер символа */
CHR: INTEGER -> CHAR;
    Соотношения: ch - CHAR;    n - INTEGER;
CHR (ORD (ch)) = ch;         /* операции CHR и ORD */
IF ((-1<n) AND (n<128))      /* взаимно обратны    */
   THEN  (ORD (CHR (n)) = n) = TRUE
   ELSE FALSE;
```

В этом определении не только используется уже определенный тип INTEGER, но и вводится на нем новая операция CHR, определяющая символ по его порядковому номеру, и соотношение, использующее операцию IF_THEN_ELSE типа BOOLEAN.

Кроме базисных типов данных в языках программирования высокого уровня имеются средства определения сложных структурированных типов данных, с помощью которых представляются в программе состояния сложных объектов или их совокупностей. К ним относятся конструкции построения перечислимых типов, массивов, множеств, записей, списков и т. д.

Рассмотрим некоторые примеры структурированных типов.

В разных языках форма описания структурированных типов различна. В приведенных ниже примерах приводятся, в основном, конструкции языка PASCAL.

Перечислимый тип задается именем типа и перечислением его элементов. Например, тип: дни_недели = (понедельник, вторник, среда, четверг, пятница, суббота , воскресенье).

Тип массив задается следующей конструкцией:

Имя_типа = ARRAY[k..n] OF тип_элемента.

При этом имеются операции, позволяющие:

определить новый массив;

положить элемент в массив по адресу s, где $k \leq s \leq n$;

получить s-ый элемент массива (тот, который был положен по адресу s в последний раз).

Тип запись задается следующей конструкцией:

```
Имя_типа  RECORD
       имя_поля_1:  тип_1
       ..................
       имя_поля_k:  тип_k
      END
```

Каждый элемент объявленного выше типа мыслится в виде набора полей: имя_поля_1,..., имя_поля_k, соответственно типов: тип_1,...,тип_k. При этом предполагается, что имеются операции построения нового элемента этого типа, присвоения нового значения для каждого поля, выдачи текущего значения любого поля записи.

В некоторых языках имеются конструкции построения множественного типа. В языке PASCAL эта конструкция имеет вид SET OF T, где T может быть скалярным (перечислимым), ограниченным (интервальным) или символьным типом. Значениями переменных множественного типа являются множества значений типа T. Над множествами определены операции объединения (+), пересечения (*) и вычитания (−). Допускается проверка того, принадлежит ли данное значение X множеству S. Для этого применяется конструкция X IN S, имеющая логическое значение.

В программах обработки символьной информации часто используются списочные типы данных, которые реализуются через стандартные конструкции языка (например, в языках PASCAL, C), либо задаются специальными конструкциями языка (например, в языках LISP, PROLOG).

Ввиду важности этого типа приведем его описание.

```
П р и м е р 1.4. Тип: LIST_OF_T /*список элементов типа T*/
 /* элементы этого типа в LISP имеют вид (t1 t2 ... tn); */
 /* в языке PROLOG тип LIST_OF_T обозначается T* , а его */
 /*             элементы - [t1,t2,...,tn]               */
    Операции:
 NIL: -> LIST_OF_T;         /* пустой список; [] в PROLOGe */
 CONS:T x LIST_OF_T -> LIST_OF_T;/*вставить элемент в список*/
 CAR: LIST_OF_T -> T; /* начальный элемент непустого списка */
 CDR: LIST_OF_T -> LIST_OF_T /* список без первого элемента */
 /*  в PROLOGe операции CAR и CDR задаются выражением  */
 /*       List = [CAR|CDR]                             */
 APPEND:LIST_OF_T x LIST_OF_T -> LIST_OF_T;
                                       /*соединить списки*/
 IS_NIL: LIST_OF_T -> BOOLEAN;
 EQ: LIST_OF_T x LIST_OF_T->BOOLEAN; /* равенство списков */
 Соотношения: t,t1 - T;  list, list1 - LIST_OF_T;
 CAR(CONS(t,list)) =  t;
 CDR(CONS(t,list)) = list;
 APPEND(NIL,list) = list;
 APPEND(CONS(t,list),list1)=CONS(t, APPEND (list,list1));
 IS_NIL(NIL) = TRUE;
 IS_NIL(CONS(t,list)) = FALSE;
 EQ(NIL,NIL) = TRUE;
 EQ(NIL,CONS(t,list)) = FALSE;
 EQ(CONS(t,list),NIL) = FALSE;
 EQ(CONS(t,list),CONS(t1,list1)) = IF (t=t1)
                     THEN EQ(list,list1)  ELSE FALSE;
```

В этом примере приведены основные операции над списками и соотношения между операциями. Реализация типа данных список может быть возложена на трансляторы языка, как например в LISP, PROLOG, или на программистов, как например в языках PASCAL и C. В этом случае реализации могут быть в различной степени правильными (соответствовать описанию) и отличаться эффективностью. По счастью, реализации типа данных список описаны в литературе и имеются в библиотеках примеров программ на этих языках.

Однако, программисты придумывают все новые типы данных (множество, файл, процедура, окно, меню), которые удобны в различных классах задач. С другой стороны, в прикладных областях складываются свои типы данных (матрицы, диаграммы, таблицы, алгебраические выражения, химические формулы, шахматные позиции), которые естественны для этих областей при постановках задачи, описаний алгоритмов, для понимания промежуточных результатов в процессах отладки и доводки алгоритмов.

Таким образом, хотелось бы иметь языковые и программные средства для произвольного расширения системы типов данных в программе и средства, поддерживающие процессы введения новых типов данных и процессы программирования с новыми типами данных. Но сначала нужно сформулировать, что такое тип данных, и каких целей хотят достичь при введении этого механизма.

Из приведенных здесь примеров следует, что тип данных определяется следующими составляющими:

именем — разным именам типов соответствуют разные типы;

системой операций — тем, что с элементами этого типа можно делать;

соотношениями между операциями, выражающими смысл операций;

реализацией — конкретным представлением элементов этого типа и объявленных операций.

Введение в программирование механизма определения произвольных типов данных должен позволить писать и отлаживать программы в общих понятиях (типах и операциях над ними) данной прикладной области или области программирования, не влезая в подробности реализации типа данного.

С другой стороны, отделение реализации типов от их определения и использования позволяет повысить эффективность работы прикладных программ без их переписывания, а только путем усовершенствования реализаций введенных типов. При этом, программы, использующие новые типы, и программы, эффективно реализующие типы, могут писать разные программисты, специализирующиеся в своих областях. Для стыковки работы этих программистов нужны средства контроля правильности реализации типа данного, то есть средства проверки выполнения в реализации объявленных соотношений между операциями.

1.3 Абстрактные типы данных

Рассмотрим более подробно некоторые математические средства, используемые в литературе по типам данных. В основном, в этом подразделе мы будем следовать [46]

О п р е д е л е н и е 1.1. Многосортной алгебраической системой A называется система, состоящая из трех наборов:

- набора множеств $s_1, ..., s_n$, которые называются несущими множествами алгебраической системы — сортами;

- набора $f_1, ..., f_k$ отображений вида

$$f_i : s_{i1} \times ... \times s_{ni} \to s_{mi}, \text{ для } i = 1, ..., k,$$

которые называются операциями из сортов $s_{i1}, ..., s_{ni}$ в сорт s_{mi};

- набора $r_1, ..., r_l$ подмножеств вида

$$r_j \subset s_{a1} \times ... \times s_{aj}, \text{ для } j = 1, ..., l,$$

которые называются отношениями алгебраической системы.

Многосортной алгеброй называется многосортная алгебраическая система с пустым набором отношений.

Примеры, рассмотренные в предыдущем разделе, показывают, что типы данных являются многосортными алгебрами.

З а м е ч а н и е 1.1. В отличие от моделей баз данных, где отношения играют главную роль, в типах данных отношения принято представлять не подмножествами, а соответствующими им функциями в тип данных BOOLEAN. Таким способом достигается терминологическая однородность: отношения в типах данных мыслятся как операции.

Понятие алгебраической системы было введено в математике и изучалось в связи с математическими задачами [65]. Первоначально работы по изучению алгебраических систем казались экзотическими и слишком абстрактными даже для самих математиков. Использование этого понятия в теории программирования придало ему образность и обогатило новыми задачами.

При работе с типами данных, представленными в виде многосортных алгебраических систем, программисту-пользователю может быть неизвестна (чаще всего и неважна) конкретная реализация алгебраической системы, но необходимо знание имен её составных частей и знание некоторых свойств операций и отношений, отражающих существо типов данных, с которыми имеют дело. Такое знание программисту достаточно для того, чтобы задать данные и определить способ их обработки в программах, отвлекаясь от несущественных подробностей реализации. В связи с этим, в типах данных используется понятие абстракции данных, то есть переход от множеств, операций и отношений алгебраических систем, соответствующих типам данных, к их именам и соотношениям между ними.

Для определения абстрактного типа данных необходимо ввести понятия сигнатуры и множества определяющих соотношений.

О п р е д е л е н и е 1.2. Сигнатура Σ многосортной алгебры задается двумя наборами:

- набором $S = \{S_1, ..., S_n\}$ имен несущих множеств;

- набором $OP = \{F_1, ..., F_k\}$ имен операций с указанием типа операции вида

$$F_i : S_{i1} \times ... \times S_{ni} \to S_{mi}, \text{ для } i = 1, ..., k.$$

Прежде, чем ввести определение множества определяющих соотношений, необходимо определить множество правильно построенных выражений — термов в сигнатуре Σ.

Пусть $\Sigma = (S, OP)$ — сигнатура. Пусть для каждого сорта S_i из S задано множество переменных X_i. Множество всех переменных X есть объединение всех множеств X_i . Предполагается, что X_i и X_j не имеют общих элементов, если сорта S_i и S_j — различные сорта.

Множество термов $Term_\Sigma(X, S_i)$ сорта S_i с множеством переменных X определяется индуктивно по следующим правилам:

1. Каждая переменная сорта S_i является термом сорта S_i, то есть $X_i \subset Term_\Sigma(X, S_i)$ для всех S_i из S;

2. Каждое имя нульарной операции из множества OP сорта S_i является термом сорта S_i;

3. Для любого имени операции F из OP типа:

$$F : S_{d1} \times ... \times S_{dm} \to S_i,$$

и термов $t_1 \in Term_\Sigma(X, S_{d1})$, ..., $t_m \in Term_\Sigma(X, S_{dm})$ слово $F(t_1, ..., t_m)$ является термом сорта S_i.

Рассмотрим пример сигнатуры и термов в этой сигнатуре.

```
П р и м е р 1.5. Сигнатура типа Nat имеет:
 Сорта: nat   /* т.е. S = { nat } */
 Операции: 0 -> nat;     /* т.е. OP = { 0,succ, add} */
           succ: nat -> nat;
           add: nat x nat -> nat;
```

Пусть $\{m, n\} = X$ — множество переменных типа nat. Тогда слова $succ(0)$, $succ(n)$, $add(m, n)$, $succ(add(m, n))$, $succ(add(0, add(m, m)))$ являются термами типа $Term_{Nat}(X, nat)$, потому что они могут быть построены по правилам 1-3.

Слова $succ(m, n), add(0)$ не являются правильно построенными термами в сигнатуре Nat, так как их нельзя построить по правилам 1-3. Слово $add(x, m)$ не является термом типа $Term_{Nat}(X, nat)$, так как x не входит в множество переменных $X = \{n, m\}$.

Множество всех термов в сигнатуре Σ от переменных, входящих в множество X, обозначается через $Term_\Sigma(X)$, то есть

$$Term_\Sigma(X) = \bigcup_{Si \in S} Term_\Sigma(X, S_i).$$

Итак, мы определили термы, которые представляют собой синтаксические конструкции, выражающие композиции операций сигнатуры от констант и переменных. Поэтому, если переменным терма придать значения в какой либо алгебре с сигнатурой Σ, то и всему терму можно однозначно придать значение, которое получается в результате применения операций в последовательности, указанной в терме.

Более точно. Пусть A — алгебра с сигнатурой $\Sigma = (S, OP)$. Тогда для любого множества переменных X сортов S и любого отображения $ev : X \to A$, сохраняющего сорта, существует продолжение этого отображения в A на все множество термов $Term_\Sigma(X)$, заданное следующим индуктивным правилом:

1. если терм t — переменная из множества X, то $ev(t)$ уже определено; если терм $t = F$— нульарная операция, то $ev(t) = F$ — элемент алгебры A, определенный нульарной операцией F;

2. если терм $t = F(t_1, ..., t_s)$ для некоторой операции F из OP и некоторых термов $t_1, ..., t_s$, то

$$ev(F(t_1, ..., t_s)) = F(ev(t_1), ..., ev(t_s)).$$

Так как множество термов в сигнатуре Σ является алгеброй в этой сигнатуре, то примером такого продолжения отображения является подстановка термов.

Пусть $X = \{x_1, ...x_n\}$ — множество переменных и $x_1 \mapsto b_1, ..., x_n \mapsto b_n$ — сопоставление переменным термов $b_1, ..., b_n \in Term_\Sigma(X)$ того же сорта. Тогда подстановка σ этих термов в произвольный терм $t \in Term_\Sigma(X)$ определяется обычным индуктивным правилом:

1. Если $t = x_i,\ x_i \in X$, то $\sigma(t) = b_i$;

2. Если $t = F(),$ где $F() \in OP$ — нульарная операция, то $\sigma(t) = F()$;

3. Если $t = F(t_1, ..., t_n)$, то $\sigma(t) = F(\sigma(t_1), ...\sigma(t_n))$.

В качестве другого примера рассмотрим множество неотрицательных целых чисел как алгебру в сигнатуре Nat, в которой нульарной операции 0 соответствует число 0, операции $succ$ соответствует сопоставление числу n число $n+1$, а операции add — сумма чисел. Пусть $X = \{n, m\}$ и $ev(n) = 1,\ ev(m) = 5$. Тогда по определению
$ev(add(succ(n), m)) = add(ev(succ(n), ev(m)) = add(succ(ev(n)), 5) =$
$= add(succ(1), 5) = add(2, 5) = 7.$

О п р е д е л е н и е 1.3. Уравнением в сигнатуре $\Sigma = (S, OP)$ называется тройка $e = (X, L, R)$, где X — множество переменных, L, R —термы одного сорта из множества $Term_\Sigma(X)$ в сигнатуре Σ. Иногда, для краткости, уравнение e мы будем записывать также в виде $L = R$, если множество переменных X очевидно. Уравнение выполняется в Σ–алгебре A, если для любого отображения $ev : X \to A$ выполняется равенство $ev(L) = ev(R)$ в A. Если уравнение e выполняется в алгебре A, то говорят также, что алгебра A удовлетворяет уравнению e.

Теперь мы готовы к тому, чтобы дать определение абстрактного типа данных.

Определение 1.4. Абстрактным типом данных $\mathcal{A}$ называется тройка $\mathcal{A} = (Name, \Sigma, E)$, где $Name$ — имя типа, $\Sigma = (S, OP)$ — сигнатура типа данного, $E = \{e_1, ..., e_u\}$ — определяющие соотношения абстрактного типа данных (конечное множество уравнений в сигнатуре Σ).

Мы уже видели, что именование является важным средством построения новых типов. В сигнатуре отражаются имена операций этого типа данных, а в соотношениях требуемые программистом - пользователем свойства операций.

```
Пример 1.6. Абстрактный тип данных Nat.
 Сорта: nat   /* т.е. S = { nat } */
 Операции: 0 -> nat;     /* т.е. OP = { 0,succ, add} */
           succ: nat -> nat;
           add: nat x nat -> nat;
 Соотношения: /* множество E */
           add(0,n) = n;
           add(n,0) = n;
           add(succ(n),m) = succ((add(n,m));
           add(n,succ(m)) = succ((add(n,m));
```

Реализацией абстрактного типа данных называется сопоставление именам несущих множеств конкретные множества, именам операций конкретные операции на этих множествах.

Определение 1.5. Синтаксически корректной реализацией абстрактного типа данных $\mathcal{A} = (Name, \Sigma, E)$ называется произвольная алгебра A с сигнатурой Σ . Семантически корректной реализацией абстрактного типа данных $\mathcal{A}$ называется алгебра A , удовлетворяющая уравнениям множества E.

Семантически корректная реализации называется также моделью абстрактного типа данных $\mathcal{A}$. У одного абстрактного типа данных может быть несколько моделей.

Определение 1.6. Пусть A и A' две модели абстрактного типа данных $\mathcal{A}$. Гомоморфизмом $j : A \to A'$ моделей (алгебр) называется отображение j несущих множеств A в несущие множества A', сохраняющее сорта элементов и обладающее следующим свойством: для любого F

из множества имен операций OP абстрактного типа данных $\mathcal{A}$ и любого элемента модели A вида $F(a_1, ..., a_s)$ выполнено равенство:

$$j(F(a_1, ..., a_s)) = F(j(a_1), ..., j(a_s)).$$

Если, кроме того, j является взаимно однозначным отображением, то такой гомоморфизм называется изоморфизмом.

П р и м е р ы 1.7. (Реализации, гомоморфизм, изоморфизм)
Рассмотрим следующие реализации абстрактного типа данных Nat с сигнатурой $\Sigma = (nat; 0, succ, add)$. Это натуральные числа N и целые числа $Integer$. При этом, сорту nat сопоставляются множества N и $Integer$, соответственно, а именам операций $0, succ, add$ сопоставляются операции $0, +1, +$ в натуральных и целых числах. Нетрудно видеть, что в этих реализациях выполняются все соотношения абстрактного типа данных Nat, то есть это корректные реализации, и естественное вложение N в $Integer$ совместимо с действием операций, то есть это вложение — гомоморфизм моделей Nat. Так как вложение N в $Integer$ не является взаимно однозначным отображением, то этот гомоморфизм не изоморфизм. Обратное отображение $j : Integer \to N$, сопоставляющее целому числу его абсолютное значение, не гомоморфизм, так как $j(3 + (-3)) = 0$ и не равно $j(3) + j(-3) = 6$.

Другую модель Nat, изоморфную N, дает следующая реализация. Сорту nat сопоставляется множество всех слов A^* в однобуквенном алфавите $A = \{a\}$. Именам операций $0, succ, add$ сопоставляются, соответственно, пустое слово, приписывание к слову одной буквы, конкатенация слов. Отображение $\_ : A^* \to N$ — взаимно однозначный гомоморфизм, то есть изоморфизм.

Прикладные программисты, как правило, не отличают две изоморфные реализации абстрактного типа данных, потому что все, что выполняется и может быть построено в одной реализации, выполняется и может быть построено в другой реализации (хотя может отличаться эффективностью выполнения операций и объемами памяти, занимаемого данными). С этим свойством связано следующее понятие абстракции данных.

О п р е д е л е н и е 1.7. Абстрактной структурой данных называется модель абстрактного типа данных, заданная с точностью до изоморфизма.

Для построения данных и определения способов их обработки в программах должны указываться лишь имена, входящие в определение абстрактного типа данных, без ссылок на конкретные свойства реализации типа. Это делает возможным использование одной и той же программы для работы с разными реализациями абстрактного типа данных. Так обеспечивается свойство, которое программисты называют независимостью работы программы от физического представления данных.

Таким образом, в описаниях (спецификациях) абстрактных типов данных обязательно задаются имя типа, сигнатура и система соотношений между операциями, то есть дается описание абстрактного тапа данных. Кроме того, иногда в спецификации указывается, а иногда подразумевается некоторая абстрактная структура данных, то есть некоторая реализация абстрактного типа данных с точностью до изоморфизма. Это делается в тех случаях, когда программист хочет использовать свойства данных из некоторой абстрактной структуры данных, не ограничивая себя только теми свойствами, которые следуют из множества определяющих соотношений соответствующего абстрактного типа данных.

Распространенный прием задания абстрактных структур данных — это задание их в виде инициальных моделей [50].

О п р е д е л е н и е 1.8. Инициальная модель $I(\mathcal{A})$ абстрактного типа данных $\mathcal{A}$ — это модель, обладающая следующим универсальным свойством: для любой модели A абстрактного типа данных $\mathcal{A}$ существует единственный гомоморфизм из модели $I(\mathcal{A})$ в модель A.

У т в е р ж д е н и е 1.1. Если $I(\mathcal{A})$ и $I'(\mathcal{A})$ — инициальные модели абстрактного типа данных $\mathcal{A}$, то они изоморфны.

Д о к а з а т е л ь с т в о. Пусть $I(\mathcal{A})$ и $I'(\mathcal{A})$ — инициальные модели абстрактного типа данных $\mathcal{A}$. Тогда, по определению инициальности существуют гомоморфизмы:

$$h : I(\mathcal{A}) \to I'(\mathcal{A}) \text{ и } h' : I'(\mathcal{A}) \to I(\mathcal{A}).$$

Композиция гомоморфизмов h и h' является гомоморфизмом из $I(\mathcal{A})$ в $I(\mathcal{A})$. Но из определения инициальности следует, что гомоморфизм из $I(\mathcal{A})$ в $I(\mathcal{A})$ только один, и, следовательно, эта композиция совпадает с тождественным отображением на $I(\mathcal{A})$. Аналогично показывается, что композиция h' и h является тождественным отображением на $I'(\mathcal{A})$. Отсюда следует, что h — взаимно однозначное отображение и, следовательно, — изоморфизм.

Т е о р е м а 1.2.[50] Для любого абстрактного типа данных с соотношениями типа уравнений существует инициальная модель.

Доказательство теоремы проводится стандартным образом. На множестве термов без переменных в сигнатуре абстрактного типа данных индуктивно строится отношение эквивалентности, удовлетворяющее естественным свойства конгруэнтности, исходя из соотношений абстрактного типа данных. Инициальная модель представляется в виде фактор множества таких термов по построенному отношению эквивалентности.

Пусть $\mathcal{A} =< Name, \Sigma, E >$ — произвольный абстрактный тип данных. Рассмотрим множество термов $Term_\Sigma(\emptyset)$ с пустым множеством переменных. Тогда, для любой модели A абстрактного типа данных $\mathcal{A}$ существует единственное отображение

$$ev : Term_\Sigma(\emptyset) \to A,$$

сопоставляющее каждому терму его значение в алгебре A.

Обозначим через $eq_\mathcal{A}$ множество всех пар термов (t_1, t_2) таких, что уравнение $t_1 = t_2$ выполняется во всех моделях A, то есть $ev(t_1) = ev(t_2)$ верно во всех моделях $\mathcal{A}$.

У т в е р ж д е н и е 1.3. Отношение $eq_\mathcal{A}$ обладает следующими свойствами:

1. (Рефлексивность.) Для любого терма $t \in Term_\Sigma(\emptyset)$ пара $(t, t) \in eq_\mathcal{A}$;

2. (Симметричность.) Если $(t_1, t_2) \in eq_\mathcal{A}$, то $(t_2, t_1) \in eq_\mathcal{A}$;

3. (Транзитивность.) Если (t_1, t_2) и $(t_2, t_3) \in eq_\mathcal{A}$, то $(t_1, t_3) \in eq_\mathcal{A}$;

4. Если $(t_1, t'_1), ..., (t_n, t'_n) \in eq_\mathcal{A}$ и $F(t_1, ..., t_n)$ —терм для некоторого $F \in OP$, то $(F(t_1, ..., t_n), F(t'_1, ..., t'_n)) \in eq_\mathcal{A}$;

5. Если соотношение $(X, L, R) \in E$ и $\sigma : X \to Term_\Sigma(\emptyset)$ — произвольное означивание переменных термами того же типа, то пара $(\bar{\sigma}(L) = \bar{\sigma}(R)) \in eq_\mathcal{A}$.

Доказательство утверждения следует из определений и очевидно.

О п р е д е л е н и е 1.9. Отношение, обладающее свойствами 1–3, называется отношением эквивалентности. Отношение, обладающее свойствами 1–4, называется конгруэнцией. Отношение, обладающее свойствами 1–5, называется конгруэнцией, удовлетворяющей равенствам E. Таким образом, отношение $eq_\mathcal{A}$ — отношение конгруэнции на множестве термов $Term_\Sigma(\emptyset)$, удовлетворяющее соотношениям E.

О п р е д е л е н и е 1.10. Обозначим через $\cong_E$ отношение эквивалентности на множестве термов $Term_\Sigma(\emptyset)$, построенное по правилам 1–5, то есть $\cong_E$ — наименьшее отношение, обладающее свойствами 1–5. Если терм $t \in Term_\Sigma(\emptyset)$, то через $[t]$ обозначается множество всех термов, эквивалентных t по отношению $\cong_E$, то есть $[t] = \{t' : t' \cong_E t\}$.

У т в е р ж д е н и е 1.4. Для любых двух термов $t, t' \in Term_\Sigma(\emptyset)$ множества $[t_1]$ и $[t_2]$ либо совпадают, либо не пересекаются.

Утверждение верно для любого отношения эквивалентности. Действительно, если $[t_1]$ и $[t_2]$ имеют общий элемент t, то, по определению, $t \cong_E t_1$ и $t \cong_E t_2$. Отсюда, из свойств симметричности и транзитивности следует, что $t_1 \in [t_2]$, $t_2 \in [t_1]$ и $[t_1] = [t_2]$.

Теперь мы готовы к доказательству теоремы существования инициальной алгебры.

В качестве множества элементов алгебры $I(\mathcal{A})$ рассмотрим множество:

$$I(\mathcal{A}) = \{[t] : t \in Term_\Sigma(\emptyset)\},$$

которое называют фактор-множеством множества термов по конгруэнции $\cong_E$.

Операции в этом множестве действуют по следующим правилам:

1. Если $F \in OP$ — нульарная операция,то ее действием будет элемент $[F]$ в $I(\mathcal{A})$;

2. Если $F(x_1, ..., x_n) \in OP$ — n-арная операция и $[t_1], ..., [t_n] \in I(\mathcal{A})$, то $F([t_1], ..., [t_n])$ полагается равным $[F(t_1, ..., t_n)] \in I(\mathcal{A})$ (корректность этого определения обеспечивается свойством 4).

Таким образом, $I(\mathcal{A})$ — Σ - алгебра. Более того, благодаря свойству 5, $I(\mathcal{A})$ — модель абстрактного типа данного $\mathcal{A}$. Докажем, что эта модель инициальна.

Пусть A — произвольная модель абстрактного типа данного $\mathcal{A}$. Тогда, как было показано выше, существует единственное отображение $ev : Term_\Sigma(\emptyset) \to A$, совместимое с действием операций. При этом, если $t \cong_E t'$, то $ev(t) = ev(t')$, так как в алгебре A выполняются соотношения E абстрактного типа данных $\mathcal{A}$. Поэтому отображение $j : I(\mathcal{A}) \to A$, заданное соотношением $j([t]) = ev(t)$, определено корректно. Так как отображение ev по определению перестановочно с именами операций, то j — гомоморфизм. Его единственность следует из единственности отображения ev, перестановочного с именами операций.

Следствиями этой теоремы и приведенных построений являются следующие утверждения.

Т е о р е м а 1.5. Уравнение $t_1 = t_2$ для термов $t_1, t_2 \in Term_\Sigma(\emptyset)$ выполняется во всех моделях абстрактного типа данных $\mathcal{A}$ тогда и только тогда, когда оно может быть получено по правилам 1–5, то есть, когда оно выполняется в инициальной модели.

Т е о р е м а 1.6. Отношения $eq_{\mathcal{A}}$ и $\cong_E$ совпадают.

1.4 Фрагменты схем баз данных как примеры понятий

В настоящее время возникают базы данных со сложно организованными схемами. К таким базам данных относятся распределенные базы данных с большим числом локальных баз данных, научные базы данных со сложной организацией областей, базы данных со сложно организованными объектами.

В таких базах данных, прежде чем обратиться к самим данным, необходим поиск по схеме базы данных для определения места и структуры требуемых пользователю данных. Знания о схеме базы данных требуется для составления запроса к базе данных. Учитывая сложность и величину схемы базы, для удобства просмотра и составления запросов схема базы данных разбивается по смыслу на фрагменты.

Каждый фрагмент представляет собой осмысленную часть схемы базы данных, которая может быть использована в других фрагментах схемы и в запросах. Спецификация различных фрагментов базы данных может производиться различными людьми — специалистами в областях знаний, к которым относятся данные фрагменты. Более того, некоторые фрагменты схем баз данных, являются стандартами, общеизвестными специалистам и используемыми как в базах данных, так и в запросах. Использование стандартных фрагментов является существенным требованием при проектировании схем современных баз данных.

Рассмотрим некоторые примеры спецификаций фрагментов схем баз данных.

П р и м е р 1.8.

Имя фрагмента: PERSON

Образующие:

PersonIdentityNo : PERSON $\to$ INT;
Name: PERSON $\to$ STRING;
Age: PERSON $\to$ INT;
Children: PERSON $\to$ Set_of(PERSON);
Соотношения: Var p_1, p_2 : PERSON;
IF PersonIdentityNo(p_1) = PersonIdentityNo (p_2) THEN
$p_1 = p_2$.

Имя фрагмента: STUDENT
Образующие:
Isa: STUDENT $\hookrightarrow$ PERSON;
GroupNamber: STUDENT $\to$ INT;
Advisor: STUDENT $\to$ PROF;
....

Имя фрагмента: PROF
Isa: PROF $\hookrightarrow$ PERSON;
....

В этих примерах используются стандартные типы данных INT, STRING. В общем случае, в описаниях фрагментов схем баз данных могут использоваться произвольные типы данных, включая структурированные типы и абстрактные типы данных. В приведенных фрагментах, как и в абстрактных типах данных, вводятся имена областей: PERSON, STUDENT, PROF; имена функций: PersonIdentityNo, Name, Age, Advisor, отражающие свойства элементов соответствующих областей; соотношения. Кроме того, в фрагменте PERSON используется конструкции Set_of и рекурсивная конструкция Set_of(PERSON). В фрагментах STUDENT и PERSON используется конструкция Isa:$T_1 \hookrightarrow T_2$, определяющая область T_1 как подобласть области T_2.

Другие примеры дают обычные схемы реляционных баз данных, рассмотренные в предыдущей главе. В этом случае, схема базы данных задается набором доменов $D_1, ..., D_n$, набором схем отношений

$$R^1(a^1_1, ..., a^1_{n_1}),\ \ ...,\ \ R^k(a^k_1, ..., a^k_{nk})$$

(они называются базисными), где $R^1, ..., R^k$ — имена отношений, $^1_1, ..., a^k_{nk}$

имена атрибутов, для каждого из которых отображением

$$t : \{_1^1, ..., a_{nk}^k\} \to \{D_1, ..., D_n\}$$

определен домен значений. Кроме того, задаются отображения вложения

$$R^j(a_1^j, ..., a_{nj}^j) \hookrightarrow D_{t(a_1)} \times ... \times D_{t(a)}, \text{ for } j = 1, ..., k$$

и соотношения (зависимости, ограничения целостности базы данных) между этими отношениями вида:

$$\begin{array}{rcl} f_1(R^1, \ldots, R^k) & = & g_1(R^1, \ldots, R^k) \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ f_l(R^1, \ldots, R^k) & = & g_l(R^1, \ldots, R^k), \end{array}$$

где f_i и g_i — операции над отношениями, выраженные в виде композиций основных операций.

В частности, в виде подобных соотношений можно записать условие функциональной зависимости какого–либо атрибута от набора других, вхождение одного отношения в другое как подмножества, пустоту пересечения каких–либо отношений и т.д. Соотношения, зафиксированные в базе данных, отражают условия целостности базы данных, которые обязательно должны выполняться, отражают смысл представленных данных и используются в приложениях.

1.5 Примеры представления общих понятий

В этом разделе рассмотрим некоторые примеры понятий из научных областей знаний и некоторые принципы их представления для автоматизированной обработки.

Нас будут интересовать средства представления понятий, ориентированные, главным образом, не на программирование отдельных задач, а на создание автоматизированной информационной системы о знаниях в некоторой предметной области и, затем, на автоматизированную поддержку постановок и решений многочисленных задач, для формулировок которых используются понятия, представленные в этой системе. При этом знания предметной области представляются в виде системы понятий, в которой при определении одних понятий могут использоваться другие, ранее определенные в системе понятия.

Естественно, что одну и ту же предметную область можно по-разному структурировать, и у определений понятий могут быть разные эквивалентные (или не совсем эквивалентные) формы. Выбор структуризации предметной области и форм определений при представлении знаний предметной области средствами языка представления должно, в основном, определяться традициями в этой предметной области и искусством проблемного специалиста, представляющего знания.

С этой точки зрения задача представления знаний для автоматизированной обработки аналогична задаче подготовки учебника по специальности представляемой области знания. Использование традиций данной предметной области при представлении знаний необходимо для облегчения восприятия представленных знаний и для удобства использования опыта специалистов в предметной области при обращении к системе. Система представления знаний должна организовывать естественную среду знаний для специалистов проблемной области. Это тем более важно в случаях, когда представлением знаний занимается одна группа проблемных специалистов, а использованием представленных в системе знаний — другая группа специалистов.

Структуризация представления знаний обеспечивает возможность работы с обозримыми фрагментами представленных знаний. Она выбирается также, исходя из удобства выражения знаний, постановок задач, проверки правильности выражений и локализации ошибок. Система понятий должна быть достаточно полной, обеспечивающей эффективное выражение постановок типичных задач данной предметной области. Кроме того, структуризация представления знаний и система понятий строится так, чтобы обеспечить устойчивость системы понятий при изменениях в определениях отдельных понятий.

С точки зрения проблемного специалиста средства языка представления служат для выражения определений понятий проблемной области, для выражения доказательств некоторых свойств введенных понятий, для постановок задач и задания запросов к системе представленных знаний. С другой стороны, выражения на языке представления имеют внутреннюю интерпретацию, которая может быть скрыта от пользователя и служит для организации эффективного получения ответов на запросы и проверки непротиворечивости введенных спецификаций понятий.

Программирование с использованием языка представления понятий состоит из следующих процессов:

формулировка и ввод определений понятий предметной области;

формулировка и доказательство некоторых свойств понятий, которые не входят в определение понятий, но известны специалисту и по его мнению должны быть отражены в системе;

постановка задач проблемной области с использованием введенных понятий.

Процесс программирования в такой системе — это диалог, в котором специалист формулирует поставки задач проблемной области, используя уже введенные понятия как модули, или вводит новые понятия и их свойства.

Пусть, к примеру, наша предметная область — это понятия школьного учебника физики. Первый пример соответствует понятию механическое движение.

Имя понятия: Движение.
ПРОСТРАНСТВО, ВРЕМЯ, ТЕЛА — области.
Место — отображение.
Место: ТЕЛА $\times$ ВРЕМЯ $\rightarrow$ ПРОСТРАНСТВО.

По этому определению каждое механическое движение требует наличия трех областей, которые называются ПРОСТРАНСТВО, ВРЕМЯ, ТЕЛА и отображения Место, которое каждому телу в каждый момент времени задает положение в пространстве. Например, в движениях по прямой область ПРОСТРАНСТВО представляет собой прямую, в движениях по окружности — окружность, в движениях в трехмерном пространстве — трехмерное пространство. Область ВРЕМЯ обычно представляется множеством действительных чисел, но в задачах с дискретном временем может быть представлена множеством целых чисел. Область ТЕЛА в разных задачах также может быть разной. В задачах о движении одного тела она состоит из одного тела, а в задачах о движении n тел область ТЕЛА состоит из n элементов. В любом случае, когда речь идет о механическом движении, то выделяются области ПРОСТРАНСТВО, ВРЕМЯ, ТЕЛА и подразумевается, что есть отображение, задающее место в пространстве для каждого тела в каждый момент времени.

При представлении знаний, исходим из следующих требований к системе представлений:

модульность — возможность формирования новых определений и запросов на основе уже введенных в систему;

открытость — возможность ввести в любой момент новые определения и запросы и использовать их затем как модули;

понятность — близость формальных выражений на языке представления исходным формулировкам определений и запросов на естественном языке;

возможность использования сокращений и условных обозначений для понятий и формул;

способность интерпретировать неполную или противоречивую информацию.

Проиллюстрируем некоторые из этих принципов примером.

Имя понятия: Равномерное движение Т.
Другое имя: Т равномерно движется.
Это: Движение. Используются: ЧИСЛА.
Образующие:
Т — элемент области ТЕЛА.
СКОРОСТЬ — элемент области ЧИСЛА.
Промежуток времени: ВРЕМЯ$\times$ВРЕМЯ $\to$ ЧИСЛА.
Если t_1, t_2 — ВРЕМЯ, то Длина пути(t_1, t_2) — ЧИСЛА.
Сокращения:
t = ПРОМЕЖУТОК ВРЕМЕНИ.
S = ДЛИНА ПУТИ.
V = СКОРОСТЬ.
Соотношения:
Если t_1, t_2 — ВРЕМЯ, то $S(t_1, t_2) = V * t(t_1, t_2)$.
Если t_1, t_2, t_3 — ВРЕМЯ, то $t(t_1, t_3) = t(t_1, t_2) + t(t_2, t_3)$.
Утверждения:
Если t_1, t_2 — ВРЕМЯ, то $V = S(t_1, t_2)/t(t_1, t_2)$ и
$t(t_1, t_2) = S(t_1, t_2)/V$

Здесь определяется понятие с двумя синонимичными именами "Равномерное движение Т" и "Т равномерно движется". В этом примере определяется понятие с выделенным параметром Т .

Использование в определении имен ранее построенных понятий или терминов, входящих в ранее построенные понятия, означает, что тексты определений и внутренние представления этих понятий включаются в текст и внутреннее представление формируемого понятия. В данном случае это означает, что внутреннее представление понятия "Равномерное движение Т" строится из внутренних представлений ранее определенного понятия "Движение" и базисного понятия "ЧИСЛА".

Здесь не приводится формального определения понятия "ЧИСЛА", но предполагается, что по определению оно содержит область "ЧИСЛА", на которой определены операции сложения (+), вычитания (−), умножения (∗), деления (/) удовлетворяющие стандартным аксиомам поля. Предполагается также, что аппроксимация понятия "ЧИСЛА" содержит машинные числа с соответствующими операциями на них.

При построении представления понятия "Равномерное движение" к образующим и соотношениям понятий "Движение" и "ЧИСЛА" добавляются новые преобразования, "Промежуток времени", "Длина пути" и элементы "Т" и "Скорость". При этом область определения и область значений преобразователя "Промежуток времени" указаны явно, а область определения и область значений преобразования:

$$\text{Длина пути: ВРЕМЯ}\times\text{ВРЕМЯ} \rightarrow \text{ЧИСЛА}$$

задается с помощью оператора языка "Если ..., то ...".

Выражения " Т — элемент области ТЕЛА" и "СКОРОСТЬ — элемент области ЧИСЛА" означают , что вводимые в определении термины " Т " и "скорость" являются элементами, соответственно, областей "ТЕЛА" и "ЧИСЛА".

Сокращения t, S, V являются сокращенными синонимами для имен соответствующих преобразований.

Дополнительные определяющие соотношения понятия "Равномерное движение" представляют собой равенства между композициями некоторых преобразований и также задаются с помощью оператора языка "Если ... , то ... ".

Равенства, входящие в "утверждения" определения, могут быть получены (выведены) из образующих и определяющих соотношений понятия "Равномерное движение", но вводятся сюда для того, чтобы они вошли во во внутреннее представление этого понятия.

Используя введенные понятия, может быть сформулирована следующая задача.

Задача:
Автомобиль равномерно движется.

t_1, t_2, t_3 — элементы области ВРЕМЯ.
$t_1 =$Время начала движения.
$t_2 =$Время замера.
$t_3 =$Время конца движения.
$t(t_1, t_2) = 10$. $S(t_1, t_2) = 5$. $S(t_1, t_3) = 15$.
Чему равно $t(t_1, t_3)$?

Тексту этой задачи сопоставляется внутреннее представление, которое строится из внутреннего представления понятия "Т равномерно движется", где вместо параметра Т подставлен термин "Автомобиль", добавлены элементы t_1, t_2, t_3 в области ВРЕМЯ и соотношения $t(t_1, t_2) = 10$, $S(t_1, t_2) = 5$, $S(t_1, t_3) = 15$. Вопрос задачи интерпретируется как запрос к внутреннему представлению задачи. В ответ выдается элемент области "ЧИСЛА", равный $t(t_1, t_3)$.

Можно было бы сформулировать и другие задачи с несколькими телами и другими вопросами. Можно было бы ввести понятия равноускоренного движения и задачи на равноускоренное движение по аналогии с приведенными. Если бы мы захотели, чтобы система отслеживала единицы измерения физических величин, то нужно было ввести понятия единицы измерения, величины времени, величины длины и потребовать, чтобы преобразования "Промежуток времени" и "Длина пути" принимали значения, соответственно, в областях "ВЕЛИЧИНА ВРЕМЕНИ" и "ВЕЛИЧИНА ДЛИНЫ", а не в области "ЧИСЛА".

В языке представления кроме понятий, которые используются для формулировки других понятий, можно определить схемы вопросов, которые могут быть использованы для формулировки других вопросов или вопросов задач. Примерами таких определений являются следующие схемы вопросов "где?" и "когда?".

Имя понятия: Где Т?
Образующие:
ТЕЛА, ПРОСТРАНСТВО — области.
Положение: ТЕЛА $\to$ ПРОСТРАНСТВО.
Т — элемент области ТЕЛА.
Чему равно Положение (Т)?

Имя понятия: Когда S?
Образующие:

СОБЫТИЯ, ВРЕМЯ — области.
Время: СОБЫТИЯ → ВРЕМЯ.
S — элемент области СОБЫТИЯ.
Чему равно Время (S)?

Чтобы применить эти вопросы в конкретной задаче или понятии, нужно проинтерпретировать схемы вопросов в представлении задачи или понятия.

Итак, перечислим некоторые функции системы представления понятий.

Система строит внутреннее представление понятия по выражениям языка представления, описывающим данное понятие или задачу. Строит запрос к представлению, соответствующий вопросу задачи или схеме вопроса. Кроме того, система осуществляет логический контроль за правильностью написания выражений на языке представления прикладным специалистом и может задавать вопросы пользователю для уточнения интерпретации текста представления, а также постоянно проводит контроль на непротиворечивость представления, уже построенного системой. Система автоматически расширяет текст пользователя за счет текстов модулей, указание на которые делает пользователь, а также выполняет необходимые справочные функции о текстах, введенных в систему.

2 Категорный подход к представлению понятий

2.1 Общее описание подхода

В настоящее время одним из основных средств, которое применяется для представления декларативных знаний (понятий) в системах искусственного интеллекта являются семантические сети. Семантическая сеть представляет собой сеть с поименованными элементами. В методе представления знаний семантическими сетями предполагается, что всякое декларативное знание можно закодировать в виде сети.

Такой подход полностью аналогичен сетевому подходу к моделированию баз данных. Он имеет те же достоинства и недостатки. Основной недостаток заключается в том, что из-за закодированности информации внешняя логика понятий сложно представляется, и сама информация становится трудно доступной проблемному специалисту.

Альтернативой семантическим сетям для представления понятий является категорный подход, использующий аппарат и методы математической теории категорий для определения семантики логических конструкций. Соотношение между подходом, использующим семантические сети, и предлагаемым почти такое же, как между сетевыми и реляционными подходами к моделированию баз данных. Здесь, как и в реляционном подходе, предполагается, что понятие (базу данных) можно представить в виде набора областей, системы отображений и отношений, но имеются меньшие требованиями к полноте представляемой информации.

Категорный подход к представлению знаний является естественным развитием реляционного подхода к базам данных, дополненного идеями и методами абстрактных типов данных в программировании [3],[72],[50],[41], [53],[54]. При этом используются достижения приложений математической теории категорий к математической логике [64],[16].

В категорном подходе к представлению знаний предполагается, что представляемые знания прикладной области могут быть структурированы в виде системы понятий. С каждым понятием связывается имя понятия, определение и знания, вытекающие из определения. Теория категорий предоставляет математические средства для отражения семантики и логики понятий. Каждому представляемому понятию в этом подходе сопоставляется алгебраическая модель понятия (категория), в которой потенциально отражается полное знание о предмете понятия, вытекающее из определения этого понятия, а также строится конечная аппроксимация категории (фрагмент категории), в которой отражаются уже имеющиеся знания о представляемом понятии, отображенные в системе представления и непосредственно доступные пользователю.

Средства категорного подхода к представлению знаний условно можно разделить на три основные части:

- средства построения алгебраической модели (категории);
- средства построения конечной аппроксимации категории;
- язык представления знаний.

Все три составляющие подхода сильно связаны между собой:

- категория и ее аппроксимация задаются языковыми средствами;
- корректность новых языковых выражений определяется по уже построенной аппроксимации;
- категория, отражающая полные знания о представляемом понятии, — это идеальный объект, который является пределом своих конечных аппроксимаций.

В основе категорного подхода к представлению знаний так же, как и в основе реляционного подхода к моделированию баз данных, лежит некоторый набор операций. Мы будем называть их категорными операциями, так как они были введены в математической теории категорий (см.,например, [16]).

Категорные операции позволяют выразить представляемые понятия и определить требуемую их обработку. Особенностью категорных операций является возможность построения любых теоретико множественных конструкций из областей, не все элементы которых известны.

Единая категорная интерпретация понятий, основанная на общем наборе операций, позволяет достичь высокой степени интеграции понятий в систему знаний.

В данной работе дается описание предлагаемых средств категорного подхода к представлению понятий. Некоторые из представленных здесь результатов были опубликованы автором в [8], [9], [10], [11].

В настоящее время разработан макет диалоговой системы, основанный на категорном подходе, для создания баз понятий. Макет разработан на языке PDC Prolog и функционирует на персональных ЭВМ типа IBM PC AT.

2.2 Категорные средства представления понятий

В этом разделе неформально описываются некоторые средства математической теории категорий, служащие для представления понятий.

Почему именно категории выбраны для представления понятий и что это такое?

Основным тезисом категорного подхода к представлению знаний является предположение, что определение всякого понятия может быть представлено в виде наборов имен областей, имен преобразований и соотношений. При этом, не все элементы областей могут быть известны внутри данного понятия и могут уточняться в других понятиях.

Широта принятого тезиса подтверждается тем, что схемы базы данных и абстрактные типы данных являются примерами таких понятий. Эксперименты показывают, что так представляются понятия таких предметных областей, как механика, химия и т.д.

Уточним теперь, какими свойствами обязательно обладают совокупности имен областей и отображений независимо от того, какое понятие ими представляется.

Так как именам отображений в конкретных примерах соответствуют отображения одних множеств в другие, то выполняются следующие свойства.

Если $f_1 : T_1 \to T_2$ и $f_2 : T_2 \to T_3$ — имена отображений такие, что $_(f_1) = _(f_2)$, то строится имя $f_2 *$
f_1 отображения из T_1 в T_2, которое называется композицией отображений f_1 и f_2 .

Композиция отображений обладает свойством ассоциативности. Это значит, что для любых отображений f_1, f_2 и любого отображения $f_3 : T_3 \to T_4$ выполнено равенство

$$f_3 * (f_2 * f_1) = (f_3 * f_2) * f_1. \tag{1}$$

Для отображений множеств равенство (1) легко проверяется на элементах.

Кроме того, для любой области T можно ввести тождественное отображение $id(T) : T \to T$, композиция которого с любыми отображениями $f : T_1 \to T$ и $g : T \to T_2$ дает следующие равенства:

$$id(T) * f = f, \quad g * id(T) = g. \tag{2}$$

Математическая структура, обладающая операцией типа операции композиция со свойствами (1) и (2), называется категорией [16]. Таким образом, система имен областей и отображений, которая сопоставляется представляемому понятию и удовлетворяет условиям (1) и (2), — это категория. Объектами этой категории являются имена областей, а морфизмами — имена отображений, рассмотренные с точностью до равенства.

Обозначим эту категорию через $\mathcal{C}$. Объекты этой категории мы будем называть областями. Множество всех морфизмов из области $_1$ в область $_2$ категории $\mathcal{C}$ обозначается через $\mathcal{C}(T_1, T_2)$.

Для указания элементов объектов в теории категорий имеется специальный способ, который естественно использовать и в представлении знаний.

Вводится особая область ТОЧКА, которая сокращенно обозначается через I, обладающая следующим свойством: для каждой области T из категории $\mathcal{C}$ существует единственный морфизм из T в I, который обозначается через $I(T) : T \to I$. Это свойство полностью характеризует область ТОЧКА. В теории категорий ее называют финальным объектом.

Полезно также иметь пустую область $\emptyset$, которая задается следующим свойством: для любой области категории $\mathcal{C}$ множество всех морфизмов $\mathcal{C}(\emptyset, T)$ состоит из единственного морфизма $\emptyset(T) : \emptyset \to T$. В теории категорий такой объект называют инициальным.

В дальнейшем будет предполагаться, что рассматриваемые здесь категории $\mathcal{C}$ имеют области $\emptyset$ и I.

Категория областей $\mathcal{C}$ называется непротиворечивой, если область $\emptyset$ неизоморфна области I. Соответственно, если понятие представляется непротиворечивой категорией, то представление называется непротиворечивым.

Пусть — некоторая область из категории $\mathcal{C}$. Известные в представлении элементы области T задаются морфизмами $t : I \to T$ из области ТОЧКА в область T. Множество морфизмов $\mathcal{C}(I, T)$ соответствует множеству всех известных элементов области T в категории $\mathcal{C}$.

Если $f : T_1 \to T_2$ — морфизм, то он индуцирует отображение известных элементов области T_1 в известные элементы области T_2. Действительно, пусть $t : I \to T_1$ — произвольный элемент области T_1, тогда композиция морфизмов $f * t : I \to T_2$ — элемент области T_2.

В общем случае возможно, что два разных морфизма индуцируют одно и то же отображение на множестве известных элементов. Таким образом, область в категорном подходе не полностью характеризуется множеством ее известных элементов. В общем случае область определяется взаимоотношениями со всеми областями категории $\mathcal{C}$, а не только взаимоотношениями с областью ТОЧКА. Это означает, что области в категорном подходе это не множества. Их можно себе представлять как недостроенные множества.

Более того, в приведенном ранее примере определения движения области ПРОСТРАНСТВО, ВРЕМЯ, ТЕЛА вообще не имеют известных элементов, так как их нельзя получить из этого определения. Это распространенная ситуация при представлении общих понятий, в которых неполнота является важным свойством. Такие понятия в разных случаях наполняются разным содержанием. Например, понятие движение может быть дополнено до понятий движения по прямой, движения по окружности, движения с дискретным временем, в которых области с одинаковыми именами могут иметь разные множества элементов.

Для представления понятий категорными средствами нужны категории с операциями, которые позволяют по одним областям и морфизмам строить другие области и морфизмы. Некоторые операции такого типа были уже введены — это $\_()$, $\_()$, которые каждому морфизму сопоставляют соответствующие области, операция $id()$, которая каждой области сопоставляет тождественное отображение. Если $\_(f_1) = \_(f_2)$, то определена операция (f_1, f_2), которая сопоставляет паре морфизмов f_1, f_2 морфизм $f_2 * f_1$. Другие примеры операций — это выделенные области $I, \emptyset$, операции $I(), \emptyset()$, которые области T сопоставляют морфизмы $I(T), \emptyset(T)$.

Для представления сложных понятий перечисленных выше операций недостаточно. Нужны также операции типа декартова произведения областей, суммы областей, пересечения областей, фактор областей и т.д. При этом следует учесть, что область, как показано было выше, не определяется своими элементами, поэтому теоретико множественные определения этих операций не подходят.

Математики давно искали конструкции, соответствующие теоретико множественным операциям, но не использующие элементы областей. Такая работа была проделана в теории категорий. Точные определения таких конструкций можно найти в [16].

Для примера рассмотрим операцию декартова произведения. Декартово произведение $T_1 \times T_2$ двух множеств T_1, T_2 определяется в теории множеств формулой: $T_1 \times T_2 = \{(t_1, t_2) : t_1 \in T_1, t_2 \in T_2\}$. А как определить произведение областей T_1, T_2, если не все элементы областей известны? Имеется следующий вариант определения декартова произведения в теории категорий.

Пусть T_1, T_2 — объекты категории $\mathcal{C}$. Декартово произведение $T_1 \times T_2$ — это объект вместе с морфизмами (проекциями) $pr_1 : T_1 \times T_2 \to T_1$ и $pr_2 : T_1 \times T_2 \to T_2$, обладающие следующими свойствами: для любой пары морфизмов $f_1 : T \to T_1$, $f_2 : T \to T_2$ есть морфизм $(f_1, f_2) : T \to T_1 \times T_2$, композиция которого с проекциями дает равенства $pr_1 * (f_1, f_2) = f_1$ и $pr_2 * (f_1, f_2) = f_2$, и, наоборот, любой морфизм вида $f : T \to T_1 \times T_2$ представляется в виде $f = (pr_1 * f, pr_2 * f)$.

В этом определении декартово произведение объектов задается набором операций: произведение объектов $T_1 \times T_2$, проекции $pr_1(T_1, T_2)$, $pr_2(T_1, T_2)$, произведение морфизмов (f_1, f_2), которые по областям T_1, T_2 и морфизмам f_1, f_2 строят соответствующие объект и морфизмы. Кроме того, задаются определяющие соотношения между этими операциями.

Сходным способом могут быть определены аналоги многих других теоретико множественных конструкций. В теории категорий (см., например [16]) определен некоторый набор категорных операций, моделирующих полный набор обычных теоретико множественных операций. Математический объект с этой системой операций называется в теории категорий топосом.

В категорном подходе к представлению понятий, в соответствии со стандартной категорной идеологией, области представляются объектами, отображения — морфизмами, отношения — подобъектами, а предикаты — морфизмами в выделенный объект Ω, который называется классификатором подобектов. Объект Ω имеет два выделенных элемента $true : I \to \Omega$ и $false : I \to \Omega$. Свое название объект Ω получил в связи с требованием, чтобы для любого подобъекта $A \subset B$ существовал единственный "характеристический" морфизм $\chi(A) : B \to \Omega$, область истинности которого, то есть предел диаграммы

$$\begin{array}{ccl} & & B \\ & & \downarrow \chi(A) \\ I & \xrightarrow{true} & \Omega \end{array}$$

совпадает с подобъектом A.

В общем случае, объект Ω может иметь другие элементы $I \to \Omega$, отличные от $true$ и $false$, то есть другие значения истинности, связанные с неполнотой представленной информации в данной категории. В связи с этим объект Ω будет также называться областью истин.

2.3 Конечные задания категорий и конечные аппроксимации категорий

В алгебраическом топосе для каждой аксиомы элементарного топоса вместо требования существования объекта или морфизма с требуемыми свойствами полагается, что этот объект или морфизм строится соответствующей операцией. Алгебраические топосы сопоставляемые представляемому понятию задаются конечным множеством образующих и определяющих соотношений. Существование таких топосов следует из следующего утверждения.

У т в е р ж д е н и е 2.1. Аксиоматика алгебраического топоса может быть задана хорновскими формулами вида $P_1 \wedge ... \wedge P_n \ \rightarrow \ P$, где P_i — элементарные формулы, выражающие либо равенство объектов, либо равенство морфизмов, а P — также либо равенство, либо условие применимости одной из операций алгебраического топоса.

Доказательство утверждения носит технический характер и здесь не приводится. Оно состоит в вводе операций над объектами и морфизмами, требуемых в определении топоса и проверке того, что все условия накладываемые в аксиоматике топоса на соотношения между этими операциями, и условия их применимости имеют хорновский вид.

Как известно (см., например, [54]), среди моделей теории с аксиомами, заданными в виде хорновских формул, всегда найдется наименьшая (инициальная) модель, которая может быть индуктивно построена как фактор множество правильно построенных термов из имен операций по индуктивно построенному отношению эквивалентности на термах. Эта конструкция позволяет задавать алгебраические топосы, сопоставляемые представляемым понятиям, конечным множеством образующих — именами областей, отображений и отношений — и конечным набором соотношений в виде хорновских формул относительно равенства.

Однако требование машинной реализации этих категорий предъявляет более сильные требования к конечности представления, точно такие же, как к машинной реализации целых чисел: возможно конечное определение множества целых чисел, но в машине целые числа представляются в виде машинных целых чисел, которые образуют конечное множество. Аналогично возникает задача о конечной аппроксимации категорий, т.е. представления категорий в виде конечных множеств с некоторой точностью.

Примерами конечных аппроксимаций конечно заданных категорий являются конечные множества термов этой категории и конечные подотношения эквивалентности, полученные на каком-либо конечном шаге индуктивного построения категории. Аппроксимация считается более точной, если она получена на более позднем шаге индуктивного построения категории.

Заметим, что категория, сопоставляемая представляемому понятию, является идеальным (как правило, бесконечным и с алгоритмически неразрешимой проблемой равенства) математическим объектом, отражающим полное знание о понятии, а конечная аппроксимация этой категории отражает знания полученные о представляемом понятии и доступные пользователю. Предполагается, что "хорошие" аппроксимации содержат все, что обычно используется при обращении к аппроксимируемому понятию. С другой стороны, предполагается, что в "хороших" системах имеются возможности расширения аппроксимации, если в этом возникнет необходимость, по ходу использования понятия автоматически или самим пользователем, исходя из знаний прикладной области. В аппроксимациях должны быть отражены уже известные полезные знания, которые могут быть введены в систему в виде теорем и их доказательств, либо представлены в виде правил переписывания термов.

Особая роль имеет использование аппроксимации для распознавания эквивалентности термов. Так как категория $\mathcal{C}$ является фактор множеством термов по некоторому отношению эквивалентности, то возникает задача о каноническом представлении областей и морфизмов категории $\mathcal{C}$ среди эквивалентных термов.

Поясним эту задачу на примере. Если в категории $\mathcal{C}$ представляется тип данных целые числа, то в множестве правильно построенных выражений имеются следующие эквивалентные выражения: $2*10+4$; $(8+4)*2$; $3*8$. Однако, когда мы спрашиваем, чему равно $(8 + 4) * 2$, то в ответ хотим получить один выделенный элемент в классе эквивалентных выражений, содержащих выражение $(8+4)*2$. Вычислить арифметическое выражение $(8 + 4) * 2$, это значит выдать более простое (в каком-то смысле) или каноническое выражение, равное данному. Если каноническое представление арифметических выражений есть их представление в десятичном исчислении, то в результате мы ожидаем получить $2 * 10 + 4$. Если система исчисления другая, например восьмеричная то результат будет $3 * 8$. Таким образом, понятие вычисления термов тесно связано с понятием канонической системы термов.

Выделенный элемент в классе эквивалентных (синонимичных) термов будем называть дескриптором или каноническим термом этого класса. В теории вычисления в абстрактных типах данных его принято называть каноническим [46, 27].

О п р е д е л е н и е 2.1. Множество термов $C \subset Term_\Sigma(\emptyset)$ называется канонической системой термов абстрактного типа данных $\mathcal{A} =< Name, \Sigma, E >$, если выполнены следующие условия:

для всякого $t \in Term_\Sigma(\emptyset)$ существует $c \in C$ такой, что $t \cong_E c$;

для любых $c_1, c_2 \in C$ из соотношения $c_1 \cong_E c_2$ следует, что термы c_1 и c_2 совпадают.

Т е о р е м а 2.2. Если C — каноническая система термов, то отображение, сопоставляющее элементу $c \in C$ его класс эквивалентности $[c] \in I(\mathcal{A})$, является взаимно однозначным отображением.

Доказательство теоремы очевидно.

О п р е д е л е н и е 2.2. Вычислением значений термов в I(A) при выбранной канонической системе термов C называется функция $calc : Term_\Sigma(\emptyset) \to C$, сопоставляющая $t \in Term_\Sigma(\emptyset)$ элемент $calc(t) \in C$ такой, что $t \cong_E calc(t)$.

Так как вычисление термов должно проводиться на машине, то хотелось бы иметь алгоритм определения значений функции $calc$. Однако доказано, что в общем случае такого алгоритма не существует. Имеются примеры алгебр, заданных конечной сигнатурой и конечным множеством определяющих соотношений, для которых проблема равенства двух термов алгоритмически неразрешима. Поэтому можно надеяться в получении единой техники вычислений лишь для случаев, когда это возможно, либо можно получить лишь частичное вычисление, когда система термов уточняется в процессе пополнения знаний об известных соотношениях в множестве термов.

Поэтому выделение дескрипторов в аппроксимации категории производится либо пользователем системы как при определении понятия, так и в диалоге с системой во время получения ответов на запросы, либо автоматически в процессе построения конечной аппроксимации. В последнем случае выделение дескрипторов может происходить, например, по следующему правилу: терм является дескриптором в данной аппроксимации категории, если все остальные эквивалентные ему в аппроксимации термы получены на более поздних шагах индуктивного построения категории.

Среди способов вычисления термов выделяют два способа: вычисление снизу по шагам в той последовательности, как это указано в терме, и переписывание термов — упрощение их до канонического вида.

В первом способе предполагается, что функция $calc()$ уже определена на термах вида $F()$, где F— имя нульарной операции, и термах вида $F(c_1, ..., c_n)$, где F— имя n-арной операции и $c_1, ..., c_n \in C$. Для конечного фрагмента алгебры или категории эта функция может быть представлена в некоторой базе данных. Тогда на произвольном терме вида $F(t_1, ..., t_n)$, где $t_1, ..., t_n \in Term_\Sigma(\emptyset)$, функция $calc$ определяется следующим индуктивным правилом:

$$calc(F(t_1, ..., t_n)) = F(calc(t_1), ..., calc(t_n)).$$

Именно этим способом обычно вычисляются значения арифметических выражений.

Объемы данных, необходимые для хранения аппроксимации категории в таком виде, могут быть очень велики, поэтому возникает проблема хранения аппроксимации в памяти ЭВМ. Представляется, что это хранение должно быть организовано средствами баз данных.

Если такой системе предъявляется терм категории $\mathcal{C}$, то, последовательно применяя операции категории в порядке, указанном термом, система, используя таблицы результатов операций, найдет дескриптор, соответствующий терму, если он входит в данную аппроксимацию категории. Для проверки равенства двух термов система вычисляет дескрипторы этих термов в данной аппроксимации. Если дескрипторы совпадают, то термы эквивалентны. В противном случае в данной аппроксимации они не эквивалентны.

Другой способ вычисления термов основывается на понятиях подстановки термов в терм, подтерма и системы правил переписывания термов. В [25] дан обзор теории систем переписывания термов. Эта теория была успешно применена в системе AFFIRM [27] для вычислений в инициальных моделях абстрактных типов данных.

В конкретных реализациях систем использующих данный подход, возможны различия в выборе того, какая часть аппроксимации действительно хранится в том или ином виде, а какая часть вычисляется с помощью правил переписывания. Это связано с эффективностью системы и с конструкторскими соображениями и знаниями в предметной области о представляемом понятии. В любом случае, на запросы пользователей система должна выдавать ответы в виде выражений из дескрипторов.

Теперь несколько слов о запросах к аппроксимации категории и построении ответов. Рассмотрим для примера следующие запросы: "Чему равно A?","Равны A и B?", "Какие элементы области D известны системе?". В этих запросах A, B — термы, задающие область или морфизм категории $\mathcal{C}$, а D — терм, задающий область в $\mathcal{C}$.

Ответом на первый вопрос является дескриптор терма A в данной аппроксимации. Если A не входит в аппроксимацию, то система найдет первое неизвестное ей применение операции в терме A и выдаст в ответ сообщение, что значение терма A неизвестно, так как неизвестно значение такой-то операции на таких-то дескрипторах.

Второй вопрос распадается для системы на следующие вопросы: "Чему равно A?", "Чему равно ?", "Совпадают дескриптор A и дескриптор B".

В ответ на третий вопрос, если дескриптор D является дескриптором области, выдается множество всех дескрипторов подобластей области D, изоморфных ТОЧКЕ.

Учитывая выразительные возможности операций рассматриваемых категорий, уже эти типы запросов позволяют построить большое разнообразие запросов к системе. Запрос типа "Чему равно A?" позволяет, например, проводить вычисления и преобразование термов. Частным случаем запроса "Равны A и B?" является запрос об истинности замкнутой формулы в категории $\mathcal{C}$ (при моделировании логик в топосах, замкнутой формулой по определению называется терм, задающий морфизм из области ТОЧКА в область истин Ω [16]). В этом случае в запросе нужно положить $A =$ и $B =$ — выделенный элемент в области Ω.
Таким образом, в язык запросов входит любая замкнутая формула языка категорий, которая может быть получена из образующих и определяющих соотношений категории.

К запросу типа: "Какие элементы области D известны системе?", относятся запросы: "Выдать всех поставщиков завода X", где D — область "поставщики завода X", или запросы: "Выдать все химические элементы, удовлетворяющие условию P", где D — это область "химические элементы, удовлетворяющие условию P".

Все перечисленные запросы являются частными случаями запросов к реляционной базе данных, "хранящей" аппроксимацию категории $\mathcal{C}$ в виде набора отношений (для каждой категорной операции свое отношение) между дескрипторами. В общем случае возможен произвольный запрос языка запросов реляционной модели к этой базе данных. Ответами на такие запросы являются отношения из дескрипторов данной аппроксимации, получающиеся в результате применения операций запроса к базисным отношениям аппроксимации. Отношения - ответы трактуются как известные в данной аппроксимации наборы дескрипторов, удовлетворяющие условиям запроса. Пустые отношения трактуются как "неизвестно".

3 Операции над типами данных и параметрические типы данных

В данном разделе предлагается некоторое общее определение понятия операции над типами данных в рамках алгебраического подхода к типам данных в программировании [12], [52], [53].

Пусть имеются две модели A и A' абстрактного типа данных $\mathcal{A}$ (см. определения 1.4, 1.5 этой главы). Напомним, что взаимно однозначное отображение $h : A \to A'$ несущих множеств моделей называется изоморфизмом, если оно переводит функции и отношения одной модели в соответствующие функции и отношения другой модели.

О п р е д е л е н и е. 3.1. Операцией α над типами данных из набора абстрактных типов данных $\mathcal{A}_1, ..., \mathcal{A}_n$ в абстрактный тип данных $\mathcal{B}$ называется сопоставление наборам моделей $A_1, ..., A_n$, соответствующих абстрактным типам данных $\mathcal{A}_1, ..., \mathcal{A}_n$, некоторой модели $\alpha(A_1, ..., A_n)$ абстрактного типа данных $\mathcal{B}$. При этом требуется, чтобы такое сопоставление удовлетворяло следующему условию естественности: по набору $h = (h_1, ..., h_n)$ изоморфизмов $h_1 : A_1 \to A'_1, ..., h_n : A_n \to A'_n$ между моделями абстрактных типов данных операция строит изоморфизм $\alpha(h_1, ..., h_n)$ между результатами — моделями $\alpha(A_1, ..., A_n)$ и $\alpha(A'_1, ..., A'_n)$ абстрактного типа данных $\mathcal{B}$. Требуется, кроме того , чтобы это сопоставление $h \mapsto \alpha(h)$ было функториальным, т.е. тождественному изоморфизму сопоставлялся тождественный изоморфизм, а композиции изоморфизмов - композиция изоморфизмов.

Таким образом, операция α по реализациям $A_1, ..., A_n$ абстрактных типов данных $\mathcal{A}_1, ..., \mathcal{A}_n$ строит реализацию B абстрактного типа данных $\mathcal{B}$, а ее естественность означает, что по переименованию элементов исходных типов данных она должна построить переименование элементов в результате операции.

Нетрудно видеть, что обычные конструкции структурных типов такие, как $ARRAY$, $RECORD$,$SETOF$, $LIST\_OF\_T$ являются операциями в смысле приведенного выше определения.

Наиболее важными примерами операций над типами данных из абстрактного типа $\mathcal{A}$ в абстрактный тип $\mathcal{B}$ являются операции, индуцированные корректными логическими интерпретациями [32] абстрактного типа $\mathcal{B}$ в тип $\mathcal{A}$, рассмотренными как теории в исчислении предикатов. Рассмотрим здесь более подробно частный случай логической интерпретации —

морфизм абстрактных типов [46].

О п р е д е л е н и е. 3.2. Морфизмом $h : \Sigma \to \Sigma'$ сигнатуры $\Sigma = (S, OP)$ в сигнатуру $\Sigma' = (S', OP')$ называется пара функций

$$h = (h_S : S \to S', h_{OP} : OP \to OP')$$

таких, что для каждого функционального символа $F \in OP$ типа $F : T_1 \times ... \times T_n \to T_k$ выполняется соотношение

$$h_{OP}(F) : h_S(T_1) \times ... \times h_S(T_n) \to h_S(T_k).$$

Если задан морфизм сигнатур $h : \Sigma \to \Sigma'$, то он определяет преобразование термов в сигнатуре Σ в термы в сигнатуре Σ' по следующему правилу: переменные X типа T в терме t преобразовываются в переменные $h(X)$ типа $h_S(T)$, а функциональные символы F в терме t преобразовываются в функциональные символы $h_{OP}(F)$. Нетрудно видеть, что в результате такого преобразования получается терм. Этот терм будет обозначаться выражением $h(t)$.

О п р е д е л е н и е. 3.3. Для двух абстрактных типов данных $\mathcal{B} = (S, OP, E)$ и $\mathcal{A} = (S', OP', E')$ морфизм сигнатур $h : (S, OP) \to (S', OP')$ называется морфизмом абстрактных типов данных $h : \mathcal{B} \to \mathcal{A}$, если для каждого уравнения $(r = l) \in E$ уравнение $h(r) = h(l)$ выводится из определяющих соотношений E' абстрактного типа данных $\mathcal{A}$.

Пусть $h : \mathcal{B} \to \mathcal{A}$ — морфизм абстрактных типов данных. Операцией обратного образа h^* над данными $\mathcal{A}$ (в литературе эту операцию называют также забывающим функтором) из абстрактного типа данных $\mathcal{A}$ в абстрактный тип данных $\mathcal{B}$, индуцированной морфизмом h, называется операция, которая модели A абстрактного типа данных $\mathcal{A}$ сопоставляет модель $B = h^*(A)$ абстрактного типа данных $\mathcal{B}$ по следующему правилу:

– B_T — множество элементов сорта $T \in OP$ алгебры B совпадает с множеством элементов $A_{h(T)}$ алгебры A;

– для любого имени операции $F \in OP$ соответствующая ей операция алгебры B совпадает, как функция, с операцией $h_{OP}(F)$ алгебры A.

Естественность операции h^*, то есть функториальность относительно изоморфизмов моделей $\mathcal{A}$, очевидна.

Для морфизма абстрактных типов данных $h : \mathcal{B} \to \mathcal{A}$ можно определить другую операцию над типами, которая обозначается h_* и называется операцией прямого образа типа данных $\mathcal{B}$, индуцированной морфизмом h.

Она сопоставляет каждой модели B абстрактного типа данных $\mathcal{B}$ модель $A = h_*(B)$ абстрактного типа данных $\mathcal{A}$, которая строится как свободная $\mathcal{A}$ -алгебра, порожденная множествоми элементов B_T, рассматриваемых как элементы типа $h_S(T)$, для всех $T \in OP$ с дополнительными соотношениями $h_{OP}(F)(b_1, ..., b_n) = F(b_1, ..., b_n)$ для любых $F \in OP$ и $b_1, ..., b_n \in B$, для которых эти соотношения имеют смысл. Естественность (функториальность) операции h_* также очевидна.

В литературе [46] особое внимание уделяется морфизмам $h : \mathcal{B} \to \mathcal{A}$, удовлетворяющим соотношению $h^*(h_*(B)) = B$ для любой модели B абстрактного типа данных $\mathcal{B}$. Смысл этого соотношения в том, что операция h_* свободного расширения алгебры B по морфизму h не портит исходной алгебры. Такие морфизмы называются сохраняющими (persistent).

При задании операций, индуцированных морфизмами абстрактных типов данных, желательна программная поддержка проверок, что данный морфизм сигнатур является морфизмом абстрактных типов данных или что данный морфизм является сохраняющим. Такую поддержку могут осуществить автоматизированные системы доказательства, основанные на алгоритме Кнута - Бендикса [61], или на методе резолюций, или на их сочетании.

В общем случае операции, индуцированные морфизмами абстрактных типов данных, позволяют расширять множество сигнатурных символов и множество определяющих соотношений абстрактного типа данных за счет определений и следствий из определяющих соотношений. Этими операциями можно выполнить любое сужение множества сигнатурных символов и множества определяющих соотношений, а также переименование сигнатурных символов и т.д.

Операции, индуцированные логическими интерпретациями (морфизмами), дают примеры унарных (одноместных) операций над типами данных. Пример нульарной операции I для алгебраических абстрактных типов данных $\mathcal{A}$ дает операция $I(\mathcal{A})$, выделяющая инициальную модель $\mathcal{A}$.

Все операции над типами данных можно свести к нульарным и унарным с помощью следующей операции объединения типов. Пусть $\mathcal{A}_1, ..., \mathcal{A}_n$ - произвольные абстрактные типы данных, а $\mathcal{B}$ - абстрактный тип данных, множества сигнатурных символов и соотношений которого есть разъединенные объединения соответствующих множеств исходных типов $\mathcal{A}_1$,...,$\mathcal{A}_n$. Операция объединения типов $\mathcal{A}_1$,...,$\mathcal{A}_n$ набору их моделей $A_1, ..., A_n$ сопоставляет этот же набор $A_1, ..., A_n$, рассматриваемый как модель абстрактного типа $\mathcal{B}$.

Используя понятие операции, индуцированной морфизмом абстрактных типов данных, можно дать следующее определение спецификации абстрактного параметрического типа данных [46].

О п р е д е л е н и е. 3.4. Синтаксически параметрический абстрактный тип данных задается четырьмя абстрактными типами данных и четырьмя морфизмами:

PAR— абстрактным типом данных, описывающим параметры;

IMP — абстрактным типом данных, описывающим внешние типы данных, которые используются в данной спецификации;

EXP — абстрактным типом данных, в котором приведены сигнатурные символы и соотношения, доступные для внешних пользователей данной спецификации;

$BODI$ — абстрактный тип данных, описывающий построение параметрического типа данных;

$i : PAR \to IMP$ — морфизм - вложение параметров во внешние типы данных;

$j : PAR \to EXT$ — морфизм, связывающий типы данных параметров с типами данных, выдаваемых спецификацией;

$s : IMP \to BODI$ — морфизм, устанавливающий соответствие между внешними импортируемыми типами данных и типами данных в теле параметрического типа данных;

$h : EXP \to BODI$ — морфизм, выделяющий выдаваемые спецификацией типы данных в теле параметрического типа.

При этом требуется, чтобы следующая диаграмма была коммутативна:

$$\begin{array}{ccc} PAR & \xrightarrow{j} & EXP \\ i\downarrow & & \downarrow h \\ IMP & \xrightarrow{s} & BODI \end{array}$$

То есть выполняется соотношение $h \circ j = s \circ i$.

Спецификация параметрического абстрактного типа данных называется семантически корректной, если $s : IMP \to BODI$ является сохраняющим (persistent) морфизмом.

По спецификации параметрического абстрактного типа данных строится операция из абстрактного типа данных IMP в абстрактный тип данных EXP по следующему правилу: реализации A абстрактного типа данных

IMP сопоставляется реализация $B = h^*(s_*(A))$ абстрактного типа данных EXP, где h^* — забывающий функтор по морфизму h, а s_* — функтор, строящий свободную алгебру по мофизму s и алгебре A.

Другими примерами операций над типами данных являются хорошо известные операции декартова произведения, разъединенного объединения, построения множества всех преобразований из одного типа в другой, массивов элементов какого-либо типа, конечных множеств элементов некоторого типа и т.д., то есть операции, часто используемые в программировании. Многие из них могут быть описаны в виде конструкции, описанной выше.

Так, например, операция примера 1.4 — тип данных LIST_OF_T, приведенный в подразделе 1.2, описывается операцией индуцированной морфизмом абстрактных типов данных. В нем абстрактный тип данных $\mathcal{B} = (T, \emptyset, \emptyset)$ без функциональных символов и соотношений, а абстрактный тип данных $\mathcal{A}$ задан спецификацией LIST_OF_T примера 1.4.

Нетрудно видеть, что композиции операций над типами данных также являются операциями над типами данных.Таким образом, мы имеем клон операций. Исследование этого клона как алгебраической системы — одна из важных задач развития механизма типов в программировании.

Литература

[1] Агафонов В. Н. *Спецификация программ: понятийные средства и их организация.* Новосибирск: Наука,Сиб.отд-ние,1990.

[2] Агафонов В. Н. *Типы и абстракция данных в языках программирования.* В кн. Данные в языках программирования. М.:Мир,1982, с.263–327.

[3] Бениаминов Е. М. *Алгебраический подход к моделям баз данных реляционного типа.* В кн.:Семиотика и информатика, 1980, вып.14, с.44-80.

[4] Бениаминов Е. М. *Алгебраическая стуктура реляционных моделей баз данных.* НТИ, сер.2, 1980, N9, с. 23-25.

[5] Бениаминов Е. М., Березина Н. А. *Об алгебраическом подходе к описанию схем баз данных.*// В сб. Вопросы создания Автоматизированной системы НТИ по документам ГАФ СССР, Москва: ГАУ при Совете Министров СССР, ВНИИДАД, 1981, с.69-77.

[6] Бениаминов Е. М. *О роли симметрии в реляционных моделях баз данных и логических структурах.* НТИ, сер.2, 1984, N5, с.17-25.

[7] Бениаминов Е. М., Березина Н. А., Дунская М. В. *Разработка методов моделирования автоматизированной обработки, поиска и размещения данных в больших информационных системах.* СИФ ОЦНТИ, ВНИИДАД, депонированная рукопись, инв. N031–85,М.,1985.

[8] Бениаминов Е. М. *О некотором подходе к представлению знаний.* Тезисы докладов 4 Всесоюзн.конференции "Применение методов мат. логики" г. Таллин, 1986, с.34-36.

[9] Бениаминов Е. М. *Основания категорного подхода к представлению знаний. Категорные средства.* Изв. АН СССР Техн. кибернет.,N 2, 1988 , с.21–33.

[10] Бениаминов Е. М. *Рефлексивные топосы в категорном подходе к представлению знаний.* Тезисы докладов Всесоюзн. школы-семинара

"Семиотические аспекты формализации интеллектуальной деятельности" г. Боржоми, 1988, с.111-113.

[11] Бениаминов Е.М., Вайнтроб А. Ю. *Основные принципы диалогового языка для представления знаний средствами категорного подхода.* Материалы конференции ДИАЛОГ-87, г. Тбилиси, 1988, с.174-177.

[12] Бениаминов Е. М. *Алгебраические системы и типы данных.* //В кн.:Системное и теоретическое программирование, Ростов-н-Д: РГУ, 1988, с.83-92.

[13] Бениаминов Е. М. *Система представления и обработки понятий, основанная на алгебраическом (категорном) подходе.* Труды II Всесоюзной конференции "Искусственный интеллект- 90", т.2, 1990, с.8-11.

[14] Вигнер П. *Программирование на языке АДА.* М.:Мир, 1983.

[15] Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л. *Алгебра, логика, языки, программирование.* Киев:Наукова думка, 1974.

[16] Голдблатт Р. *Топосы. Категорный анализ логики.* М.:Мир, 1983.

[17] Гончаров С. С., Ершов Ю. Л., Свириденко Д. И. *Семантические основы логического программирования*// В сб. Концептуализация и смысл, под ред. Полякова И. В., Новосибирск:Наука, 1990, с.6-20.

[18] Джонстон П. Т. *Теория топосов.* М.:Наука, 1986.

[19] Жожикашвили А.В., Стефанюк В.Л. *Теория категорий в задачах представления знаний и обучения.* Изв. АН СССР. Техн. кибернет., N 2, 1986.

[20] Замулин А. В. *Системы программирования баз данных и знаний.* Новосибирск:Наука, 1990.

[21] Калиниченко Л. А. *Методы и средства интеграции неоднородных баз данных.* М.:Наука, 1983.

[22] Калиниченко Л. А., Рывкин В. М. *Машины баз данных и знаний.* М.:Наука, 1990.

[23] Капитонова Ю. В., Летичевский А. А. *Математическая теория проектирования вычислительных систем.* М.:Наука, 1988.

[24] Кондрашина Е. Ю., Литвинцева Л. В., Поспелов Д. А. *Представление знаний о времени и пространстве в интеллектуальных системах.* М.:Наука, 1989.

[25] Кучеров Г. А. *Системы подстановок термов.* Препринт 601, ВЦ АН СССР Сиб. отделение, Новосибирск, 1985.

[26] Мальцев А. И. *К общей теории алгебраических систем.* Мат. сборник, 1954, т.35, вып.1.

[27] Массер Д. *Спецификация абстрактных типов данных в системе AFFIRM.//* В сб.: Требования и спецификации в разработке программ, М.:Мир, 1984, с.199-222.

[28] Плоткин Б. И. *Универсальная алгебра, алгебраическая логика и базы данных.* М.:Наука, 1991.

[29] Поспелов Д. А. *Логико-лингвистические модели в системах управления.* М.:Энергоиздат, 1981.

[30] Свириденко Д. И. *Проект Сигма. Цели и задачи.* // В сб. Логические методы в программировании под ред. Ершова Ю. Л. (Вычислительные системы, вып. 133), РАН, Сиб. отд.-ние, Ин.-т математики, Новосибирск, 1990, с.68-94.

[31] Цаленко М. Ш. *Моделирование семантики в базах данных.* М.:Наука, 1989.

[32] Шенфилд Дж. *Математическая логика.* М.:Наука, 1975.

[33] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 1)* Comput. Lingust. Comput. Lang., 13, 1979, p. 152-282.

[34] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 2)* Comput. Lingust. Comput. Lang., 14, 1980, p. 43-65.

[35] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 3)* Lect. Notes in Comp. Sci., V.117, Springer-Verlag, Berlin, 1981, p. 281-290.

[36] Banchilhon F. *On the Compliteness of Query Language for Relational Data Bases.* Lect. Notes in Comp. Sci., V.64, Springer-Verlag, 1978, pp.76-98.

[37] Beniaminov E. M. *Concept Bases and Algebraic Modeling Methods.* Proceedings of the International Workshop on Advances in Databases and Information Systems (ADBIS'94), 1994, p.133-135.

[38] Beniaminov E. M. *A Categorical Approach to Knowledge Representation.* Japan-CIS Symposium on Knowledge Based Software Engineering'94 (JCKBSE'94), 1994, p.181-182.

[39] Beniaminov E. M. *Algebraic Invariants of Database Schemes.* Proceedings of the Second International Workshop on Advances in Databases and Information Systems (ADBIS'95), V.1, 1995, p.259-263.

[40] Breu R. *Algebraic Specification Techniques in OOP Environments.* Lect. Notes in Comp. Sci., V.562, Springer-Verlag, 1991.

[41] Burstall R. M., Goguen J. A. *The Semantics of Clear, a Specification Language* In: Proceedings, of the 1979 Copenhagen Winter Shcool on Abstrakt Software Specification, Springer-Verlad, Berlin, 1980, 292-332.

[42] Chandra A. K., Harel D. *Computable queries for relational data bases.* J. Comput. & Syst. Sci., V.21, N2, 1980, p.156-172.

[43] Codd E. F. *A relational model for large shared data banks.* Comm. of ACM 13, 6, 1970, p.377-387.

[44] Dershowitz N. *Orderings for term-rewriting systems.* Theor. Comput. Sci., 1982, V.17, 3, pp.279-302.

[45] Elmasri R., Weeldrever J., Hevner A. *The category concept: an extension to the entity-relatconship model* Data & Knowledge Engineering, V.1, N1,1985.

[46] Ehrig H., Mahr B. *Fundamentals of Algebraic Specification.* V.1, Springer-Verlag, 1985.

[47] Ehrig H., Mahr B. *Fundamentals of Algebraic Specification.* V.2, Springer-Verlag, 1990.

[48] *First International Symposium on Category Theory Applied to Computation and Control* Lect. Notes in Com. Sci, V.25, Springer - Verlad, Berlin, 1975.

[49] Georgescu I. *A Categorial approach to knowledge-based systems.* Computers and Artifical Intelligence, V.3, N2, 1984, pp.105-113.

[50] Goguen J. A., Thatchar J. W., Wagner E. G. *An inititial algebra approach to the specification, correctness and implementation of abstract data types.* In Current Trends in Programming Methodology IV: Data Structuring, Prentice Hall, 1978, pp.80-144.

[51] Goguen J. A. *Some design principles and theory for OBJ-O, a language to express and execute algebraic specifications of problems.* In: Lect. Notes Comput. Sci.,V.75, 1979, pp.425-473.

[52] Goguen J. A., Burstall R. M. *Some fundamental algebraic tools for the semantics of computation.* Theoretical Computer Science, V.31, N2, 3, 1984.

[53] Goguen J. A., Burstall R. M. *Introducing Institution.* Lect. Not. Comp.Sci., V.164, 1984.

[54] Goguen J. A., Meseguer J. *Equality, types, modules, and (why not?) generics for logic programming.* Conf. of Logical Prog., Uppsala, Sweden, 1984, 179-210.

[55] Goguen J. A., Meseguer J. *Eqlog: Equality, types, and generic modules for logic programming.* In Douglas SeGroot and Gary Lindstrom, eds. "Logic Programming: Functions, Relations and Equations", Prentice-Hall, Englewood Cliffs, N.J., 1986, pp.295-363.

[56] Goguen J. A., Meseguer J. *Order-sorted algebra solves the constructor selector, multiple representation and coercion problems.* Symposium on Logic in Comp. Sci., IEEE Comp. Society Press, 1987, pp.18-29.

[57] Goguen J. A., Burstall R. M. *Institutions: Abstract Model Theory for Specification and Programming.* Jorn. of ACM, V.39, 1, 1992, pp.95-146.

[58] Grothendieck A., Verdier J. L. *Théorie des Topos.* (SGA 4, exposés I-VI).–Second edition.–Berlin; Heidelberg; N. Y.:Springer, 1972.

[59] Huet G. *Confluent reductions: abstract properties and applications to term rewriting systems.* Journ. of ACM, 1980, v.27, N4, pp.797-821.

[60] Kaplan S. *Symplifying conditional term rewriting systems: unification, termination and confluence.* Jorn. Symbolic Computation 4(3), 1987, pp.295-334.

[61] Knuth D., Bendix P. *Simple word problems in universal algebras.* In: Conputational Problems in Abstract Algebra, Pergamon Press, Elmsford, New York, 1970, pp.263-297.

[62] Krasner M. I. *Generalization et analogues de la theorie de Galois.* Comptes Rendus de Congress de la Victorie de I'Ass. Franc. pour I'Avancem. Sci., 1945, pp. 54-58.

[63] Lawvere F. W. *Functorial semantics of algebraic theories.* Proc. Nat. Acad. Sci., 1963, V.50, N5, pp.869-872.

[64] Lawvere F. W. *Introduction* In: Toposes, Algebraic Geometry and Logic, Lect. Notes in Math., V.174, 1972.

[65] Maltsev A. *Algebraic Systems.* Springer-Verlag, 1973.

[66] Melton A., Schmidt D., and Strecher G. *Galois connections and computer science applications.* Lect. Notes in Comp. Sci., V. 240, Springer-Verlag, 1986.

[67] *Proceedings, Category Theory and Computer Programming,* Lect. Notes in Comp. Sci., V. 240, Springer-Verlag, 1986.

[68] *Proceedings, Category Theory and Computer Science,* Lect. Notes in Comp. Sci. V. 283, Springer-Verlag, 1987.

[69] Rydeheard D.F., Burstall R.M. *Computational category theory.* Prentice Hall, 1988.

[70] Tuijn C., Gyssens M., Paredaens J. *A Categorical Approach to Object-Oriented Data Modelling.* Proceedings of Third Workshop on Foundation of Models and Languages for Data and Objects, Aigen, 1991, pp.187–196.

[71] Rusinowitch M., Rémy J. L. (Eds.) *Conditional Term Rewriting Systems.* Lect. Notes in Comp. Sci., V. 656, Springer-Verlag, 1993.

[72] Zilles S.N. *Introduction to data algebras.* Lect. Notes Comput. Sci., V.86, Springer-Verlag, 1980.

Глава 3. Алгебраические средства представления понятий

1 Алгебраическая структура топосов конечного типа

Требование представления сложных понятий с использованием декартовых произведений или областей, состоящих из функций,а также использование подобъектов и отношений для областей, в которых не все элементы известны, приводит к необходимости использования соответствующих конструкций топоса [18]. В этом разделе рассматриваются топосы с конечным множеством образующих и некоторым условием конечности объектов. Показано, что такие топосы (с некоторым указанным ниже условием булевости, соответствующим требованию, что в топосе реализуется классическая логика) представляются в виде конечного произведения топосов конечных G-множеств $GFinset$ для некоторого набора конечных групп G. В частности показывается, что каждый подтопос топоса конечных множеств $Finset$, порожденный конечным набором функций и отношений эквивалентен категории G-множеств, где G — группа симметрии этих функций и отношений.

В приложениях топосов к моделированию схем баз данных особую роль играют булевы топосы.

О п р е д е л е н и е 1.1. Топос называется булевым, если классификатор подобъектов Ω с выделенным элементом $true : I \to \Omega$, где I — финальный объект категории, изоморфен $I \sqcup I$ — копроизведению I и I с выделенным элементом $true$ — вложением на первое слагаемое.

В литературе булевы топосы называют также классическими, так как в этом случае (и только в этом случае) соотношения между логическими операциями — преобразованиями Ω, являются соотношениями классической логики. Как известно [16], в общем случае в топосе реализуются соотношения интуиционистской логики.

Для задач представления знаний с использованием средств вычислительной техники подходят не произвольные категории, а лишь те, которые могут быть конструктивно заданы. В связи с этим рассматривается понятие алгебраического топоса, заданного конечным множеством образующих и соотношений и удовлетворяющего следующему условию конечности.

Пусть T — некоторый объект топоса $\mathcal{C}$. Через T^n обозначается декартова степень объекта T , через $T^n(x_1, ..., x_n)$ отношение в топосе $\mathcal{C}$ с переменными $x_1, ..., x_n$, совпадающее со всем T^n. Напомним, что отношение в топосе — это произвольный подобъект в декартовом произведении соответствующих объектов.

Через $\Delta_{i,j}(T^n)$ обозначим область истинности утверждения $(x_i = x_j)$ на объекте T^n. Подобъект $\Delta_{i,j}(T^n)$ представляет собой диагональное вложение T в i-ый и j-ый сомножители T^n, домноженное на T по остальным сомножителям.

О п р е д е л е н и е 1.2. Будем говорить, что объект T имеет мощность меньше n, если область истинности утверждения $\bigvee\limits_{i \neq j}(x_i = x_j)$, где дизъюнкция берется по всевозможным парам неравных чисел $i, j \in \{1, ..., n\}$, задающего морфизм $T^n \to \Omega$ в область истин Ω, совпадает со всем T^n. Или, другими словами, выполняется равенство

$$T^n = \bigcup_{i \neq j} \Delta_{i,j}(T^n),$$

где объединение подобъектов берется по всевозможным парам неравных чисел $i, j \in \{1, ..., n\}$. Объект T в топосе $\mathcal{C}$ называется конечным, если найдется некоторое натуральное число n такое, что T имеет мощность меньше n.

З а м е ч а н и е 1.1. В определении 1.1 отражается в топосах на категорном языке формула

$$\forall x_1 \in T ... \forall x_n \in T(\bigvee_{i \neq j}(x_i = x_j)) = true,$$

которая означает, что в T нет n различных элементов.

О п р е д е л е н и е 1.3. Алгебраический топос $\mathcal{C}$ называется топосом конечного типа, если он булев имеет конечное множество образующих и все его объекты конечны.

О п р е д е л е н и е 1.4. Топос называется полным или двузначным, если его область истин (классификатор подобъектов) Ω имеет лишь два элемента $true : I \to \Omega$ и $false : I \to \Omega$.

Условие полноты топоса соответствует тому, что любое утверждение (формула, выражающая морфизм из I в Ω) в полном топосе либо истинно, либо ложно, то есть полнота топоса соответствует полноте теории в исчислении предикатов.

Топос конечных множеств $Finord$, объектами которого являются пустое множество и множества вида $\{1, 2, ..., n\}$ для любого натурального n, а морфизмами — любые отображения между ними, является примером полного топоса конечного типа.

Другие примеры получаются следующим образом. Пусть G — конечная группа. Рассмотрим категорию $GFinset$. Объектами категории $GFinset$ являются конечные множества M вместе с выделенным действием группы G на них, то есть для каждого элемента g группы G определено отображение $g : M \to M$, и произведению элементов $g_1 \circ g_2$ соответствует композиция отображений $g_1 : M \to M$ и $g_2 : M \to M$. Морфизмами категории $GFinset$ являются все отображения G-множеств $f : M \to N$, сохраняющие действием группы G, то есть $f(g(m)) = g(f(m))$ для любых $g \in G$ и $m \in M$.

Нетрудно проверить, что категория $GFinset$ образует топос. Действительно, одноточечное множество с тривиальным действием группы G на нем является финальным объектом I. Классификатор подобъектов Ω представляет собой двуточечное множество с тривиальным действием группы G. Декартово произведение и дизъюнктное объединение в категории $GFinset$, а также уравнитель и фактор совпадают с соответствующими операциями в категории конечных множеств при естественном определении действия группы G на получающихся множествах. Экспоненциалом $M \Rightarrow N$ двух G-множеств является множество всех отображений $h : M \to N$ из M в N, на котором элементы $g \in G$ действуют по правилу: $g(h)$ — это композиция трех отображений

$$g(h) : M \xrightarrow{g^{-1}} M \xrightarrow{h} N \xrightarrow{g} N,$$

то есть $g(h) = g \circ h \circ g^{-1}$.

У т в е р ж д е н и е 1.1. Алгебраический топос с конечным множеством образующих, эквивалентный категории $GFinset$, является полным топосом конечного типа.

Доказательство очевидно.

Категория $GFinset$ дает нам пример топоса с классической логикой, в котором каждое утверждение либо истинно, либо ложно, так как это полный топос. Поэтому, например, для любого объекта M из этого топоса можно выразить пустой он или нет и, даже, определить мощность объекта M. Но, если группа G действует на M без неподвижных точек, то в этом топосе мы не сможем выразить элемент объекта M. Это связано с тем, что элементы в топосе по определению есть морфизмы из I в M, то есть соответствуют в топосе $GFinset$ неподвижным точкам в G-множестве M.

Таким образом, о категории областей типов данных, эквивалентной категории $GFinset$, можно сказать, что в ней о данных мы знаем все, кроме способа указания некоторых элементов в областях. С точки зрения представления знаний это также можно сформулировать следующим образом: если категория, отражающая наши знания о представляемом фрагменте мира, эквивалентна категории $GFinset$, то это соответствует полному знанию об этом фрагменте с точностью до симметрии, которая не позволяет нам как-то выделять элементы среди симметричных.

Утверждение 1.2. Категории $G_1 Finset$ и $G_2 Finset$ эквивалентны тогда и только тогда, когда изоморфны группы G_1 и G_2.

Доказательство. Группу G_1 можно рассматривать как объект категории $G_1 Finset$, на которой G_1 действует умножениями слева. Объект G_1 не имеет собственных подобъектов (атомарен), так как G_1 действует на себе транзитивно.

Пусть $F : G_1 Finset \to G_2 Finset$ — эквивалентность категорий. Тогда, так как G_1 — непустой неразложимый в прямую сумму объект категории $G_1 Finset$, то $F(G_1)$ — непустой неразложимый объект категории $G_2 Finset$, то есть непустое G_2-множество, имеющее одну траекторию действия G_2 на нем. Отсюда следует существование эпиморфного G_2-отображения $\alpha : G_2 \to F(G_1)$. Применяя к α функтор F^{-1}, получим эпиморфное G_1-отображение $F^{-1}(\alpha) : F^{-1}(G_2) \to F^{-1}(F(G_1))$ на объект $F^{-1}(F(G_1))$, который по определению эквивалентности категорий изоморфен G_1. Композиция эпиморфизма $F^{-1}(\alpha)$ и этого изоморфизма дает эпиморфное отображение $\beta : F^{-1}(G_2) \to G_1$. Аналогичные рассуждения, переставляя местами G_1 и G_2, показывают существование эпиморфного G_1-отображения $\alpha' : G_1 \to F^{-1}(G_2)$. Так как композиция $\beta \circ \alpha' : G_1 \to G_1$ — изоморфизм, то α' — моноэпиморфизм и, следовательно, α' — изоморфизм между G_1 и $F^{-1}(G_2)$.

Рассмотрим, теперь, множество морфизмов $F^{-1}(G_2)$ в себя. С одной стороны, так как $F^{-1}(G_2)$ изоморфен G_1, то множество морфизмов $F^{-1}(G_2)$ в себя образует относительно операции композиции группу, изоморфную G_1. С другой стороны, так как F^{-1} — эквивалентность категорий, это множество биективно множеству морфизмов G_2 в себя в категории $G_2 Finset$, и, следовательно, относительно той же операции композиции морфизмов образует группу, изоморфную G_2. Отсюда следует требуемый в утверждении 1.2 изоморфизм G_1 и G_2.

Оказывается, категории вида $GFinset$ исчерпывают примеры полных топосов конечного типа.

Т е о р е м а 1.3. Если $\mathcal{C}$ — полный топос конечного типа, то существует такая конечная группа G, что категория $\mathcal{C}$ эквивалентна категории $GFinset$.

Прежде, чем перейти к доказательству самой теоремы, сформулируем и докажем несколько вспомогательных утверждений.

У т в е р ж д е н и е 1.4. Произведение непустых объектов в полном топосе непусто.

Д о к а з а т е л ь с т в о. Пусть T_1, T_2 — объекты полного топоса. Рассмотрим образ морфизма $I(T_1) : T_1 \to I$, где I — финальный объект топоса. Этот образ является подобъектом в I. По определению полного топоса, подобъекты в I — это либо пустой объект $\emptyset$, либо само I. Если образ $I(T_1)$ пуст, то T_1 отображается на пустой объект и, следовательно, по определению пустого объекта, ему изоморфен, что противоречит условию непустоты T_1. Отсюда следует, что образ $I(T_1)$ есть I, и $I(T_1)$ — эпиморфизм. Аналогично показывается, что $I(T_2)$ — эпиморфизм. Так как в топосе каждый эпиморфизм является фактор объектом по отношению эквивалентности, то произведение эпиморфизмов — эпиморфизм [18]. Следовательно, морфизм $I(T_1) \times I(T_2) : T_1 \times T_2 \to I \times I \approx I$ — эпиморфизм, и $T_1 \times T_2$ не может быть пустым объектом.

У т в е р ж д е н и е 1.5. Для объекта мощности n полного булева топоса не существует $n+1$ непустых попарно не пересекающихся подобъектов $T_i \subset T$, $i = 1, ..., n+1$.

Д о к а з а т е л ь с т в о. Предположим, что такие подобъекты существуют. Тогда декартово произведение $C = T_1 \times ... \times T_{n+1}$ содержится в декартовой степени T^{n+1}. Согласно утверждению 1.4 объект C непуст. С другой стороны, так как пересечения $T_i \cap T_j$ пусты, то C не пересекается ни с одной из диагоналей $\Delta_{i,j}(T^{n+1})$, и, следовательно, объединение этих диагоналей не пересекается с C, то есть, по определению 1.2, мощность объекта T не меньше $n+1$, что противоречит условию утверждения 1.5. Следовательно, предположение было неверным, и утверждение доказано.

У т в е р ж д е н и е 1.6. Если объект T в полном булевом топосе имеет конечную мощность, то булева алгебра $P(T)$ всех его подобъектов конечна.

Д о к а з а т е л ь с т в о. Пусть n — мощность объекта T. Предположим, что булева алгебра подобъектов T бесконечна. Тогда существует 2^{n+1} различных подобъектов в T. Рассмотрим конечную булеву подалгебру $B \subset P(T)$, порожденную этими подобъектами. Пусть k — число атомов конечной булевой алгебры B. Так как по теореме Стоуна конечные булевы алгебры изоморфны булевым алгебрам всех подмножеств множества ее атомов, то число элементов в булевой алгебре B равно 2^k. Так как по построению булева алгебра B содержит 2^{n+1} различных элементов, то отсюда следует, что $k > n$. То есть булева алгебра $P(T)$ содержит $k > n$ непустых попарно не пересекающихся элементов, но это противоречит утверждению 1.5. Следовательно, булева алгебра $P(T)$ конечна.

О п р е д е л е н и е 1.5. Объект называется атомарным, если он непуст и у него нет подобъектов, отличных от него самого или пустого подобъекта.

У т в е р ж д е н и е 1.7. Каждый конечный объект в полном булевом топосе представляется в виде канонической прямой суммы своих атомарных подобъектов.

Д о к а з а т е л ь с т в о. Это утверждение следует из предыдущего утверждения о конечности булевой алгебры всех подобъектов конечного объекта и теоремы Стоуна о представлении единицы конечной булевой алгебры в виде канонической суммы ее атомарных элементов.

Итак, пусть теперь T — произвольный объект полного топоса конечного типа, и n — мощность объекта T. Рассмотрим булеву алгебру подобъектов декартовой степени T^n. Пусть $X = \{x_1, ..., x_n\}$ — множество переменных. Рассмотрим в T^n дополнение к областям истинности предикатов $x_i = x_j$ (диагоналям) для всех пар разных переменных из X. Из определения мощности конечного объекта T следует, что так построенный подобъект непуст и, следовательно, (так как T^n — это тоже конечный объект и, по утверждению 1.7, алгебра подобъектов $P(T^n)$ — конечная булева алгебра) содержит атомарный (минимальный непустой) подобъект $A_T \subset T^n$.

О п р е д е л е н и е 1.6. Атомарный подобъект $A_T \subset T^n$, построенный по конечному объекту T полного булевого топоса по правилу, описанному выше, назовем максимальным атомарным объектом над объектом T.

У т в е р ж д е н и е 1.8. Пусть T — произвольный объект полного топоса конечного типа, и A — произвольный атомарный объект, для которого существует морфизм $j : A \to A_T$ в определенный выше атомарный объект A_T. Тогда множество морфизмов $\mathcal{C}(A, T)$ состоит из n элементов, где n — мощность объекта T. То есть $|\mathcal{C}(A, T)| = |T|$.

Д о к а з а т е л ь с т в о. По построению A_T, композиции морфизмов $A_T \stackrel{i}{\to} T^n \stackrel{pr\_k}{\to} T$, где i — каноническое вложение подобъекта A_T, и pr_k — проекция декартова произведения на k-ый множитель, различны для различных $k = 1, ..., n$. Так как образ атомарного объекта должен быть атомарным, то $j : A \to A_T$ — эпиморфизм, и, следовательно, все морфизмы $pr_k \circ i \circ j : A \to T$ различны для $k = 1, ..., n$.

С другой стороны, множество морфизмов $\mathcal{C}(A, T)$ находится во взаимно однозначном соответствии с множеством морфизмов $(i \circ j, f) : A \to T^n \times T$, где f — произвольный морфизм из $\mathcal{C}(A, T)$, так как по определению произведения пары $(i \circ j, f)$ и $(i \circ j, f')$ равны, если они равны покоординатно. Образом A относительно морфизма $i \circ j$ является A_T. Обозначим через A' атом — образ атома A относительно морфизма $(i \circ j, f)$. Пусть $p_{A'}(x_1, ..., x_n, x')$ предикат (морфизм из $T^n \times T$ в Ω) с подобластью истинности $A' \subset T^n \times T$. По построению подобласти A_T и предиката $p_{A'}$ выполняются равенства $p_{A'}(x_1, ..., x_n, x') \wedge (x_i \neq x_j) = p_{A'}(x_1, ..., x_n, x')$ для любых пар разных переменных из множества $X = \{x_1, ..., x_n\}$. Так как область истинности предиката $p_{A'}$ атомарна, то предикат $p_{A'} \wedge (x_i \neq x')$ либо ложен, либо совпадает с $p_{A'}$. Если выполняется равенство $p_{A'} \wedge (x_i \neq x') = p_{A'}$, то будем говорить, что у предиката $p_{A'}$ значения переменных x_i и x' не равны. Если выполняется равенство $p_{A'} \wedge (x_i \neq x') = false$, то будем говорить, что у предиката $p_{A'}$ значения переменных x_i и x' равны. Из определения мощности следует, что не существует предикатов на $T^n \times T$ с непустой областью истинности, у которого значения переменных попарно не равны для числа переменных больше, чем n. Поэтому для предиката $p_{A'}$ среди переменных $x_1, ..., x_n, x'$ обязательно есть переменные, принимающие равные значения. Так как $x_1, ..., x_n$ попарно не равны, то переменная x' в предикате $p_{A'}$ равна некоторой переменной из $x_1, ..., x_n$, например x_k. То есть коммутативна следующая диаграмма морфизмов:

$$\begin{array}{ccccc} A & \stackrel{j}{\to} & A_T & \stackrel{i}{\longrightarrow} & T^n \\ || & & || & & \downarrow (id(T^n), pr_k) \\ A & \stackrel{j}{\to} & A_T & \stackrel{(i,f)}{\longrightarrow} & T^n \times T \end{array}$$

Тем самым доказано, что любой морфизм $f : A \to T$ совпадает с $pr_k \circ i \circ j$ для некоторого k. Что завершает доказательство утверждения 1.8.

У т в е р ж д е н и е 1.9. Для любого объекта T полного топоса конечного типа $\mathcal{C}$ и любого атомарного объекта A этой категории выполняется следующее неравенство $|\mathcal{C}(A,T)| \leq |T|$, где $|\mathcal{C}(A,T)|$ — мощность множества морфизмов, а $|T|$ — мощность объекта.

Доказательство. Рассмотрим декартово произведение $A \times A_T$, где A_T — атомарный объект, построенный в соответствии с определением 1.6. Так как в полной категории произведение непустых объектов непусто (см. утверждение 1.4), то существует атомарный объект $A' \subset A \times A_T$. Композиция этого вложения с проекцией на первый сомножитель дает морфизм $A' \to A$. Так как образ непустого объекта непуст, и A — атомарный объект, то этот морфизм — эпиморфизм и, следовательно, имеется вложение $\mathcal{C}(A,T) \hookrightarrow \mathcal{C}(A',T)$. Отсюда $|\mathcal{C}(A,T)| \leq |\mathcal{C}(A',T)|$. С другой стороны, композиция вложения $A' \subset A \times A_T$ с проекцией на второй сомножитель дает морфизм $A' \to A_T$. Применяя утверждение 1.8, получим равенство $|\mathcal{C}(A',T)| = |T|$. Сопоставив это равенство с предыдущим неравенством, получим требуемое в утверждении 1.9 неравенство.

Утверждение 1.10. Пусть T — произвольный объект полного топоса конечного типа, и A_T — максимальный атомарный объект над объектом T, построенный по определению 1.6. Тогда для любого атомарного подобъекта $A \subset T^k$ существует эпиморфизм $A_T \to A$, где k — произвольное натуральное число.

Доказательство. Пусть $A \subset T^k$ — произвольный атомарный подобъект объекта T^k. Так как A_T и A — атомы, то они непусты и, согласно утверждению 1.4, непусто их декартово произведение $A_T \times A$. Рассмотрим атом A' в декартовом произведении $A_T \times A \subset T^n \times T^k$. Композиции вложения $A' \subset A_T \times A$ с проекциями дают морфизмы $A' \to A_T$ и $A' \to A$. Так как образ атома в полном булевом топосе является атомом, то, тем самым, построены эпиморфизмы из A' на A_T и A. Для доказательства утверждения достаточно показать, что существует морфизм из A_T на A', тогда композиция этого морфизма с построенным морфизмом из A' на A даст требуемый морфизм.

Для доказательства существования морфизма из A_T на A' введем обозначения. Обозначим через $p_{A'}(x_1, ..., x_n, x'_1, ..., x'_k)$ предикат (морфизм из $T^n \times T^k$ в Ω) с подобластью истинности $A' \subset T^n \times T^k$. По построению подобласти A_T и предиката $p_{A'}$ выполняются равенства

$$p_{A'}(x_1, ..., x_n, x'_1, ..., x'_k) \wedge (x_i \neq x_j) = p_{A'}(x_1, ..., x_n, x'_1, ..., x'_k)$$

для любых пар разных переменных из множества $X = \{x_1, ..., x_n\}$. Так как область истинности предиката $p_{A'}$ атомарна, то предикат $p_{A'} \wedge (x_i \neq$

x'_j) либо ложен, либо совпадает с $p_{A'}$. Если выполняется равенство $p_{A'} \wedge (x_i \neq x'_j) = p_{A'}$, то будем говорить, что у предиката $p_{A'}$ значения переменных x_i и x'_j не равны. Если выполняется равенство $p_{A'} \wedge (x_i \neq x'_j) = false$, то будем говорить, что у предиката $p_{A'}$ значения переменных x_i и x'_j равны.

Из определения мощности следует, что не существует предикатов на $T^n \times T^k$ с непустой областью истинности , у которых значения переменных попарно неравны для числа переменных больше, чем n. Поэтому для предиката $p_{A'}$ среди переменных $(x_1, ..., x_n, x'_j)$, для $j = 1, ..., k$, обязательно есть переменные,принимающие равные значения. Так как $x_1, ..., x_n$ попарно не равны, то переменная x'_j равна некоторой переменной из $x_1, ..., x_n$, которую мы обозначим через $\varphi(x'_j)$.

Рассмотрим морфизм ϕ из декартова произведения $T^n = T_{x_1} \times ... \times T_{xn}$ в декартово произведение $T^{n+k} = T_{x_1} \times ... \times T_{xk} \times T_{x'_1} \times ... \times T_{x'n}$, где $T_{xi}, T_{x'j}$ — экземпляры объекта T, для $i = 1, ..n$ и $j = 1, ..., k$, такой, что композиция морфизма ϕ с проекцией T^{n+k} на T_{xi} совпадает с проекцией T^n на T_{xi}, а композиция морфизма ϕ с проекцией T^{n+k} на $T_{x'j}$ совпадает с проекцией T^n на $T_{\varphi(x'j)}$. По построению, образ подобъекта $A_T \subset T^n$ относительно морфизма ϕ содержится в подобъекте $A' \subset T^{n+k}$. (Заметим, кроме того, что по построению композиция ограничения морфизма ϕ на A_T, то есть $\phi|_{AT} : A_T \to A'$, с морфизмом $A' \to A_T$ дает тождественный морфизм на A_T.) Так как в полном булевом топосе образ атома является атомом, то $\phi(A_T) = A'$. Что завершает доказательство утверждения 1.10.

У т в е р ж д е н и е 1.11. Для любых двух морфизмов $j, j' : A \to A_T$ из произвольного атомарного объекта A полного топоса конечного типа в атомарный объект A_T, построенный по определению 1.6, существует морфизм $g : A_T \to A_T$, что j' равен композиции морфизмов j и g, то есть $j' = g \circ j$.

Д о к а з а т е л ь с т в о. Рассмотрим морфизм $A \to T^n \times T^n$, который является композицией морфизмов $A \overset{(j,j')}{\to} A_T \times A_T$ и $A_T \times A_T \subset T^n \times T^n$, где n — мощность объекта T. Обозначим через A' образ этого морфизма в объекте $T^n \times T^n$. Так как A — атомарный объект, то и $A' \subset T^n \times T^n$ — атомарный подобъект. Повторяя рассуждения доказательства предыдущего утверждения, построим морфизм $\phi : T^n \to T^n \times T^n$, композиция которого с проекцией на первый сомножитель есть тождественный морфизм, и образ подобъекта $A_T \subset T^n$ относительно ϕ

есть A'. Отсюда, нетрудно проверить, что требуемый в утверждении 1.11 морфизм g есть композиция ограничения морфизма ϕ на A_T с проекцией $T^n \times T^n$ на второй сомножитель.

С л е д с т в и е 1.12. Для любого объекта T полного топоса конечного типа произвольный морфизм $h : A_T \to A_T$ атомарного объекта A_T является изоморфизмом, и множество всех морфизмов $\mathcal{C}(A_T, A_T)$ образует группу относительно операции композиции.

Д о к а з а т е л ь с т в о. Так как A_T атомарный объект, и образ атомарного объекта атомарен,то h — эпиморфизм. С другой стороны, если рассмотреть утверждение 1.11 в случае, когда $A = A_T$, $j' = 1_T$ и $j = h$, то получим, что h имеет левый обратный, то есть h — мономорфизм. Следовательно, h — изоморфизм и $\mathcal{C}(A_T, A_T)$ — группа автоморфизмов объекта A_T.

У т в е р ж д е н и е 1.13. Пусть T_1 и T_2 — произвольные объекты полного топоса конечного типа $\mathcal{C}$. Тогда мощность копроизведения $T_1 \sqcup T_2$ равна сумме мощностей объектов T_1 и T_2, то есть $|T_1 \sqcup T_2| = |T_1| + |T_2|$.

Д о к а з а т е л ь с т в о. Введем обозначение $T_3 = T_1 \sqcup T_2$. Рассмотрим атомарный подобъект A в декартовом произведении $A_{T_3} \times A_{T_1} \times A_{T_2}$, где A_{T_3}, A_{T_1}, A_{T_2} — атомарные объекты, построенные по определению 1.6 из соответствующих объектов. Композиции этого вложения с проекциями дадут морфизмы $A \to A_{T_3}$, $A \to A_{T_1}$, $A \to A_{T_2}$. Отсюда и по утверждению 1.8, имеем равенства: $|\mathcal{C}(A, T_1 \sqcup T_2)| = |T_1 \sqcup T_2|$, $|\mathcal{C}(A, T_1)| = |T_1|$, $|\mathcal{C}(A, T_2)| = |T_2|$.

С другой стороны, так как A — атом, то образ морфизма из A в $T_1 \sqcup T_2$ должен лежать либо в T_1, либо в T_2, поэтому $\mathcal{C}(A, T_1 \sqcup T_2) \approx \mathcal{C}(A, T_1) \sqcup \mathcal{C}(A, T_2)$. Отсюда и из предыдущих равенств следует требуемое в утверждении 1.13 равенство.

У т в е р ж д е н и е 1.14. Для любых двух объектов T_1 и T_2 полного топоса конечного типа мощность объекта $T_1 \times T_2$ — произведения объектов T_1 и T_2, равна произведению мощностей объектов T_1 и T_2, то есть $|T_1 \times T_2| = |T_1| * |T_2|$.

Д о к а з а т е л ь с т в о. Введем обозначение $T_3 = T_1 \times T_2$. Рассмотрим атомарный подобъект A декартова произведения атомарных объектов A_{T_3}, A_{T_1} и A_{T_2}, которые строятся для объекта T_3, T_1 и T_2 согласно определению 1.6. Композиции вложения с проекциями являются морфизмами из A в A_{T_3}, A_{T_1} и A_{T_2}. Отсюда, согласно утверждению 1.8, имеем равенства:

$$|\mathcal{C}(A, T_1 \times T_2)| = |T_1 \times T_2|, |\mathcal{C}(A, T_1)| = |T_1| \text{ и } |\mathcal{C}(A, T_2)| = |T_2|.$$

С другой стороны, по определению декартова произведения имеем биекцию

$$\mathcal{C}(A, T_1 \times T_2) \approx \mathcal{C}(A, T_1) \times \mathcal{C}(A, T_2).$$

Отсюда и из предыдущих трех равенств получаем требуемое в утверждении 1.14 равенство.

У т в е р ж д е н и е 1.15. Для любых двух объектов T_1 и T_2 полного топоса конечного типа мощность объекта $T_1 \Rightarrow T_2$ — экспоненциала объектов T_1 и T_2, равна мощности объекта T_2 в степени мощности объекта T_2, то есть $|T_1 \Rightarrow T_2| = |T_2|^{|T_1|}$.

Д о к а з а т е л ь с т в о. Пусть $T_3 = T_1 \times T_2$, A_{T_3}, A_{T_1} и A_{T_2} те же объекты, что и в доказательстве утверждения 1.14. Обозначим через A — атомарный объект над копроизведением атомарных объектов $A_{T_3} \sqcup A_{T_1} \sqcup A_{T_2}$. Согласно утверждению 1.10, где T полагается равным $A_{T_3} \sqcup A_{T_1} \sqcup A_{T_2}$, $A_T = A$ и $k = 1$, существуют морфизмы из A в A_{T_3}, A_{T_1} и A_{T_2}. Отсюда, согласно утверждению 1.8 имеем равенства $|\mathcal{C}(A, T_1)| = |T_1|$ и $|\mathcal{C}(A, T_2)| = |T_2|$. С другой стороны, по определению экспоненциала объектов имеем биекцию множеств

$$\mathcal{C}(A, T_1 \Rightarrow T_2) \approx \mathcal{C}(A \times T_1, T_2).$$

Согласно утверждению 1.7 объект $A \times T_1$ представляется в виде конечной прямой суммы своих атомов, то есть $A \times T_1 = A_1 \sqcup ... \sqcup A_k$. С другой стороны, каждый атом $A_i \subset A \times T_1$ изоморфен A, так как композиция вложения A_i с проекцией $A \times T_1$ на первый сомножитель дает морфизм $q_i : A_i \to A$ из A_i на A, а по утверждению 1.10 существует морфизм $p_i : A \to A_i$ из A на A_i. Обозначим через $g_i : A \to A$ композицию морфизмов p_i и q_i, то есть $g_i = q_i \circ p_i$. Так как, в соответствии со следствием 1.12, все морфизмы из A в A изоморфизмы, то существует g_i^{-1} и $g_i^{-1} \circ q_i \circ p_i = 1_A$. Отсюда, так как p_i имеет левый обратный, морфизм p_i — мономорфизм. С другой стороны, так как образ атомарного объекта атомарен, то p_i — эпиморфизм. Следовательно, A_i изоморфны A, для $i = 1, ..., k$. Таким образом, $A \times T_1 \approx kA$. Для определения числа k рассмотрим подобъект $A \subset A \times T_1$. Композиция вложения с проекцией на первый сомножитель дает тождественный морфизм, а композиция с проекцией на второй сомножитель дает некоторый морфизм $f : A \to T_1$. Отсюда, каждый такой подобъект $A \subset A \times T_1$ является графиком некоторого морфизма. С другой стороны, для любых двух морфизмов $f_1, f_2 : A \to T_1$ их графики являются атомами, изоморфными A, и они либо совпадают (в этом случае $f_1 = f_2$), либо не пересекаются. Таким образом, подобъекты $f : A \to T_1$ находятся во взаимно однозначном соответствии с морфизмами $f : A \to T_1$, и число k равно мощности множества $\mathcal{C}(A, T_1)$, которое по утверждению 1.8 равно $|T_1|$. Итак, доказан изоморфизм $A \times T_1 \approx |T_1|A$. Отсюда и по определению экспоненциала объектов и копроизведения имеем:

$$\mathcal{C}(A, T_1 \Rightarrow T_2) \approx \mathcal{C}(A \times T_1, T_2) \approx \mathcal{C}(|T_1|A, T_2) \approx (\mathcal{C}(A, T_2))^{|T_1|}.$$

Применяя к полученному равенству утверждение 1.8, получим требуемое в утверждении 1.15 равенство $|T_1 \Rightarrow T_2| = |T_2|^{|T_1|}$.

Пусть $\mathcal{C}$ — полный топос конечного типа, и $T_1, ..., T_n$ — конечное множество объектов категории $\mathcal{C}$ таких, что они и некоторые морфизмы между ними являются образующими категории $\mathcal{C}$. Рассмотрим копроизведение этих объектов $D = T_1 \sqcup ... \sqcup T_n$. Так как T_i, для $i = 1, ..., n$, по условию конечные объекты, то и D — конечный объект (утверждение 1.13). Построим атомарный объект A_D категории $\mathcal{C}$ по объекту D в соответствии с определением 1.6. и рассмотрим полную подкатегорию $\mathcal{C}_{A_D}$ категории $\mathcal{C}$, состоящую из объектов T, для которых выполняются равенство $|T| = |\mathcal{C}(A_D, T)|$.

Утверждение 1.16. Подкатегория $\mathcal{C}_{A_D}$ полного топоса конечного типа $\mathcal{C}$ совпадает с $\mathcal{C}$.

Доказательство. Очевидно, что $I \in \mathcal{C}_{A_D}$ и $\Omega = I \sqcup I \in \mathcal{C}_{A_D}$. Докажем, что все образующие $T_1, ..., T_n$ алгебраического топоса $\mathcal{C}$ также принадлежат $\mathcal{C}_{A_D}$. Согласно утверждениям 1.13 и 1.8 имеем равенства $|T_1| + ... + |T_n| = |D| = |\mathcal{C}(A_D, D)|$. С другой стороны, так как A_T — атомарный объект и образ атомарного объекта должен содержаться в одном из слагаемых копроизведения, то

$$|\mathcal{C}(A_D, D)| = |\mathcal{C}(A_D, T_1 \sqcup ... \sqcup T_n)| = |\mathcal{C}(A_D, T_1)| \sqcup ... \sqcup |\mathcal{C}(A_D, T_n)|.$$

По утверждению 1.9 выполняется неравенства $|\mathcal{C}(A_D, T_i)| \leq |T_i|$, для $i = 1, ..., n$. Поэтому полученное ранее равенство $|\mathcal{C}(A_D, D)| = |T_1| + ... + |T_n|$ может выполниться только при условии выполнения равенств $|\mathcal{C}(A_D, T_i)| =$

$|T_i|$, для $i = 1, ..., n$. Таким образом, доказано, что образующие T_i алгебраического топоса $\mathcal{C}$ принадлежат категории $\mathcal{C}_{A_D}$.

В соответствии с определением алгебраического топоса с множеством образующих, для доказательства утверждения 1.16 осталось доказать, что $\mathcal{C}_{A_D}$ — алгебраический топос. То есть осталось доказать, что, если T_1 и T_2 — объекты категории $\mathcal{C}_{A_D}$, то произвольный подобъект $T \subset T_1$, декартово произведение объектов $T_1 \times T_2$ и экспоненциал объектов $T_1 \Rightarrow T_2$ — объекты категории $\mathcal{C}_{A_D}$.

Первое утверждение достаточно очевидно, так как, если бы выполнялось неравенство $|\mathcal{C}(A_D, T)| < |T|$, то выполнялось бы неравенство

$$\begin{aligned} |\mathcal{C}(A_D, T_1)| &= |\mathcal{C}(A_D, T \sqcup T')| = |\mathcal{C}(A_D, T) \sqcup \mathcal{C}(A_D, T')| = \\ &= |\mathcal{C}(A_D, T)| + |\mathcal{C}(A_D, T')| < |T| + |T'| = |T_1|, \end{aligned}$$

где T' — дополнение подобъекта T в T_1. Это неравенство противоречит исходному предположению, что T_1 — объект категории $\mathcal{C}_{A_D}$, то есть выполняется равенство $|\mathcal{C}(A_D, T_1)| = |T_1|$.

Второе утверждение также достаточно очевидно в силу равенства $|T_1 \times T_2| = |T_1| * |T_2|$, которое выполняется по утверждению 1.14, и из равенства

$$\begin{aligned} |\mathcal{C}(A_D, T_1 \times T_2)| &= |\mathcal{C}(A_D, T_1) \times \mathcal{C}(A_D, T_2)| = \\ &= |\mathcal{C}(A_D, T_1)| * |\mathcal{C}(A_D, T_2)| = |T_1| * |T_2|, \end{aligned}$$

которое следует из определения декартова произведения и исходных предположений относительно объектов T_1 и T_2.

Для доказательства оставшегося равенства

$$|\mathcal{C}(A_D, T_1 \Rightarrow T_2)| = |T_1 \Rightarrow T_2|$$

докажем сначала изоморфизм $A_D \times T_1 \approx |T_1| A_D$.

С одной стороны, так как график каждого морфизма $A_D \to T_1$ выделяет подобъект в $A_D \times T_1$, изоморфный A_D, и для разных морфизмов эти подобъекты не пересекаются в силу атомарности A_D, имеем мономорфизм $|\mathcal{C}(A_D, T_1)| A_D \hookrightarrow A_D \times T_1$. Заметим, что по предположению $|\mathcal{C}(A_D, T_1)| = |T_1|$. Для доказательства, что этот мономорфизм является изоморфизмом покажем, что дополнение этого подобъекта пусто. Если бы дополнение было непусто, то, по утверждению 1.13, мощность объекта должна быть строго больше мощности подобъекта, но, согласно утверждению 1.13, мощность подобъекта равна $||T_1|A_D| = |T_1| * |A_D|$, а мощность объекта, согласно утверждению 1.14, $|A_D \times T_1| = |A_D| * |T_1|$. То есть они совпадают, и $A_D \times T_1 \approx |T_1|A_D$.

З а м е ч а н и е 1.2. В [18] рассматривалось понятие локально постоянного конечного объекта в топосе $\mathcal{C}$. Объект T в топосе называется локально постоянно конечным, если существует такой объект V, имеющий глобальный носитель (то есть образ морфизма $V \to I$ в финальный объект совпадает со всем I), что декартово произведение $T \times V$ вместе с проекцией на второй сомножитель изоморфно копроизведению конечного числа финальных объектов в топосе $\mathcal{C}_V$ — локализации топоса $\mathcal{C}$ над объектом V (то есть $T \times V \approx V \sqcup ... \sqcup V$, причем проекция на V переходит в копроизведение тождественных морфизмов $id(V) \sqcup ... \sqcup id(V)$). Доказанный только что изоморфизм $A_D \times T_1 \approx |T_1|A_D$ по сути показывает, если взять $T = T_1$ и $V = A_T$, что в топосе конечного типа все объекты локально постоянно конечны.

Рассмотрим, теперь, $|T_1 \Rightarrow T_2|$. В силу утверждения 1.15 имеем равенство $|T_1 \Rightarrow T_2| = |T_2|^{|T_1|}$. С другой стороны, по определению экспоненциала объектов, в силу доказанного выше изоморфизма, по определению копроизведения и в силу исходного предположения об объектах, имеем равенство:

$$\begin{aligned} |\mathcal{C}(A_D, T_1 \Rightarrow T_2)| &= |\mathcal{C}(A_D \times T_1, T_2)| = |\mathcal{C}(|T_1|A_D, T_2)| = \\ &= |\mathcal{C}(A_D, T_2)|^{|T_1|} = |T_2|^{|T_1|}. \end{aligned}$$

Из последних двух равенств следует требуемое равенство.

Итак, пусть $\mathcal{C}$ — полный топос конечного типа и A_D атомарный объект предыдущего утверждения. Согласно утверждению следствия 1.12 множество всех морфизмов $\mathcal{C}(A_D, A_D)$ образует конечную группу относительно операции композиции. Обозначим эту группу через G. Рассмотрим представимый функтор $F(-) = \mathcal{C}(A_D, -)$ из категории $\mathcal{C}$ в категорию $GFinset$, который каждому объекту T категории $\mathcal{C}$ сопоставляет множество $\mathcal{C}(A_D, T)$, на котором естественно действует группа $G = \mathcal{C}(A_D, A_D)$ композицией соответствующих морфизмов.

Как следует из утверждений 1.16, 1.13, 1.14, 1.15, это точный функтор, который сохраняет копроизведения, пустой объект, экспоненциалы объектов, классификатор подобъектов.

У т в е р ж д е н и е 1.17. Пусть $\mathcal{C}$ — полный топос конечного типа и $F : \mathcal{C} \to GFinset$ — функтор, определенный выше. Для любого морфизма

$f : T_1 \to T_2$ категории $\mathcal{C}$, если $F(f)$ — взаимно однозначное отображение, то f — изоморфизм.

Доказательство. Так как функтор F сохраняет копроизведения, то это утверждение достаточно доказать для атомарных объектов. В этом случае, так как образ атомарного объекта — атомарный объект и T_2 — атомарный объект, то f — эпиморфизм. Чтобы показать, что f — мономорфизм, рассмотрим два морфизма $t_1 : T \to T_1$ и $t_2 : T \to T_1$, композиции которых с морфизмом f совпадают. Так как, согласно утверждению 1.7, всякий объект в категории $\mathcal{C}$ представляется в виде копроизведения конечного числа атомов, то достаточно рассмотреть случай, когда T — атомарный объект. Рассмотрим некоторый морфизм $t : A_D \to T$. Так как T — атомарный объект, то t эпиморфизм, и разным морфизмам t_1, t_2 соответствуют разные морфизмы $t_1 \circ t$, $t_2 \circ t$ — элементы множества $F(T_1)$. Но $F(f)$ по предположению взаимно однозначное отображение, и, следовательно, должны быть различными морфизмы $f \circ t_1 \circ t$ и $f \circ t_2 \circ t$, что противоречит предположению равенства $f \circ t_1 = f \circ t_2$. Таким образом, f — мономорфизм и эпиморфизм и, следовательно, изоморфизм.

Определение 1.7. (Гротендик [58], V 5.1) Категорией Галуа называется пара $(\mathcal{G}, F)$, где $\mathcal{G}$ — малый булев топос, а $F : \mathcal{G} \to Finset$ — точный функтор, отражающий изоморфизмы.

Определение 1.8. Группа называется проконечной, если она представляется в виде предела конечных групп по некоторому направленному множеству.

Имеется следующая теорема.

Теорема (Гротендик [58]). Категория Галуа эквивалентна категории $GFinset$ для некоторой проконечной группы G.

Если $\mathcal{C}$ — полный топос конечного типа, то, согласно утверждению 1.17, пара $(\mathcal{C}, F)$ — категория Галуа и, по теореме Гротендика, он изоморфен топосу $GFinset$ для некоторой проконечной группы G. Утверждение теоремы 1.3 несколько более сильное. В ней утверждается, что в этом случае группа является конечной.

Доказательство теоремы 1.3. Пусть, как и раньше, $\mathcal{C}$ — полный топос конечного типа; $T_1, ..., T_n$ — конечное множество объектов категории $\mathcal{C}$ таких, что они и некоторые морфизмы между ними являются образующими категории $\mathcal{C}$; $D = T_1 \sqcup ... \sqcup T_n$, G — группа автоморфизмов атомарного объекта A_D, построенного по определению 1.6 из объекта D; и $F : \mathcal{C} \to GFinset$ — функтор $F(-) = \mathcal{C}(A_D, -)$.

В утверждениях 1.13–1.17 было показано, что F — точный функтор, сохраняющий копроизведения, экспоненциалы объектов и отражающий изоморфизмы. Для доказательства теоремы 1.3 осталось показать, что для любого G-множества M существует объект T, что $F(T)$ изоморфен M, и для любых двух объектов X_1 и X_2 топоса $\mathcal{C}$, если существует G-отображение из $F(X_1)$ в $F(X_2)$, то существует морфизм $X_1 \to X_2$, представляющий это отображение. Так как функтор F сохраняет копроизведения, то эти утверждения достаточно доказать для атомарных объектов.

Любой атомарный объект M в $GFinset$ имеет вид G/H, где H — произвольная подгруппа в G. Поэтому в качестве требуемого объекта T в этом случае можно взять копредел конечной диаграммы морфизмов $h : A_D \to A_D$, для $h \in H \subset G = \mathcal{C}(A_D, A_D)$, в топосе $\mathcal{C}$, который будет обозначаться через A_D/H.

Если X_1 и X_2 — атомарные объекты в $\mathcal{C}$, то, как следует из утверждения 1.16, они являются образами объекта A_D, и группа G транзитивно действует на $F(X_1)$ и $F(X_2)$, то есть $F(X_1) = G/H_1$, $F(X_2) = G/H_2$. Если есть G-отображение $G/H_1 \to G/H_2$, то H_1 — подгруппа в H_2. Соответственно, $X_1 \approx A_D/H_1$ и $X_2 \approx A_D/H_2$. Так как H_1 — подгруппа в H_2, то диаграмма множества морфизмов $h : A_D \to A_D$, для $h \in H_1$, является поддиаграммой множества морфизмов $h : A_D \to A_D$, для $h \in H_2$, и это вложение диаграмм определяет требуемый морфизм их копределов $A_D/H_1 \to A_D/H_2$ в топосе $\mathcal{C}$. Это завершает доказательство теоремы 1.3.

Рассмотрим, теперь, конечно порожденный алгебраический подтопос топоса $Finset$. Так как такой подтопос булев полон и конечного типа, то, по теореме 1.3, он эквивалентен топосу $GFinset$ для некоторой группы G. Хотелось бы понять, нет ли более явного способа построения группы G.

Итак, пусть подтопос $\mathcal{C}$ топоса $Finset$, порожден конечным множеством несущих множеств $D_1, ..., D_n$, и конечными множествами функций и отношений $f_1, ..., f_k; r_1, ..., r_m$ на множествах, построенных из несущих множеств операциями топоса.

Обозначим, как и в разделе 6 главы 1, через $D = D_1 \sqcup ... \sqcup D_n$ разъединенное объединение несущих множеств, а через $S(D)$ множество всех взаимно однозначных отображений D на себя, отображающих каждое несущее подмножество в себя. Очевидно, что $S(D)$ образует группу относительно композиции отображений.

По сути, каждый элемент $h \in S(D)$ определяет перестановку элементов множества D, и действие этой перестановки можно воспринимать просто, как переименование элементов множества D. Это переименование действует на множествах $D_1, ..., D_n$ и естественно может быть продолжено на все множества, построенные из $D_1, ..., D_n$ операциями топоса. При этом, некоторые множества могут переходить в другие, изоморфные.

Будем говорить, что множество, функция или отношение категории $Finset$ симметрично относительно $h \in S(D)$, если они под действием h переходят в себя.

Обозначим через $S(\mathcal{C})$ множество всех отображений $h \in S(D)$, относительно которых симметричны функции $f_1, ..., f_k$ и отношения $r_1, ..., r_m$ — образующие категории $\mathcal{C}$. Очевидно, что множество отображений $S(\mathcal{C})$ образует подгруппу в $S(D)$, которая мы будем называть группой симметрии образующих топоса $\mathcal{C}$.

Непосредственной проверкой можно доказать, что, если все множества, отношения и функции, используемые в некоторой операции топоса, симметричны относительно h (инвариантны относительно переименования элементов, заданного отображением h), то и результат этой операции инвариантен относительно h. Отсюда следует следующее утверждение.

У т в е р ж д е н и е 1.18. Группа симметрии $S(\mathcal{C})$ образующих топоса $\mathcal{C} \subset Finset$ действует на каждом множестве из категории $\mathcal{C}$, и каждое отображение из категории $\mathcal{C}$ инвариантно относительно этого действия, то есть является $S(\mathcal{C})$-отображением. Другими словами категория $\mathcal{C}$ является подкатегорией топоса $GFinset$, где $G = S(\mathcal{C})$.

На самом деле, это вложение является эквивалентностью топосов, так как саму группу $G = S(\mathcal{C})$ можно построить в $\mathcal{C}$ используя операции топоса, как подобъект $D \Rightarrow D$, удовлетворяющий условиям группы симметрии конечного множества отображений и отношений — образующих топоса $\mathcal{C}$.

В результате получим следующую теорему.

Т е о р е м а 1.19. Любой подтопос $\mathcal{C}$ топоса $Finset$, порожденный конечным множеством несущих множеств $D_1, ..., D_n$, и конечными множествами функций и отношений $f_1, ..., f_k$; $r_1, ..., r_m$ на множествах, построенных из несущих множеств операциями топоса, совпадает с подтопосом G-множеств, где $G = S(\mathcal{C})$ — группа симметрии образующих топоса $\mathcal{C}$.

Рассмотрим, теперь, топосы конечного типа без условия полноты.

Т е о р е м а 1.20. Пусть $\mathcal{C}$ — произвольный топос конечного типа. Тогда существует такой конечный набор конечных групп $G_1, ..., G_k$, что категория $\mathcal{C}$ эквивалентна конечному произведению топосов $G_iFinset$, для $i = 1, ..., k$, то есть $\mathcal{C} \approx G_1Finsen \times ... \times G_kFinset$.

Д о к а з а т е л ь с т в о. Пусть I — финальный объект топоса $\mathcal{C}$. Рассмотрим множество его подобъектов. Так как топос булев, то подобъекты объекта I образуют некоторую булеву алгебру B. Согласно теореме Стоуна каждая булева алгебра B вкладывается в полную булеву алгебру $\bar{B}$, атомами которой являются максимальные ультрафильтры алгебры B.

Напомним, что ультрафильтром булевой алгебры называется множество элементов булевой алгебры B, дополнения которых образуют идеал булевой алгебры B. Другими словами, $\Phi \subset B$ — ультрафильтр, если выполняются следующие условия:

Φ не содержит 0 булевой алгебры;

если $b_1 \in \Phi$, то любой больший элемент $b \supset b_1$ также принадлежит Φ;

если $b_1, b_2 \in \Phi$, то $b_1 \cap b_2 \in \Phi$.

Для любого подобъекта $b \subset I$ рассмотрим подкатегорию $\mathcal{C}_b \subset \mathcal{C}$ объектов над b вида $T \to b$. Если $b2 \subset b1 \subset I$, то стандартным образом определим функтор локализации $L_{b2}^{b1} : \mathcal{C}_{b1} \to \mathcal{C}_{b2}$, который объекту $T \to b1$ сопоставляет предел диаграммы

$$\begin{array}{ccc} & & T \\ & & \downarrow \\ b2 & \subset & b1. \end{array}$$

Известно (см., например, [18]), что функторы локализации L_{b2}^{b1} сохраняют все операции топоса. Кроме того, заметим, что функтор L_b^I переводит образующие топоса $\mathcal{C}$ в образующие топоса $\mathcal{C}_b$, и прямой предел $\mathcal{C}_\Phi$ топосов конечного типа $\mathcal{C}_b$, для $b \in \Phi$, где Φ — максимальный ультрафильтр, является полным топосом конечного типа. Отсюда, по теореме 1.3, каждый топос $\mathcal{C}_\Phi$ с выделенным в нем множеством образующих, соответствующих множеству образующих топоса $\mathcal{C}$, эквивалентен некоторому топосу $GFinset$ для подходящей группы G. Таким образом, $GFinset \approx \mathcal{C}_\Phi$ — это предел топосов $\mathcal{C}_b$ по множеству объектов $b \in \Phi$, принадлежащих максимальному ультрафильтру Φ и $L_\Phi : \mathcal{C} \to \mathcal{C}_\Phi$ — гомоморфизм алгебраических топосов, построенный по определению предела.

Композиции построенных функторов $\mathcal{C} \to \mathcal{C}_\Phi \approx GFinset$ со стирающим функтором $GFinset \to Finset$, который каждому конечному G-множеству сопоставляет его же, но в категории $Finset$, дает функтор $M_\Phi : \mathcal{C} \to Finset$. Функтор M_Φ естественно назвать моделью топоса $\mathcal{C}$ в категории конечных множеств, построенной по максимальному ультрафильтру Φ.

Такие модели строятся по ультрафильтру Φ с точностью до изоморфизма. Напомним, что две модели $M : \mathcal{C} \to Finset$ и $M' : \mathcal{C} \to Finset$ называются эквивалентными, если существует естественное преобразование функторов $h : M \to M'$, что для любого объекта T топоса $\mathcal{C}$ отображение $h(T) : M(T) \to M'(T)$ является взаимно однозначным.

Покажем, теперь, что для топосов конечного типа, множество максимальных ультрафильтров конечно. Пусть Φ и Φ' два различных ультрафильтра топоса конечного типа $\mathcal{C}$. Так как $\Phi \neq \Phi'$, то существует такой элемент $b \in \Phi$, что $b \notin \Phi'$. Отсюда, $L_{\Phi}(b) = I$, где I — финальный объект топоса $\mathcal{C}_{\Phi} \approx GFinset$, и $L_{\Phi'}(b) = \emptyset$, где $\emptyset$ инициальный объект топоса $\mathcal{C}_{\Phi'} \approx G'Finset$. Таким образом, для разных максимальных ультрафильтров Φ и Φ' модели M_{Φ} и $M_{\Phi'}$, построенные по ним, не могут быть изоморфными. С другой стороны, если $T_1, ..., T_n$ — конечное множество объектов категории $\mathcal{C}$ таких, что они и некоторое конечное число морфизмов между ними являются образующими категории $\mathcal{C}$, и M — произвольная модель топоса $\mathcal{C}$, то для набора множеств $D_1, ..., D_n$ той же мощности, что и множества $M(T_1), ..., M(T_n)$, то есть $|D_i| = |M(T_i)|$, для $i = 1, ..., n$, существует модель M', изоморфная модели M, с несущими множествами $M'(T_i) = D_i$, для $i = 1, ..., n$. Такая модель строится просто переименованием элементов у несущих множеств модели $h_i : M(T_i) \to D_i$, где h_i — взаимно однозначное отображение, для $i = 1, ..., n$. Таким образом, каждая модель с фиксированной мощностью несущих множеств изоморфна некоторой модели с несущими множествами $D_1, ..., D_n$, а так как $\mathcal{C}$ имеет конечное множество образующих, которые в модели реализуются элементами некоторых конечных множеств, построенных из множеств $D_1, ..., D_n$, то таких моделей конечное число. Учитывая, что $\mathcal{C}$ — топос конечного типа, и мощности объектов $T_1, ..., T_n$ ограничены сверху, окончательно получим, что множество неизоморфных моделей топоса $\mathcal{C}$ конечное число и, следовательно число максимальных ультрафильтров в булевой алгебре B подобъектов финального объекта I топоса $\mathcal{C}$ конечно.

Из конечности числа максимальных ультрафильтров булевой алгебры B в свою очередь следует, что B — конечная булева алгебра. Пусть $At(B)$ конечное множество всех атомарных элементов булевой алгебры B. Далее, нетрудно видеть, что топос конечного типа $\mathcal{C}$ изоморфен конечному произведению полных топосов конечного типа $\mathcal{C}_a$ по множеству $a \in At(B)$, где B — конечная булева алгебра подобъектов финального объекта топоса $\mathcal{C}$. Это завершает доказательство теоремы 1.20.

2 Рефлексивные топосы

Если категорными средствами нужно отражать знания не только о внешнем мире, но и о самих средствах представления знаний, и о всей системе представления знаний в целом, то понадобится расширение операций топоса, приводящее к понятию рефлексивного топоса, определяемого ниже.

Пусть $\mathcal{C}$ — топос, в котором выделены два объекта NOb и $NMor$, которые называются область имен объектов и область имен морфизмов, соответственно.

Множества элементов объектов NOb и $NMor$, то есть множества $\mathcal{C}(I, NOb)$ и $\mathcal{C}(I, NMor)$, где I — финальный объект категории $\mathcal{C}$, называются множеством имен объектов и множеством имен морфизмов, соответственно.

Пусть имеются также морфизмы между объектами NOb и $NMor$, которые задают структуру внутренней категории [18]. То есть заданы морфизмы

$$e : NOb \to NMor, \quad ndom : NMor \to NOb, \quad ncodom : NMor \to NOb,$$

где e соответствует отображению, сопоставляющему объекту тождественный морфизм, а морфизмы $ndom$ и $ncodom$ соответствуют отображениям, которые морфизму сопоставляют объекты — откуда и куда этот морфизм действует.

По смыслу этих морфизмов между ними должны выполняться соотношения:

$$ndom \circ e = id(NOb), \quad ncodom \circ e = id(NOb),$$

которые означают, что имена области определения и области значений для имени тождественного морфизма совпадают с именем объекта этого морфизма.

Обозначим через $\mathcal{M}2$ объект в категории $\mathcal{C}$, который получается операцией уравнитель пары морфизмов $ndom \circ pr_1$ и $ncodom \circ pr_2$ из произведения объектов $NMor \times NMor$ в NOb, где pr_1 — проекция произведения $NMor \times NMor$ на первый сомножитель, а pr_2 — проекция на второй сомножитель. То есть $\mathcal{M}2$ — это подобъект в $NMor \times NMor$, состоящий из тех пар имен морфизмов, у которых имя области определения первой компоненты пары совпадает с именем области значений второй.

Пусть в категории $\mathcal{C}$ задан морфизм вида

$$com\_n : \mathcal{M}2 \to NMor,$$

который называется композицией имен морфизмов. Предполагается также, что для морфизмов $com\_n, ndom, ncodom, e$ выполняются соотношения, являющиеся аналогами аксиом теории категорий, то есть $Name\_Cat = (NOb, NMor, com\_n, ndom, ncodom, e)$ — внутренняя категория [18] категории $\mathcal{C}$. Тогда множества элементов объектов NOb и $NMor$, то есть множества $\mathcal{C}(I, NOb)$ и $\mathcal{C}(I, NMor)$, где I — финальный объект категории $\mathcal{C}$, и операции на них, индуцированные морфизмами $com\_n, ndom, ncodom, e$, по определению внутренней категории образуют малую категорию. Эту малую категорию обозначим здесь через $NCat$ и назовем категорией имен.

О п р е д е л е н и е 2.1. Топос $\mathcal{C}$ вместе с внутренней категорией $Name\_Cat$ называется топосом с категорией имен.

О п р е д е л е н и е 2.2. Топос $\mathcal{C}$ вместе с внутренней категорией $Name\_Cat$ называется рефлексивным топосом, если существуют функторы $Name : \mathcal{C} \to NCat$ и $Denote : NCat \to \mathcal{C}$ такие, что их композиция $Denote \circ Name = 1\mathcal{C}$ является тождественным функтором на $\mathcal{C}$. Функтор $Name$ называется функтором именования, функтор $Denote$ — денотатом, а $NCat$ — категорией имен топоса $\mathcal{C}$.

Напомним, что топос называется непротиворечивым, если элементы $true$ и $false : I \to \Omega$ в классификаторе подобъектов Ω этого топоса различны.

Т е о р е м а 2.1 Существует непротиворечивый рефлексивный топос.

Д о к а з а т е л ь с т в о. Пусть Set — категория всех множеств, в которой определены все категорные операции алгебраического топоса. Так как теория множеств непротиворечива, то классификатор подобъектов Ω в категории множеств Set — это двуточечное множество.

Обозначим через $\mathcal{T}^1$ свободный алгебраический топос, порожденный из пустого множества образующих операциями алгебраического топоса. Конструкция построения $\mathcal{T}^1$ стандартна, так как аксиоматика алгебраического топоса задается хорновскими формулами. Топос $\mathcal{T}^1$ непротиворечив, так как существуют непротиворечивые топосы, которые являются моделями алгебраического топоса $\mathcal{T}^1$. Примером такого топоса является минимальный алгебраический подтопос $Set^1 \subset Set$ топоса множеств, состоящий из множеств и отображений, построенных только из операций топоса, включая нульарные операции топоса. (Нетрудно видеть, что так как в топосе все операции имеют конечную арность, то топос Set^1 эквивалентен топосу конечных множеств.) Так как $\mathcal{T}^1$ — свободный топос с пустым множеством образующих, то имеется единственный гомоморфизм $j^1 : \mathcal{T}^1 \to Set^1$.

Определим, далее, по индукции алгебраические топосы $\mathcal{T}^k$, подтопосы $Set^k \subset Set$ и гомоморфизмы $j^k : \mathcal{T}^k \to Set^k$, где k — произвольное натуральное число.

Пусть уже определены алгебраический топос $\mathcal{T}^k$ подтопос $Set^k \subset Set$ и гомоморфизм $j^k : \mathcal{T}^k \to Set^k$ для некоторого k.

Рассмотрим множества объектов $Ob\mathcal{T}^k$ и морфизмов $Mor\mathcal{T}^k$ категории $\mathcal{T}^k$. Обозначим через Set^{k+1} наименьший алгебраический подтопос топоса множеств $Set^{k+1} \subset Set$, который содержит множества $Ob\mathcal{T}^k$, $Mor\mathcal{T}^k$ и отображения между ними, определяющие категорию $\mathcal{T}^k$.

Алгебраический подтопос $Set^{k+1} \subset Set$ является алгебраическим топосом с категорией имен (см. определение 2.1), где объектом NOb является множество $Ob\mathcal{T}^k$, объектом $NMor$ является множество $Mor\mathcal{T}^k$.

Построим, теперь, алгебраический топос $\mathcal{T}^{k+1} = \mathcal{T}[Ob\mathcal{T}^k, Mor\mathcal{T}^k]$, как алгебраический топос с категорией имен, порожденный множеством образующих вида: $NT : I \to NOb$ и $Nf : I \to NMor$ для каждого $T \in Ob\mathcal{T}^k$ и $f \in Mor\mathcal{T}^k$, и определяющими соотношениями между этими элементами, которые выполняются между соответствующими элементами множеств $Ob\mathcal{T}^k$, $Mor\mathcal{T}^k$ и отображениями между ними, задающими категорию $\mathcal{T}^k$.

Таким образом, по определению алгебры, заданной образующими и соотношениями, существует единственный гомоморфизм алгебраических топосов с категорией имен $j^{k+1} : \mathcal{T}^{k+1} \to Set^{k+1}$, переводящий образующие $NT : I \to NOb$ и $Nf : I \to NMor$ категории $\mathcal{T}^{k+1}$ в элементы $j^{k+1}(NT) : I \to T \in Ob\mathcal{T}^k$ и $j^{k+1}(Nf) : I \to f \in Mor\mathcal{T}^k$ категории Set^{k+1}. Кроме того, из этого определения гомоморфизма j^{k+1} видно, что он является вложением на множестве образующих. Отсюда следует, что множества $Ob\mathcal{T}^k$ и $Mor\mathcal{T}^k$ инъективно отображается в множества элементов объектов NOb и $NMor$ категории $\mathcal{T}^{k+1}$.

Так как подтопос непротиворечивого топоса — непротиворечив, то $Set^k \subset Set$ — непротиворечивы для всех k. С другой стороны, так как имеется гомоморфизм $j^k : \mathcal{T}^k \to Set^k$, и элементы $true$ и $false$ топоса $\mathcal{T}^k$ переходят в элементы $true$ и $false$ топоса Set^k, соответственно, то они не могут совпадать в $\mathcal{T}^k$. То есть топосы $\mathcal{T}^k$ непротиворечивы.

Определим, теперь, по индукции гомоморфизмы $F_k : \mathcal{T}^k \to \mathcal{T}^{k+1}$

алгебраических топосов с категорией имен.

Так как $\mathcal{T}^2$ является расширением алгебраического топоса $\mathcal{T}^1$, то существует единственный гомоморфизм $F_1 : \mathcal{T}^1 \to \mathcal{T}^2$.

Пусть уже определен функтор $F_{k-1} : \mathcal{T}^{k-1} \to \mathcal{T}^k$. Рассмотрим категорию $\mathcal{T}^k = \mathcal{T}[Ob\mathcal{T}^{k-1}, Mor\mathcal{T}^{k-1}]$. Так как категория $\mathcal{T}^k$ является алгебраическим топосом с категорией имен, порожденным множеством образующих и соотношений, то для задания гомоморфизма $F_k : \mathcal{T}^k \to \mathcal{T}^{k+1}$ достаточно задать его на образующих и проверить выполнение определяющих соотношений. Положим, так как F_k должен быть гомоморфизмом, F_k сопоставляет объектам NOb и $NMor$ категории $\mathcal{T}^k$ объекты NOb и $NMor$ категории $\mathcal{T}^{k+1}$, а образующим топоса $\mathcal{T}^k$ (то есть элементам объектов NOb и $NMor$) — соответствующие элементы объектов NOb и $NMor$ категории $\mathcal{T}^{k+1}$ в соответствии с сопоставлением, определенным функтором F_{k-1}. Определяющие соотношения при этом выполняются, так как по индуктивному предположению F_{k-1} — функтор.

Итак, имеем индуктивную последовательность гомоморфизмов непротиворечивых алгебраических топосов с категорией имен вида:

$$\mathcal{T}^2 \xrightarrow{F_2} \mathcal{T}^3 \xrightarrow{F_3} \ldots \xrightarrow{F_{k-1}} \mathcal{T}^k \xrightarrow{F_k} \ldots$$

Обозначим через $\mathcal{T}^\infty$ индуктивный предел этой последовательности гомоморфизмов, а через $G_k : \mathcal{T}^k \to \mathcal{T}^\infty$ — отображения членов последовательности $\mathcal{T}^k$ в индуктивный предел $\mathcal{T}^\infty$, входящие в определение этого предела. Множество объектов $Ob\mathcal{T}^\infty$ категории $\mathcal{T}^\infty$ представляет собой индуктивный предел множеств $Ob\mathcal{T}^k$ при отображениях $F_k^{Ob} : Ob\mathcal{T}^k \to Ob\mathcal{T}^{k+1}$, а множество морфизмов $Mor\mathcal{T}^\infty$ категории $\mathcal{T}^\infty$ представляет собой индуктивный предел множеств $Mor\mathcal{T}^k$ при отображениях $F_k^{Mor} : Mor\mathcal{T}^k \to Mor\mathcal{T}^{k+1}$, где F_k^{Ob} и F_k^{Mor} — отображения на объектах и морфизмах функтора F_k.

Так как аксиоматика алгебраических топосов с категорией имен задается хорновскими формулами, то нетрудно видеть, что $\mathcal{T}^\infty$ также является алгебраическим топосом с категорией имен, а G_k — гомоморфизмы алгебраических топосов с категорией имен.

По определению индуктивного предела $\mathcal{T}^\infty$ множества объектов NOb и $NMor$ категории $\mathcal{T}^\infty$ являются индуктивными пределами множеств элементов объектов NOb и $NMor$ топоса $\mathcal{T}^k$ относительно отображений, задаваемых функторами F_k. Множества элементов объектов NOb и $NMor$

топоса $\mathcal{T}^k$ содержат, по построению, подмножества, изоморфные $Ob\mathcal{T}^{k-1}$ и $Mor\mathcal{T}^{k-1}$, которые отображаются функтором F_k в элементы объектов NOb и $NMor$ топоса $\mathcal{T}^{k+1}$ в соответствии с функтором F_{k-1}. Таким образом, предел этих подмножеств выделяет в множестве элементов объектов Nob и $NMor$ топоса $\mathcal{T}^\infty$ подкатегорию, изоморфную категории $\mathcal{T}^\infty$. Назовем этот изоморфизм категорий функтором $Name$. Он сопоставляет каждому объекту и морфизму категории $\mathcal{T}^\infty$ элементы объектов NOb и $NMor$ той же категории.

Для построения операции $Denote$ в топосе $\mathcal{T}^\infty$ рассмотрим алгебраический подтопос Set^∞ топоса Set, порожденный множествами $Ob\mathcal{T}^\infty$, $Mor\mathcal{T}^\infty$ и отображениями между ними, задающими категорную структуру $\mathcal{T}^\infty$. Категория Set^∞ представляет собой алгебраический топос с категорией имен, у которого NOb и $NMor$ — это множества $Ob\mathcal{T}^\infty$, $Mor\mathcal{T}^\infty$, рассмотренные как объекты категории Set^∞.

Далее рассмотрим гомоморфизмы алгебраических топосов с категорией имен $L_i : \mathcal{T}^k \to Set^\infty$, которые на образующих (подкатегории $\mathcal{T}^{k-1}$ категории имен $\mathcal{T}^k$) задаются функторами $G_{k-1} : \mathcal{T}^{k-1} \to \mathcal{T}^\infty$.

Покажем коммутативность следующих диаграмм:

$$\begin{array}{ccc} \mathcal{T}^k & \xrightarrow{Fk} & \mathcal{T}^{k+1} \\ || & & \downarrow L_{k+1} \\ \mathcal{T}^k & \xrightarrow{Lk} & Set^\infty \end{array}$$

Так как $\mathcal{T}^k$ задаюся образующими и соотношениями, то для доказательства коммутативности этих диаграмм достаточно их проверить на образующих топоса $\mathcal{T}^k$. По определению функторов F_k, L_k, и L_{k+1}, ограничения рассматриваемых диаграмм на образующие (подмножества категорий имен) представляются диаграммами:

$$\begin{array}{ccc} \mathcal{T}^{k-1} & \xrightarrow{Fk-1} & \mathcal{T}^k \\ || & & \downarrow G_k \\ \mathcal{T}^k & \xrightarrow{G(k-1)} & \mathcal{T}^\infty \end{array}$$

которые коммутативны по определению индуктивного предела $\mathcal{T}^\infty$ и гомоморфизмов G_{k-1}, G_k.

Таким образом, имеем семейство гомоморфизмов $L_k : \mathcal{T}^k \to Set^\infty$, коммутирующих с гомоморфизмами $F_k : \mathcal{T}^k \to \mathcal{T}^{k+1}$. По определению индуктивного предела $\mathcal{T}^\infty$, такое семейство гомоморфизмов индуцирует единственный гомоморфизм алгебраических топосов с категорией имен $L : \mathcal{T}^\infty \to Set^\infty$ такой, что композиция гомоморфизмов L и G_k равна L_k, то есть $L_k = G_k \circ L$.

Определим, теперь, операцию $Denote$, действующую на множествах элементов областей NOb и $NMor$ категории $\mathcal{T}^\infty$ (то есть на категории имен) в множества объектов $Ob\mathcal{T}^\infty$ и морфизмов $Mor\mathcal{T}^\infty$, равной ограничению гомоморфизма $L : \mathcal{T}^\infty \to Set^\infty$ на категорию имен категории $\mathcal{T}^\infty$ в категорию имен топоса Set^∞, которая совпадает с категорией $\mathcal{T}^\infty$.

Для доказательства того, что $\mathcal{T}^\infty$ с определенными выше операциями $Name$ и $Denote$ является рефлексивным алгебраическим топосом, осталось показать выполнение равенств

$$Denote(Name(T)) = T \text{ и } Denote(Name(f)) = f,$$

для любых $T \in Ob\mathcal{T}^\infty$ и $f \in Mor\mathcal{T}^\infty$. Доказательства этих равенств непосредственно следуют из определений индуктивного предела $\mathcal{T}^\infty$ и определений операций $Name$ и $Denote$.

Непротиворечивость $\mathcal{T}^\infty$ следует из непротиворечивости подтопоса Set^∞ непротиворечивого топоса Set и существования гомоморфизма $L : \mathcal{T}^\infty \to Set^\infty$, который элементы $true$ и $false$ переводит в неравные элементы $true$ и $false$ категории Set^∞. Это завершает доказательство теоремы 2.1.

3 Алгебры с условными операциями

В отличие от абстрактных типов данных, которые моделируются многоосновными алгебраическими системами, общие понятия и схемы баз данных моделируются категориями с категорными операциями и требуют привлечения алгебраических средств, в которых операции действуют при выполнении некоторых условий. Примером такой операции является операция композиции отображений: композиция двух отображений f_1 и f_2 определена, если только область значений f_1 совпадает с областью определения f_2, то есть выполняется равенство $codom(f_1) = dom(f_2)$. Многие другие категорные операции также являются условными. Более того, условные операции требуются и для представления понятий, возникающих в предметных областях.

Перейдем к формальным определениям.

Пусть D — некоторое конечное множество, которое мы будем называть словарем алгебры. Элементы множества D называются сигнатурными символами алгебры или терминами. Пусть X некоторое множество переменных. Как обычно, множество алгебраических выражений или термов $Term(D, X)$ со словарем D и множеством переменных X строится индуктивно по правилам:

если $t \in D$ или $t \in X$, то $t \in Term(D, X)$;

если $F \in D$ и $t_1, ..., t_n \in Term(D, X)$, то $F(t_1, ..., t_n) \in Term(D, X)$.

Уравнением называется выражение вида $L = R$, где L, R — пара термов из множества $Term(D, X)$.

Будем предполагать, что с каждым термином $F \in D$ связывается следующая система условий:

$F.cond.eq(x_1, ..., x_n)$ — система уравнений с переменными $x_1, ..., x_n$, задающая условия применения термина F;

$F(x_1, ..., x_n)$ — вид правильно построенного терма со словарным термином F при выполнении условий применения этого термина;

$F.res.eq(x_1, ..., x_n)$ — система результирующих уравнений, задающих связь термина F с другими терминами.

Заметим, что некоторые из этих систем уравнений могут быть тривиальными (пустыми).

Совместно приведенную выше систему условий в примерах будем записывать одним предложением в виде:

"Если $F.cond.eq(x_1,...,x_n)$, то выражение $F(x_1,...,x_n)$ правильно и $F.res.eq(x_1,...,x_n)$".

Кроме того, будем предполагать, что задана система условных соотношений $E=\{e_1,...,e_k\}$, где для каждого $e\in E$ заданы:

$e.cond.eq(x_1,...,x_k)$ — система уравнений, задающая условия применения соотношения e;

$e.res.eq(x_1,...,x_k)$ — система уравнений, определяющая результат соотношения e.

Условное соотношение e в примерах будем записывать в виде предложения:

"Если $e.cond.eq(x_1,...,x_k)$,то $e.res.eq(x_1,...,x_k)$".

О п р е д е л е н и е 3.1. Спецификация $S=(D,E)$ алгебраической системы с условными операциями включает в себя:

D — словарь условных операций, где для каждой операции $F\in D$ определены уравнения — условия применения этой операции, правильно построенный терм с этой операцией и система результирующих уравнений;

E — множество условных соотношений.

Пусть $S=(D,E)$ — спецификация некоторой алгебраической системы, и X — некоторое множество переменных. Определим,теперь, в множестве $Term(D,X)$ подмножество $CTerm(D,E,X)\subset Term(D,X)$, которое будем называть множеством правильно построенных выражений. Построим также отношение эквивалентности $eq_E\subset CTerm(D,E,X)\times CTerm(D,E,X)$ на множестве правильно построенных выражений. Множество правильно построенных выражений $CTerm(D,E,X)$ и отношение эквивалентности на нем eq_E строятся по следующим правилам:

1. если $x\in X$, то $x\in CTerm(D,E,X)$;

2. для любого $F\in D$, если $t_1,..,t_n\in CTerm(D,E,X)$ и для них выполняется условие применения F, то есть $F.cond.eq(t_1,...,t_n)\subset eq_E$, то $F(t_1,...,t_n)\in CTerm(D,E,X)$ и, если термы, входящие в $F.res.eq(t_1,...,t_n)$ правильно построены, то пары термов — левая и правая части уравнений из системы уравнений $F.res.eq(t_1,...,t_n)$, включаются в отношение eq_E;

3. отношение eq_E рефлексивно, симметрично и транзитивно, то есть, если $t\in CTerm(D,E,X)$, то $(t,t)\in eq_E$, если $(t_1,t_2)\in eq_E$, то $(t_2,t_1)\in eq_E$, если $(t_1,t_2)\in eq_E$ и $(t_2,t_3)\in eq_E$, то $(t_1,t_3)\in eq_E$;

4. для любого $F \in D$, если $F(t_1, ..., t_n) \in CTerm(D, E, X)$ и пар термов $(t_1, t'_1), ..., (t_n, t'_n) \in eq_E$, то $F(t'_1, ..., t'_n) \in CTerm(D, E, X)$ и пара $(F(t_1, ..., t_n),\ F(t'_1, ..., t'_n)) \in eq_E$;

5. для каждого условного соотношения $e \in E$ и для каждого набора правильно построенных выражений $t_1, .., t_k \in CTerm(D, E, X)$, удовлетворяющих условию применения соотношения e, то есть условию $e.cond.eq(t_1, ..., t_k) \subset eq_E$, если термы, входящие в $e.res.eq(t_1, ..., t_n)$, правильно построены, то уравнения из системы уравнений $e.res.eq(t_1, ..., t_n)$, включаются в отношение eq_E.

Если пара термов (t_1, t_2) принадлежит отношению eq_E, то это будет выражаться также формулой $t_1 \approx_E t_2$.

О п р е д е л е н и е 3.2. Пусть $S = (D, E)$ — некоторая спецификация. Алгеброй $A(D, E)$ с условными операциями, порожденной словарем терминов D, условиями применения операций $F.cond.eq$ и результирующими соотношениями $F.res.eq$, для $F \in D$, и множеством условных соотношений E называется фактор множество правильно построенных термов $CTerm(D, E, \emptyset)$ по отношению эквивалентности eq_E, построенных представленным выше правилам 1—5.

Будем предполагать, что словарь D алгебраической системы не содержит синонимов.

О п р е д е л е н и е 3.3. Алгебраическая система $A(D, E)$ называется непротиворечивой, если любые два словарных термина $F_1, F_2 \in D$ не эквивалентны относительно эквивалентности eq_E, то есть $F_1 \not\approx_E F_2$.

Рассмотрим несколько примеров спецификаций алгебраических систем с условными операциями.

В первом примере покажем, что инициальная модель абстрактного типа данных $\mathcal{A} = (Name, \Sigma, Eq)$ (см. определения II.1.4 и II.1.8) является примером алгебраической системы с условными операциями.

П р и м е р 3.1. Абстрактный тип данных.

Пусть $\mathcal{A} = (Name, \Sigma, Eq)$ — произвольный тип данных, где $\Sigma = \{s_1, ..., s_n; f_1, ..., f_k\}$ — сигнатура абстрактного типа данных.

В качестве словаря соответствующей алгебраической системы с условными операциями берется множество

$$D = \{true,\ \in,\ s_1, ..., s_n,\ f_1, ..., f_k\}.$$

Множество соотношений и уравнения этого примера будем выписывать отдельными группами, используя некоторые сокращенные формы записей, которые будут вводиться по мере необходимости. Переменные будут обозначаться словами, начинающимися заглавными буквами, или просто заглавными буквами.

Положим: $true$, $s_1, ..., s_n$ — безусловные правильные выражения;

Если $Y = s_i$, то $\in (X, Y)$ — правильное выражение, для $i = 1, ..., n$.

Сокращение: *выражение* $X \in Y$ *является сокращенным выражением уравнения* $\in (X, Y) = true$.

С использованием этого сокращения выпишем некоторые условные соотношения E этого примера.

Если f_j — функциональный символ типа $f_j : s_{i_1} \times ... \times s_{ir} \to s_{kj}$, то для него вписывается следующее выражение. Если $X_{i_1} \in s_{i_1}, ..., X_{ir} \in s_{ir}$, то выражение $f_j(X_{i_1}, ..., X_{ir})$ правильно, и $f_j(X_{i_1}, ..., X_{ir}) \in s_{kj}$

Кроме того, выписываются уравнения Eq, выражающие соотношения абстрактного типа данных $\mathcal{A}$.

Нетрудно видеть, что алгебраическая система с условными операциями, соответствующая спецификации примера 3.1, является инициальной алгеброй термов абстрактного типа данных $\mathcal{A}$ (см. доказательство теоремы II.1.2).

В главе 2 использовались категории, порожденные множеством образующих и соотношений, для представления понятий. В следующем примере дается спецификация такой категории в виде алгебры с условными операциями.

П р и м е р 3.2. Категория $Cat(s_1, ..., s_n, f_1, ..., f_k, Eq)$, заданная образующими объектами $s_1, ..., s_n$, морфизмами $f_1, ..., f_k$ и множеством соотношений Eq между ними.

Словарь терминов в этом примере имеет вид:

$$D = \{true, \in, \ ob, mor, dom, codom, com, id, s_1, ..., s_n, \ f_1, ..., f_k\}.$$

Множество соотношений и уравнения этого примера также будем выписывать отдельными группами, используя некоторые сокращенные формы записей, которые будут вводиться по мере необходимости. Переменные будут обозначаться, как и раньше, словами, начинающимися заглавными буквами, или просто заглавными буквами.

Положим:

$true$, ob, mor, $s_1, ..., s_n, f_1, ..., f_k$ — безусловные правильные выражения;

Если $Ob = ob$, то $\in (X, Ob)$ — правильное выражение.

Если $Mor = mor$, то $\in (X, Mor)$ — правильное выражение.

Сокращение: *выражение $X \in Y$ является сокращенным выражением уравнения $\in (X, Y) = true$.*

С использованием этого сокращения выпишем некоторые условные соотношения E этого примера.

$s_i \in ob$, для $i = 1, ..., n$; $f_j \in mor$, для $j = 1, ..., k$.

Если $F \in mor$, то выражение $dom(F)$ правильно и $dom(F) \in ob$.

Если $F \in mor$, то выражение $codom(F)$ правильно и $codom(F) \in ob$.

Сокращение: *выражение $F : X \to Y$ является сокращенным выражением системы уравнений $F \in mor$, $dom(F) = X$, $codom(F) = Y$.*

Если $X \in ob$, то выражение $id(X)$ правильно и $id(X) : X \to X$.

Если $F \in mor$, $G \in mor$, $codom(F) = dom(G)$, то выражение $com(F, G)$ правильно и $com(F, G) \in mor$, $dom(com(F, G)) = dom(F)$, $codom(com(F, G)) = codom(G)$.

Если $F : X \to Y$, то $com(id(X), F) = F$ $com(F, id(Y)) = F$.

Если $F_1 : X_1 \to X_2$, $F_2 : X_2 \to X_3$, $F_3 : X_3 \to X_4$, то $com(com(F_1, F_2), F_3)) = com(F_1, com(F_2, F_3))$.

Кроме того, выписываются уравнения, выражающие соотношения между морфизмами из множества соотношений Eq.

В этом примере введенные сокращения для систем уравнений позволяют дать определение понятия категории в виде алгебры с условными операциями и условными соотношениями в стандартной для теории категорий форме. Алгебра, заданная этим определением, является стандартной категорией, заданной образующими и соотношениями.

В приложениях для задач представления понятий требуются также категории, у которых на множестве объектов определен порядок, аналогичный порядку включения множеств. Объекты могут иметь элементы, и морфизмы обладают свойством наследования [56, 40]. Кроме того, хотелось бы, чтобы все правильно построенные термы были разбиты по типам. Средствами условных алгебраических систем эти условия выражаются в следующем примере.

П р и м е р 3.3. Категория упорядоченных типов $OrderCat$.

Используется определение $Cat(...)$ из примера 3.2. При этом, пополняется словарь терминов и множество условных соотношений.

Словарь терминов D пополняется элементами $boolean$, $false$, $\subset$, $injection$, $\emptyset$, $appl$.

В алгебре Cat выражение $\in (El, X)$ правильно, только если X совпадает с ob или mor. Так как мы хотим допустить существование элементов у объектов, то необходимо доопределить операцию $\in$ следующими правилами (напомним, что переменные обозначаются словами, начинающимися с прописной буквы, или, просто, прописными буквами)

Если $X \in ob$, то выражение $\in (El, X)$ правильно построено.

Сокращение: *выражение* $El \in X$ *является сокращенным выражением уравнения* $\in (El, X) = true$.

Положим также:

$boolean$, $false$ — правильно построенные безусловные выражения и $boolean \in ob$, $true \in boolean$, $false \in boolean$.

Если $X \in ob$, $Y \in ob$, то выражение $\subset (X, Y)$ правильно и $\subset (X, Y) \in boolean$.

Сокращение: *выражение* $X \subset Y$ *является сокращенным выражением уравнения* $\subset (X, Y) = true$.

Условные соотношения /* отношения включения */:

Если $X \in ob$, то $X \subset X$.

Если $X_1 \subset X_2$, $X_2 \subset X_3$, то $X_1 \subset X_3$.

Если $X_1 \subset X_2$, $X_2 \subset X_1$, то $X_1 = X_2$.

Если $X \subset Y$, то выражение $injection(X, Y)$ правильно и

$injection(X, Y) \in mor$,

$dom(injection(X, Y)) = X$,

$codom(injection(X, Y)) = Y$.

Если $F, G \in mor$; $X, Y \in ob$; $X \subset Y$; $codom(F) = X$, $codom(G) = X$; и $com(F, injection(X, Y)) = com(G, injection(X, Y))$, то $F = G$.

Если $X \subset Y$, $Y \subset Z$, то $com(injection(X, Y), injection(Y, Z)) = injection(X, Z)$.

Если $X \in ob$, то $injection(X, X) = id(X)$.

Условные соотношения:

/* модификация операции композиции морфизмов */

Если $F \in mor$, $G \in mor$, $codom(F) \subset dom(G)$, то выражение $com(F, G)$ правильно построено и

$com(F,G) \in mor$,

$dom(com(F,G)) = dom(F)$,

$codom(com(F,G)) = codom(G)$.

Если $F \in mor,\ \ G \in mor,\ \ codom(F) \subset dom(G)$, то

$com(com(F, injection(codom(F), dom(G)), G) = com(F,G)$.

Если $F, G, H \in mor,\ \ codom(F) \subset dom(G), codom(G) \subset dom(H)$, то $com(com(F,G),H)) = com(F, com(G,H))$.

Условные соотношения /* отношения принадлежности */:

Если $X, Y \in ob,\ \ El \in X,\ \ X \subset Y$, то $El \in Y$.

Условные соотношения /* пустой области */:

Термин $\emptyset$ определяет нульарный безусловный терм и $\emptyset \in ob$, $\in (El, \emptyset) = false$.

Если $X \in ob$, то $\emptyset \subset X$.

Условные соотношения /* термина *appl*, означающего применение функционального терма к элементу области определения */:

Если $El \in X,\ \ F : X \to Y$, то выражение $appl(F, El)$ правильно и $appl(F, El) \in Y$.

Сокращение: *выражение $F(El)$ является сокращенным выражением терма $appl(F, El)$.*

Если $El \in X,\ \ X \subset Y$, то $injection(X,Y)(El) = El$.

Если $F \in mor,\ \ G \in mor,\ \ codom(F) \subset dom(G),\ El \in dom(F)$, то $com(F,G)(El) = G((F(El))$.

В этом примере, в отличие от обычного математического понятия категории (пример 3.2), в объектах выделяются канонические подобъекты, допускается, чтобы объекты имели элементы и некоторые элементы были выразимы в спецификации. Важным для приложений обобщением является также расширение условия применения композиции морфизмов: композиция двух морфизмов определена не только тогда, когда область значений одного морфизма совпадает с областью определения другого, но и когда область значения является подобъектом области определения. Кроме того, элемент подобъекта является элементом самого объекта, поэтому морфизмы, определенные на объекте, могут быть применены и к элементам любого его подобъекта. Таким образом, алгебраическая система этого примера позволяет моделировать и использовать механизм наследования свойств подтипом у надтипа.

Для моделирования более конкретных понятий можно использовать спецификацию предыдущего примера, дополнив ее новыми элементами и соотношениями конкретного понятия.

Для моделирования топосных операций над объектами к спецификации предыдущего примера в словарь следует добавить имена соответствующих операций и соответствующие им соотношения. Для моделирования рефлексии, которая для топосов была рассмотрена в предыдущем разделе, к спецификации примера также нужно добавить соответствующие термины и соотношения.

В задачах представления понятий алгебраическими системами с условными операциями многие вопросы к представленному понятию сводятся к следующей алгебраической задаче. Пусть задана некоторая спецификация алгебраической системы $S = (D, E)$, представляющая некоторое понятие. Нужно уметь отвечать на два основных взаимосвязанных вопроса:

является ли предъявленный вам терм правильно построенным;

эквивалентны ли два тема t_1 и t_2 по отношению эквивалентности eq_E, порожденной соотношениями E.

Так как примерами алгебраических систем $A(D, E)$ являются группы и полугруппы, порожденные множеством образующих и соотношений, и известны примеры групп и полугрупп с алгоритмически неразрешимой проблемой тождества слов, то в общем случае подмножества
$CTerm(D, E, \emptyset) \subset Term(D, E, \emptyset)$ и
$eq_E \subset CTerm(D, E, \emptyset) \times CTerm(D, E, \emptyset)$
могут быть неразрешимыми (но всегда рекурсивно перечислимы). Поэтому в общем случае не существует алгоритма для получения ответов на поставленные выше вопросы и нужно найти практическое решение этих проблем. Таким решением может быть отражение в системе представления (наряду с определениями понятий — спецификациями алгебраических систем) уже известных нам равенств и известных для данного понятия алгоритмов получения новых равенств, вытекающих из спецификации и известных равенств. Набор таких равенств и алгоритмов будем называть аппроксимацией алгебраической системы. Более точно.

О п р е д е л е н и е 3.4. Аппроксимацией $Approx^a A$ алгебраической системы с условными операциями $A(D, E)$ называется следующий набор подмножеств:

$CTerm^a \subset Term(D, \emptyset)$, $NTerm^a \subset Term(D, \emptyset)$ — два алгоритмически разрешимых подмножества в множестве термов со словарем терминов D ;

$eq^a_E \subset Term(D,\emptyset) \times Term(D,\emptyset)$, $Neq^a_E \subset Term(D,\emptyset) \times Term(D,\emptyset)$ — два алгоритмически разрешимых подмножества в множестве всех пар термов со словарем D.

При этом должны выполняться следующие включения:

$$CTerm^a \subset CTerm(D,E,\emptyset), \quad eq^a_E \subset eq_E$$

и, в предположении непротиворечивости алгебраической системы $A(D,E)$, должны выполняться включения:

$$NTerm^a \subset (Term(D,\emptyset) \setminus CTerm(D,E,\emptyset)),$$

$$Neq^a_E \subset (Term(D,\emptyset) \times Term(D,\emptyset)) \setminus eq_E,$$

где символом $\setminus$ обозначена операция разности двух множеств.

Определение 3.5. Пусть построены две аппроксимации алгебраической системы с условными операциями. Аппроксимация $Approx^a = (CTerm^a, NTerm^a, eq^a_E, Neq^a_E)$ алгебраической системы $A(D,E)$ будем называть более точной, чем аппроксимация $Approx^b = (CTerm^b, NTerm^b, eq^b_E, Neq^b_E)$, если они не совпадают и множества аппроксимации $Approx^a$ содержат соответствующие множества аппроксимации $Approx^b$.

Если $Approx^a = (CTerm^a, NTerm^a, eq^a_E, Neq^a_E)$ — некоторая аппроксимация алгебраической системы $A(D,E)$ и t — некоторый терм, то по данной аппроксимации определяется, что терм правильно построен, если $t \in CTerm^a$, и неправильно построен, если $t \in NTerm^a$. В противном случае, правильность построения терма t в данной аппроксимации считается неизвестной. Аналогично, если заданы термы t_1 и t_2, то в данной аппроксимации эти термы считаются эквивалентными, если $(t_1,t_2) \in eq^a_E$, неэквивалентными, если $(t_1,t_2) \in Neq^a_E$, и эквивалентность неизвестна в противном случае.

Таким образом, для задач представления знаний, в базе понятий необходимо хранить как спецификацию алгебраической системы, представляющей некоторое понятие, так и аппроксимацию этой алгебраической системы. Кроме того, для задач представления знаний необходимо, чтобы по аппроксимации можно было бы получать некоторые ответы на следующие вопросы:

для каких значений переменных $x_1,...,x_n$ терм $t(x_1,...,x_n)$ правильно построен в текущей аппроксимации;

для каких значений переменных $x_1, ..., x_n$ терм $t(x_1, ..., x_n)$ в аппроксимации выполняется система уравнений

$$\begin{array}{rcl} L_1(x_1, \ldots, x_n) & = & R_1(x_1 \ldots, x_n) \\ \ldots & \ldots & \ldots \\ L_k(x_1, \ldots, x_n) & = & R_k(x_1 \ldots, x_n), \end{array}$$

где L_i и R_i — некоторые термы от переменных $x_1, ..., x_n$, для $i = 1, ..., k$.

На практике соответствующие множества аппроксимации алгебраической системы могут задаваться средствами баз данных и алгоритмами с указанием правил переписывания термов и ограничением числа переписывания или временем работы алгоритмов. В отличие от систем правил переписывания термов, которые возникают в теории абстрактных типов данных (см. главу 2), здесь должны использоваться системы условных правил переписывания [60, 71], соответствующие условным равенствам алгебраической системы.

Для построения алгоритмов аппроксимаций алгебраических систем с условными операциями полезны следующие утверждения.

Утверждение 3.1. Пусть $S = (D, E)$ — некоторая спецификация и t терм в словаре D. Если терм t правильно построен в спецификации S, то и любой его подтерм правильно построен.

Доказательство. Доказательство утверждения будем проводить методом математической индукции по шагам построения множеств $CTerm(D, E)$ и eq_E.

На первом шаге множества $CTerm(D, E)$ и eq_E, согласно построениям определения 3.2, пусты, и правильно построенный терм может быть построен либо по правилу 1, либо по правилу 2 определения 3.2 с пустым условием применения. И в том и в другом случае, построенный терм, очевидно, удовлетворяет утверждению 3.1.

Пусть уже построены некоторые термы множества $CTerm(D, E)$, и по предположению индукции они удовлетворяют утверждению 3.1.

Предположим, терм t построен на следующем шаге построения множества $CTerm(D, E)$. Тогда, по определению 3.2, терм t может быть построен по правилу 1, 2 или 4. Если терм t построен по правилу 1, то t — переменная, и утверждение 3.1 для него выполнено. Если терм t построен по правилу 2 или 4, то он правильно построен и имеет вид $t = F(t_1, ..., t_n)$, где $F \in D$ и $t_1, ..., t_n$ — некоторые правильно построенные термы, построенные на предыдущих шагах. Так как собственный подтерм терма

$t = F(t_1, ..., t_n)$ совпадает с одним из термов $t_1, ..., t_n$ или их подтермом, то по предположению индукции он правильно построен, и для терма t, в любом случае, утверждение 3.1 верно. Следовательно, исходя из принципа полной математической индукции, утверждение 3.1 верно для всех правильно построенных термов.

Пусть $s(x_1, ..., x_n)$ — произвольный терм, содержащий переменные $x_1, ..., x_n$ и $t_1, ..., t_n$ — набор термов. Через $s(t_1, ..., t_n)$ обозначается результат подстановки $x_1 \mapsto t_1, \dots , x_n \mapsto t_n$ в терм $s(x_1, ..., x_n)$.

Утверждение 3.2. Для любого терма $s(x_1, ..., x_n)$ и набора эквивалентных термов $t_1 \approx_E t'_1, \dots , t_n \approx_E t'_n$ относительно эквивалентности eq_E, из правильности терма $s(t_1, ..., t_n)$ следует правильность терма $s(t'_1, ..., t'_n)$ и эквивалентность $s(t_1, ..., t_n) \approx_E s(t'_1, ..., t'_n)$.

Доказательство. Доказательство будем проводить индукцией по глубине терма $s(x_1, ..., x_n)$. Если глубина этого терма ноль, то этот терм — переменная, и для него утверждение очевидно. Если этот терм имеет глубину один, то $s(x_1, ..., x_n) = F(x_1, ..., x_n)$, где $F \in D$, то утверждение 3.2 в этом случае совпадает с правилом 4 определения 3.2 построения отношения eq_E.

В соответствии с правилом математической индукции предположим, что утверждение 3.2 верно для всех термов глубины меньше или равной k. Предположим, теперь, что терм $s(x_1, ..., x_n)$ имеет глубину, равную $k + 1$. Тогда $s(x_1, ..., x_n)$ имеет вид $s(x_1, ..., x_n) = F(s_1(x_1, ..., x_n), ..., s_k(x_1, ..., x_n))$ для некоторого $F \in D$ и термов $s_1(x_1, ..., x_n), ..., s_k(x_1, ..., x_n)$ глубины меньше или равной k. Если $s(t_1, ..., t_n) = F(s_1(t_1, ..., t_n), ..., s_k(t_1, ..., t_n))$ — правильно построенный терм, то согласно утверждению 3.1 термы $s_1(t_1, ..., t_n), ..., s_k(t_1, ..., t_n)$ — также правильно построены. Отсюда, по предположению индукции, имеем эквивалентности
$s_i(t_1, ..., t_n) \approx_E s_i(t'_1, ..., t'_n)$, для $i = 1, ..., k$.
Тогда, правило 4 определения 3.2 дает нам эквивалентность
$F(s_1(t_1, ..., t_n), ..., s_k(t_1, ..., t_n)) \approx_E F(s_1(t'_1, ..., t'_n), ..., s_k(t'_1, ..., t'_n))$,
которая совпадает с требуемой эквивалентностью
$s(t_1, ..., t_n) \approx_E s(t'_1, ..., t'_n)$.
Отсюда, по индукции следует истинность утверждения 3.2.

Следствие 3.3. Для любых двух термов $s_1(x_1, ..., x_n)$, $s_2(x_1, ..., x_n)$ и набора эквивалентных термов $t_1 \approx_E t'_1, \dots , t_n \approx_E t'_n$ относительно эквивалентности eq_E из эквивалентности $s_1(t_1, ..., t_n) \approx_E s_2(t_1, ..., t_n)$ следует эквивалентность $s_1(t'_1, ..., t'_n) \approx_E s_2(t'_1, ..., t'_n)$.

Д о к а з а т е л ь с т в о. Так как выполняется эквивалентность $s_1(t_1, ..., t_n) \approx_E s_2(t_1, ..., t_n)$, то термы $s_1(t_1, ..., t_n)$, $s_2(t_1, ..., t_n)$ правильно построены, и к ним может быть применено утверждение 3.2. Имеем $s_1(t_1, ..., t_n) \approx_E s_1(t'_1, ..., t'_n)$ и $s_2(t_1, ..., t_n) \approx_E s_2(t'_1, ..., t'_n)$. Отсюда, по правилу 3 определения 3.2 (симметричности и транзитивности отношения эквивалентности) получаем требуемую эквивалентность $s_1(t'_1, ..., t'_n) \approx_E s_2(t'_1, ..., t'_n)$.

У т в е р ж д е н и е 3.4. Если терм $F(t_1, ..., t_n) \in CTerm(D, E, X)$ правильно построен для спецификации $S = (D, E)$, где $F \in D$ — словарный термин, и $t_1, ..., t_n$ — некоторые термы, то конкретизированные уравнения условия применения термина $F.cond.eq(t_1, ..., t_n)$ принадлежат отношению эквивалентности eq_E.

Д о к а з а т е л ь с т в о. Доказательство будем проводить индукцией по шагам построения множества правильно построенных термов $CTerm(D, E)$. На первом шаге правильно построенный терм может быть построен только по правилам 1 или 2 определения 3.2 с пустым условием применения. Для таких термов выполнение утверждения 3.4 очевидно. Предположим, теперь, что терм $F(t_1, ..., t_n)$ построен на некотором шаге и все правильно построенные термы, построенные на предыдущих шагах, удовлетворяют утверждению 3.4. В этом случае, терм $F(t_1, ..., t_n)$ либо построен по правилу 2 (и тогда для него утверждение 3.4, очевидно, верно), либо по правилу 3 из ранее построенного терма $F(t'_1, ..., t'_n)$ и эквивалентностей $t'_1 \approx_E t_1$,...,$t'_n \approx_E t_n$. В последнем случае, по предположению индукции равенства уравнений $F.cond.eq(t_1, ..., t_n)$ принадлежат отношению эквивалентности eq_E. Тогда, согласно утверждению следствия 3.3, равенства системы уравнений $F.cond.eq(t_1, ..., t_n)$ также принадлежат отношению эквивалентности eq_E. Следовательно, утверждение 3.4 выполняется для всех правильно построенных термов.

Доказанные утверждения позволяют ввести некоторые правила проверки правильности построения термов и эквивалентности термов. Например, пользуясь утверждениями 3.1–3.4, проверка правильности построения терма может состоять в последовательной проверке правильности построения подтермов по возрастанию их сложности и построении для них канонических эквивалентных им термов в данной аппроксимации. Если правильность подтермов проверена а сам терм не входит в текущую аппроксимацию, то можно проверить условия применения соответствующего термина, и при их выполнении расширить текущую аппроксимацию новым правильно построенным термом. Для проверки эквивалентности двух термов можно привести эти термы к каноническому виду в аппроксимации и сравнить результаты.

Спецификации алгебр приведенных примеров удовлетворяет еще одному важному для контроля правильности определения свойству: последовательность введения условных операций и условных соотношений в примерах такова, что правильность термов, использованных в уравнениях на каждом шаге, следует из части определения, приведенной на предыдущих шагах спецификации.

Кроме того, приведенные здесь примеры подтверждают отмеченную в главе 2 важность использования сокращений и модульности в организации спецификаций для задач представления знаний.

Литература

[1] Агафонов В. Н. *Спецификация программ: понятийные средства и их организация.* Новосибирск: Наука, Сиб.отд-ние, 1990.

[2] Агафонов В. Н. *Типы и абстракция данных в языках программирования.* В кн. Данные в языках программирования. М.:Мир,1982, с.263–327.

[3] Бениаминов Е. М. *Алгебраический подход к моделям баз данных реляционного типа.* В кн.:Семиотика и информатика, 1980, вып.14, с.44-80.

[4] Бениаминов Е. М. *Алгебраическая стуктура реляционных моделей баз данных.* НТИ, сер.2, 1980, N9, с. 23-25.

[5] Бениаминов Е. М., Березина Н. А. *Об алгебраическом подходе к описанию схем баз данных.*// В сб. Вопросы создания Автоматизированной системы НТИ по документам ГАФ СССР, Москва: ГАУ при Совете Министров СССР, ВНИИДАД, 1981, с.69-77.

[6] Бениаминов Е. М. *О роли симметрии в реляционных моделях баз данных и логических структурах.* НТИ, сер.2, 1984, N5, с.17-25.

[7] Бениаминов Е. М., Березина Н. А., Дунская М. В. *Разработка методов моделирования автоматизированной обработки, поиска и размещения данных в больших информационных системах.* СИФ ОЦНТИ, ВНИИДАД, депонированная рукопись, инв. N031–85,М.,1985.

[8] Бениаминов Е. М. *О некотором подходе к представлению знаний.* Тезисы докладов 4 Всесоюзн.конференции "Применение методов мат. логики" г. Таллин, 1986, с.34-36.

[9] Бениаминов Е. М. *Основания категорного подхода к представлению знаний. Категорные средства.* Изв. АН СССР Техн. кибернет.,N 2, 1988 , с.21–33.

[10] Бениаминов Е. М. *Рефлексивные топосы в категорном подходе к представлению знаний.* Тезисы докладов Всесоюзн. школы-семинара

"Семиотические аспекты формализации интеллектуальной деятельности" г. Боржоми, 1988, с.111-113.

[11] Бениаминов Е.М., Вайнтроб А. Ю. *Основные принципы диалогового языка для представления знаний средствами категорного подхода.* Материалы конференции ДИАЛОГ-87, г. Тбилиси, 1988, с.174-177.

[12] Бениаминов Е. М. *Алгебраические системы и типы данных.* //В кн.:Системное и теоретическое программирование, Ростов-н-Д: РГУ, 1988, с.83-92.

[13] Бениаминов Е. М. *Система представления и обработки понятий, основанная на алгебраическом (категорном) подходе.* Труды II Всесоюзной конференции "Искусственный интеллект- 90", Минск, т.2, 1990, с.8-11.

[14] Вигнер П. *Программирование на языке АДА.* М.:Мир, 1983.

[15] Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л. *Алгебра, логика, языки, программирование.* Киев:Наукова думка, 1974.

[16] Голдблатт Р. *Топосы. Категорный анализ логики.* М.:Мир, 1983.

[17] Гончаров С. С., Ершов Ю. Л., Свириденко Д. И. *Семантические основы логического программирования*// В сб. Концептуализация и смысл, под ред. Полякова И. В., Новосибирск:Наука, 1990, с.6-20.

[18] Джонстон П. Т. *Теория топосов.* М.:Наука, 1986.

[19] Жожикашвили А.В., Стефанюк В.Л. *Теория категорий в задачах представления знаний и обучения.* Изв. АН СССР. Техн. кибернет., N 2, 1986.

[20] Замулин А. В. *Системы программирования баз данных и знаний.* Новосибирск:Наука, 1990.

[21] Калиниченко Л. А. *Методы и средства интеграции неоднородных баз данных.* М.:Наука, 1983.

[22] Калиниченко Л. А., Рывкин В. М. *Машины баз данных и знаний.* М.:Наука, 1990.

[23] Капитонова Ю. В., Летичевский А. А. *Математическая теория проектирования вычислительных систем.* М.:Наука, 1988.

[24] Кондрашина Е. Ю., Литвинцева Л. В., Поспелов Д. А. *Представление знаний о времени и пространстве в интеллектуальных системах.* М.:Наука, 1989.

[25] Кучеров Г. А. *Системы подстановок термов.* Препринт 601, ВЦ АН СССР Сиб. отделение, Новосибирск, 1985.

[26] Мальцев А. И. *К общей теории алгебраических систем.* Мат. сборник, 1954, т.35, вып.1.

[27] Массер Д. *Спецификация абстрактных типов данных в системе AFFIRM.*// В сб.: Требования и спецификации в разработке программ, М.:Мир, 1984, с.199-222.

[28] Плоткин Б. И. *Универсальная алгебра, алгебраическая логика и базы данных.* М.:Наука, 1991.

[29] Поспелов Д. А. *Логико-лингвистические модели в системах управления.* М.:Энергоиздат, 1981.

[30] Свириденко Д. И. *Проект Сигма. Цели и задачи.* // В сб. Логические методы в программировании под ред. Ершова Ю. Л. (Вычислительные системы, вып. 133), РАН, Сиб. отд.-ние, Ин.-т математики, Новосибирск, 1990, с.68-94.

[31] Цаленко М. Ш. *Моделирование семантики в базах данных.* М.:Наука, 1989.

[32] Шенфилд Дж. *Математическая логика.* М.:Наука, 1975.

[33] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 1)* Comput. Lingust. Comput. Lang., 13, 1979, p. 152-282.

[34] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 2)* Comput. Lingust. Comput. Lang., 14, 1980, p. 43-65.

[35] Andréka H., Németi I. *Applications of universal algebra, model theory and categories in computer science (Part 3)* Lect. Notes in Comp. Sci., V.117, Springer-Verlag, Berlin, 1981, p. 281-290.

[36] Banchilhon F. *On the Compliteness of Query Language for Relational Data Bases.* Lect. Notes in Comp. Sci., V.64, Springer-Verlag, 1978, pp.76-98.

[37] Beniaminov E. M. *Concept Bases and Algebraic Modeling Methods.* Proceedings of the International Workshop on Advances in Databases and Information Systems (ADBIS'94), Moscow, 1994, p.133-135.

[38] Beniaminov E. M. *A Categorical Approach to Knowledge Representation.* Japan-CIS Symposium on Knowledge Based Software Engineering'94 (JCKBSE'94), Pereslavl-Zalesski, 1994, p.181-182.

[39] Beniaminov E. M. *Algebraic Invariants of Database Schemes.* Proceedings of the Second International Workshop on Advances in Databases and Information Systems (ADBIS'95), Moscow, V.1, 1995, p.259-263.

[40] Breu R. *Algebraic Specification Techniques in OOP Environments.* Lect. Notes in Comp. Sci., V.562, Springer-Verlag, 1991.

[41] Burstall R. M., Goguen J. A. *The Semantics of Clear, a Specification Language* In: Proceedings, of the 1979 Copenhagen Winter Shcool on Abstrakt Software Specification, Springer-Verlad, Berlin, 1980, 292-332.

[42] Chandra A. K., Harel D. *Computable queries for relational data bases.* J. Comput. & Syst. Sci., V.21, N2, 1980, p.156-172.

[43] Codd E. F. *A relational model for large shared data banks.* Comm. of ACM 13, 6, 1970, p.377-387.

[44] Dershowitz N. *Orderings for term-rewriting systems.* Theor. Comput. Sci., 1982, V.17, 3, pp.279-302.

[45] Elmasri R., Weeldrever J., Hevner A. *The category concept: an extension to the entity-relatconship model* Data & Knowledge Engineering, V.1, N1,1985.

[46] Ehrig H., Mahr B. *Fundamentals of Algebraic Specification.* V.1, Springer-Verlag, 1985.

[47] Ehrig H., Mahr B. *Fundamentals of Algebraic Specification.* V.2, Springer-Verlag, 1990.

[48] *First International Symposium on Category Theory Applied to Computation and Control* Lect. Notes in Com. Sci, V.25, Springer - Verlad, Berlin, 1975.

[49] Georgescu I. *A Categorial approach to knowledge-based systems.* Computers and Artifical Intelligence, V.3, N2, 1984, pp.105-113.

[50] Goguen J. A., Thatchar J. W., Wagner E. G. *An inititial algebra approach to the specification, correctness and implementation of abstract data types.* In Current Trends in Programming Methodology IV: Data Structuring, Prentice Hall, 1978, pp.80-144.

[51] Goguen J. A. *Some design principles and theory for OBJ-O, a language to express and execute algebraic specifications of problems.* In: Lect. Notes Comput. Sci.,V.75, 1979, pp.425-473.

[52] Goguen J. A., Burstall R. M. *Some fundamental algebraic tools for the semantics of computation.* Theoretical Computer Science, V.31, N2, 3, 1984.

[53] Goguen J. A., Burstall R. M. *Introducing Institution.* Lect. Not. Comp.Sci., V.164, 1984.

[54] Goguen J. A., Meseguer J. *Equality, types, modules, and (why not?) generics for logic programming.* Conf. of Logical Prog., Uppsala, Sweden, 1984, 179-210.

[55] Goguen J. A., Meseguer J. *Eqlog: Equality, types, and generic modules for logic programming.* In Douglas SeGroot and Gary Lindstrom, eds. "Logic Programming: Functions, Relations and Equations", Prentice-Hall, Englewood Cliffs, N.J., 1986, pp.295-363.

[56] Goguen J. A., Meseguer J. *Order-sorted algebra solves the constructor selector, multiple representation and coercion problems.* Symposium on Logic in Comp. Sci., IEEE Comp. Society Press, 1987, pp.18-29.

[57] Goguen J. A., Burstall R. M. *Institutions: Abstract Model Theory for Specification and Programming.* Jorn. of ACM, V.39, 1, 1992, pp.95-146.

[58] Grothendieck A., Verdier J. L. *Théorie des Topos.* (SGA 4, exposés I-VI).–Second edition.–Berlin; Heidelberg; N. Y.:Springer, 1972.

[59] Huet G. *Confluent reductions: abstract properties and applications to term rewriting systems.* Journ. of ACM, 1980, v.27, N4, pp.797-821.

[60] Kaplan S. *Symplifying conditional term rewriting systems: unification, termination and confluence.* Jorn. Symbolic Computation 4(3), 1987, pp.295-334.

[61] Knuth D., Bendix P. *Simple word problems in universal algebras.* In: Conputational Problems in Abstract Algebra, Pergamon Press, Elmsford, New York, 1970, pp.263-297.

[62] Krasner M. I. *Generalization et analogues de la theorie de Galois.* Comptes Rendus de Congress de la Victorie de I'Ass. Franc. pour I'Avancem. Sci., 1945, pp. 54-58.

[63] Lawvere F. W. *Functorial semantics of algebraic theories.* Proc. Nat. Acad. Sci., 1963, V.50, N5, pp.869-872.

[64] Lawvere F. W. *Introduction* In: Toposes, Algebraic Geometry and Logic, Lect. Notes in Math., V.174, 1972.

[65] Maltsev A. *Algebraic Systems.* Springer-Verlag, 1973.

[66] Melton A., Schmidt D., and Strecher G. *Galois connections and computer science applications.* Lect. Notes in Comp. Sci., V. 240, Springer-Verlag, 1986.

[67] *Proceedings, Category Theory and Computer Programming,* Lect. Notes in Comp. Sci., V. 240, Springer-Verlag, 1986.

[68] *Proceedings, Category Theory and Computer Science,* Lect. Notes in Comp. Sci. V. 283, Springer-Verlag, 1987.

[69] Rydeheard D.F., Burstall R.M. *Computational category theory.* Prentice Hall, 1988.

[70] Tuijn C., Gyssens M., Paredaens J. *A Categorical Approach to Object-Oriented Data Modelling.* Proceedings of Third Workshop on Foundation of Models and Languages for Data and Objects, Aigen, 1991, pp.187–196.

[71] Rusinowitch M., Rémy J. L. (Eds.) *Conditional Term Rewriting Systems.* Lect. Notes in Comp. Sci., V. 656, Springer-Verlag, 1993.

[72] Zilles S.N. *Introduction to data algebras.* Lect. Notes Comput. Sci., V.86, Springer-Verlag, 1980.