

ОТОБРАЖЕНИЕ ДЕСКРИПТИВНОЙ ЛОГИКИ ALC В БИНАРНУЮ РЕЛЯЦИОННУЮ СТРУКТУРУ ДАННЫХ

В статье рассматривается одна из ключевых проблем интеграции данных в семантическом вебе – создание способов манипулирования данными. На основе подхода GAV-представления в качестве центральной схемы интеграции выбрана дескриптивная логика, которая рассматривается как модель данных. В работе предложен механизм взаимодействия между дескриптивной логикой ALC и реляционной моделью данных (RDM). Он представлен в виде отображения основных компонентов синтаксиса логики ALC в RDM с помощью операций реляционной алгебры.

Ключевые слова: дескриптивная логика ALC, реляционная модель данных, интеграция моделей данных, бинарная реляционная структура данных.

Введение

Ранее в публикации [1] был приведен подробный обзор комплексной проблемы интеграции данных в семантическом вебе. В процессе анализа было выделено три составляющих: выработка схем интеграции данных, выработка отображений между моделями, выработка способов манипулирования. В данной публикации речь пойдет о второй составляющей комплексной проблемы, а именно об отображениях между различными моделями данных.

Создание универсальных механизмов взаимодействия между различными знание-ориентированными системами является актуальной научной задачей и занимает свою значимую роль в решении общей проблемы интеграции данных. Существование таких механизмов позволяет преодолеть следующие трудности:

- устранение признаков, порождающих гетерогенности данных [1];
- взаимодействие систем, в основе которых лежат устаревшие модели данных, с современными онтолого-ориентированными системами;
- устранение дублирования данных;
- возможность повторного использования различных компонентов системы;
- возможность построения комплексных информационных систем из разнородных составляющих.

Ввиду ряда особенностей, приведенных в обзоре [1], в основу нашего подхода легли следующие компоненты:

- центральная схема интеграции данных;
- GAV/LAV представление;
- формально-логический уровень абстракции.

Суть *центральной схемы интеграции данных* была рассмотрена в [1], а подробнее о GAV/LAV представлениях можно прочесть в [2]. В данной публикации рассматривается только *GAV представление*, поскольку созданные нами механизмы отображений разрешают взаимодействие от глобальной схемы к источнику. Создание отображений LAV представлений является предметом дальнейших исследований. Под *формально-логическим уровнем абстракции* понимается решение проблемы на уровне моделей, а не на уровне прикладных реализаций. В отличие от основных существующих методологий создания взаимодействий между системами, мы рассматриваем создание механизмов отображений не на уровне прикладных реализаций (онтологий, баз данных и т. д.), а на уровне моделей, лежащих в их основе. В связи с этим, в качестве центральной схемы рассматривается бинарная реляционная структура данных, в основе которой лежит дескриптивная логика (DL) ALC, а в качестве источника выступает RDM. Обоснование такого выбора приведено в разделе 1. В разделе 2 приведен краткий обзор ис-

следований по созданию отображений между онтологиями и базами данных, анализируется специфика этих исследований и в общих чертах раскрывается суть наших предложений. Раздел 3 посвящен структурной части DL ALC. В нем рассматривается бинарная реляционная структура данных, которую мы представляем в данной работе. В разделе 4 приведены основные понятия локального источника – реляционной модели данных (RDM). Раздел 5 полностью посвящен механизмам отображения DL ALC в бинарную реляционную структуру.

1. Дескриптивная логика как модель данных

Поскольку DL призвана решать совершенно иные задачи, нежели модель данных, то отдельного обоснования требует следующее утверждение: DL можно рассматривать как модель данных в классическом определении теории моделей данных.

Модель данных – интегрированный набор:

- понятий для описания и обработки данных, представляющих собой структуру модели;
- операций, предоставляющими возможность манипулирования данными;
- ограничений целостности, накладываемых на данные.

Опираясь на данное определение, мы утверждаем, что любую DL можно рассматривать как модель данных, в соответствии со следующими положениями:

1) DL обладают синтаксисом и семантикой. Именно в семантике проявляется её структурная часть (возникают понятия множеств, бинарных отношений и т. д.);

2) DL обладает манипулятивной частью, которую представляют такие её компоненты, как конструкторы концептов и ролей;

3) DL обладает целостной частью, которую представляют такие её компоненты, как аксиомы.

В своем обзоре [1] для решения комплексной проблемы интеграции данных мы предлагали в качестве центральной схе-

мы использовать онтологию как модель данных с последующей аргументацией такого выбора. А поскольку все современные языки описания онтологий основаны на DL, это позволяет нам утверждать, что в основе самих онтологий, рассматриваемых как модели данных, лежат DL.

Основываясь на описанных выше положениях, мы можем не только представить DL в виде модели данных, но и выбрать ее в качестве центральной модели в интеграционной схеме. В качестве источников можно рассматривать различные локальные модели. В текущей публикации в этой роли рассматривается реляционная модель.

2. Обзор исследований по отображению между онтологиями и базами данных

Проблема установления соответствий между онтологиями и реляционными базами данных, помимо чисто научного, имеет большой практический интерес. Он заключается в следующих фактах.

Онтологии должны иметь физическое место расположения. Они могут храниться в виде обычных файлов – это наиболее простой вариант, который обладает множеством недостатков (не обеспечивается масштабируемость, отсутствуют механизмы поддержки совместного использования, отсутствуют языки запросов и многие другие). Также есть иной подход – использование для этих целей СУБД. В данном контексте его основной недостаток – это поддержка гипотезы замкнутого мира, в то время как онтологии используют гипотезу открытого мира. Однако, в этом случае можно воспользоваться всем тем богатством в области хранения и манипулирования данными, которое было разработано в последние 50 лет в базах данных.

Согласно [3] около 77,3 % данных существующего веба содержится в реляционных базах данных, из которых формируется огромное количество веб-страниц. Данный факт делает заманчивым использовать, накопленные за эти годы, массивы данных для создания онтологий в семантическом вебе.

Последние 10 лет мы наблюдаем рождение и стремительный рост совершенно новой IT-отрасли – мобильные технологии. Это совершенно особый вид аппаратного и программного обеспечения, который активно внедряется во все отрасли человеческой деятельности – от бытовой до космической. Появление в руках принципиально новых устройств сильно изменило мировое общественное сознание. Это явление позволило рядовому обывателю интуитивно почувствовать все преимущества семантических данных, что лишь подстегнуло передовое научное сообщество к дальнейшему интенсивному развитию и внедрению семантических технологий в жизнь. Однако, реальность такова, что на сегодняшний день 100 % мобильных устройств используют для своей работы реляционные базы данных (или их адаптированную версию). Это происходит из их тесного взаимодействия с существующим вебом и сложными программными комплексами, не использующими семантические технологии для своей работы. Поэтому преодоление трудностей взаимодействия реляционной базы данных (RDB) и онтологий становятся требованием XXI века.

На основании приведенных фактов мы сделали вывод, что в контексте проблемы отображений первоочередной задачей является создание механизмов взаимодействия именно между RDM и онтологией. Мы обратились к источникам, чтобы проанализировать, какой объем работ уже накоплен для решения поставленной задачи.

Сделаем краткий обзор исследований по установлению отображений между онтологической и реляционной моделями данных. Прежде всего следует отметить, что независимо от вариантов установления отображения следует отслеживать и управлять следующими ситуациями.

Возможная потеря данных. Преобразование должно быть таким, что результирующая модель содержала в себе все данные исходной модели.

Возможная потеря семантики. Возможна ситуация, когда данные не теряются, но их семантика при преобразова-

нии «теряется». К таким семантическим характеристикам, например, в реляционной модели относятся ограничения целостности. При преобразовании следует либо гарантировать сохранение семантики, либо указывать, в каких случаях какая именно семантика теряется.

Полнота преобразования. Полнота предполагает, что все, что представимо в исходной модели, может быть преобразовано в результирующую модель.

Взаимная однозначность. Она предполагает, что результат преобразования при обратном отображении дает исходный объект исходной модели.

Существует следующая классификация направлений исследований и разработок в области описания отображений между онтологической и реляционной моделью данных [4, 5]:

- 1) использование промежуточной модели данных;
- 2) непосредственное отображение онтологии в реляционную базу данных;
- 3) непосредственное отображение реляционной базы данных в онтологию;
- 4) создание и использование онтолого-ориентированной базы данных.

Рассмотрим эти подходы.

2.1. Использование промежуточной модели данных.

Данный подход подробно изложен в фундаментальной работе [2]. Его идея заключается в том, что преобразование производится не непосредственно, а с использованием промежуточной модели, предполагая, что описание отображений между такой моделью-посредником, с одной стороны, и онтологией и реляционной моделью, с другой стороны, производится намного легче и естественнее. Так, например, в работе [6] в качестве такого посредника предлагается использовать UML. В других исследованиях в качестве модели-посредника рассматривается ER. В связи с тем, что отображение ER-RDM уже давно исчерпывающе исследовано, то до последнего времени проблема оставалась за отображением Онтология-ER. Учитывая это в работе [7] был предложен подход по автоматическому отображению ER в OWL и наоборот, с использованием DL. В рабо-

те [8] также был предложен метод автоматического отображения расширенной ER-модели в онтологию OWL.

2.2. Непосредственное отображение онтологии в базу данных.

Суть этого подхода заключается в том, что предлагается алгоритм преобразования онтологии в базу данных. Можно рассматривать различные модели баз данных для такого преобразования – реляционную, объектную, объектно-реляционную. Отображение онтологии в реляционную базу данных является не столь непосредственным, как в объектную или объектно-реляционную, так как реляционная модель данных не поддерживает отношение наследования (таксономию). Однако, реляционные СУБД обладают неоспоримыми преимуществами перед двумя упомянутыми выше, в связи с чем практически все исследования по отображению онтология – база данных базируются именно на реляционной модели. Например, в работе [4] предлагается алгоритм преобразования онтологий OWL в схему RDB, который преобразует классы, свойства, атрибуты и ограничения OWL в конструкции схемы RDB. Алгоритм преобразует OWL Lite и частично OWL DL. В работе [9] также предлагается набор правил отображения OWL в SQL-базу данных. Предложенные 12 правил не только отображают такие структурные элементы, как классы и свойства, но также учитывают характеристики свойств (однозначные-многозначные, обязательные-факультативные, функциональные, инверсно-функциональные) и другие характеристики (типы данных, перечислимые классы и др.). В работе [10] предлагается отображать онтологию не в саму РБД, а в концептуальную модель данных базы данных.

2.3. Непосредственное отображение базы данных в онтологию.

Это наиболее популярное направление среди всех, имеющих отношение к установлению взаимосвязи между онтологиями и RDB. Выделяются следующие аспекты исследований и разработок в этом направлении:

- отображение схемы (метаданных) RDB в онтологию;
- отображение с учетом анализа данных базы данных;
- отображение с учетом анализа запросов к базе данных.

Рассмотрим эти направления.

Отображение схемы (метаданных) RDB в онтологию. В работе [11] предлагается отображать таблицы в OWL-классы и затем обогащать их с использованием других информационных источников. В работе [12] в рамках решения проблемы переноса в среду семантического веба веб-приложений, интенсивно использующих данные, предлагаются правила анализа и отображения конструкций реляционной модели (отношений, атрибутов, кортежей и ограничений целостности) в соответствующие семантически эквивалентные конструкции онтологии (классы, атрибуты, экземпляры, аксиомы). Различные аспекты установления такого отображения, а также описание соответствующих трансформирующих правил также описываются в следующих работах [13–20].

Отображение с учетом анализа данных базы данных. Анализ данных базы данных может предоставить дополнительную информацию, которая является полезной для решения проблемы отображения RDM-Онтология. Например, если установлено, что домен атрибута одного отношения является поддоменом атрибута другого отношения, то это может служить основанием для заключения, что между этими отношениями, возможно, существует таксономическая иерархия. Так, например, в [16] алгоритм отображения содержит не только правила, использующие схему базы данных, но и правила, которые учитывают возможные ограничения на данные базы данных (которые в схеме не могут задаваться). Интересный подход предлагается в [11], когда онтологии генерируются из «традиционных» таблиц (например, таблиц, которые получаются при публикации базы данных или ее фрагментов в HTML-страницах) Этот подход

получил название TANGO (Table ANalysis for Generating Ontologies) и он предполагает:

- распознавание структуры таблицы и извлечение из нее концептов;
- выявление и анализ ограничений, существующих между концептами;
- сопоставление концептов с другими, более общими концептами;
- слияние результирующей структуры с другими аналогичными структурами представления знаний.

Как заявляют авторы, TANGO – это формальный метод обработки структуры и содержимого таблиц, который может использоваться для последовательного нарастающего построения релевантной повторно используемой концептуальной онтологии.

Отметим еще одно исследование в этой области [12]. Оно основывается на идее использования семантики, полученной на основании анализа HTML-страниц с содержимым базы данных, для реструктурирования и обогащения схемы базы данных, с целью ее последующего использования при построении онтологии OWL. Вопросу использования HTML-контента, для более глубокого анализа содержимого базы данных, также посвящена статья [17].

Отображение с учетом анализа запросов к базе данных. Не только содержимое базы данных, но и запросы к ней, могут служить источником дополнительной информации. Так в статье [21] предлагает следующий метод. Сначала на основании анализа схемы и базы данных строится онтология. Затем осуществляется анализ запросов пользователей, извлекается из них дополнительная информация и на ее основании совершенствуется онтология. Как следует из примеров, в результате анализа запросов в онтологии могут появляться новые классы, свойства, таксономические отношения между классами и т. д.

2.4. Создание и использование онтолого-ориентированной базы данных.

Проводятся исследования и разработки по еще одному направлению в области интеграции онтологий и баз данных – созданию специализированных реляцион-

ных баз данных для хранения и манипулирования онтологиями. Такие базы данных получили название онтолого-ориентированных баз данных (OBDB – Ontology Based DataBases). Согласно [22] OBDB – это такая база данных, в которой онтология и ее данные хранятся в единой общей модели. Согласно [23], в идеале в OBDB должны явно представляться онтология, схема данных, сами данные их связи со схемой и онтологией, но на практике, как правило, это не поддерживается в полной мере.

Приведем ряд примеров. В работе [22] предлагается разделять определения классов онтологии и их экземпляров для их хранения в базе данных. В работе [24] описывается инструментальное средство для запоминания онтологий OWL в реляционной базе данных с использованием конфигурационных XML-файлов. В работе [25] предлагается вариант OBDB, в которой расширяется классическая архитектура баз данных ANSI/SPARC введением семантических данных.

Подводя итоги проведенного анализа, можно отметить следующую специфику этих исследований, раскрывая при этом суть нашего подхода:

1) отсутствие формального подхода. В качестве информационной модели семантического веба выбирается онтология и, как правило, в ее лице берется OWL. Учитывая то, что язык OWL сам по себе не является формальным языком, описание механизма отображений дается на содержательном уровне в виде набора описательных правил. В нашем случае для описания отображений мы берем DL, что дает возможность поставить проблему отображений на более формальный уровень;

2) односторонний механизм описания отображений. Как правило, независимо описываются отображения из онтологии в RDM и наоборот, проблема установления взаимно-однозначного соответствий при этом не решается. Мы стремимся к тому, чтобы в нашем подходе эта проблема была решена;

3) отображаются только структурные составляющие моделей. Существующие механизмы описывают отображение

только структурных компонент модели данных. Абсолютно ничего не говорится об отображении конструкторов концептов и ролей, с одной стороны, и операций реляционной алгебры – с другой. Наш подход заключается в том, чтобы также отразить и эту составную часть отображений;

4) конструкции, которые не отображаются. Совершенно не понятно, каким образом отображать в классическую реляционную модель аксиомы DL (равенства, включения и непересекаемости концептов). Мы предлагаем реляционную структуру, которая предоставляет такие возможности;

5) гипотезы открытого мира и неунитарности имен. DL строится в предположении существования гипотез открытого мира и неунитарности имен. В свою очередь, реляционная модель формулируется из предположения существования гипотез замкнутого мира и уникальности имен. Предлагая свой вариант реляционной структуры, мы отчасти решаем эти противоречия.

3. Структура дескриптивной логики ALC

Основополагающей частью механизма отображений является центральная схема. Как уже было сказано, в этой роли будет выступать DL, которая должна быть представлена в виде модели данных. Среди семейства DL мы остановились на DL ALC. Такой выбор обусловлен следующим.

Современные DL, которые используются для создания языков описания онтологий, представляют собой набор операций, которые называются расширениями, ввиду того, что они расширяют базовый синтаксис DL. Это не означает, что они ограничиваются лишь двумя-тремя такими операциями. Это означает, что задание сочетания определенных расширений предполагает наличие полного базового синтаксиса. Большинство современных DL (например, SHIQ, SHOIN, DLR и т. д.) расширяют именно логику ALC. В результате она была выбрана нами на начальном

этапе для создания отображений между классической RDM и различными DL.

Теперь для создания механизмов отображений в качестве центральной схемы мы будем рассматривать модель данных DL ALC. Для этого представим, созданную нами, реляционную структуру, которая является специальным вариантом классической реляционной структуры, в терминах которой мы будем описывать модель DL ALC.

Для описания отображения между структурой ALC и реляционной структурой, в качестве посредника мы выбрали ER-модель, которая описывается при помощи ER-языка. На наш взгляд он является наиболее удобным и наглядным в данной ситуации. Каждый элемент ER-модели можно представить в терминах реляционной структуры известным способом, что является основным преимуществом такого выбора.

Мы используем диалект языка ER, предложенный Баркером [26].

Сама структура показана на рис. 1. Дадим некоторые пояснения.

В структуре присутствуют только такие сущности, которые представляют унарные и бинарные отношения, так как в DL ALC n-арные отношения отсутствуют, а также их экземпляры. Следует отметить, что DL с n-арными отношениями существуют, однако они представляют собой специальные расширения ALC, поэтому на данном этапе мы решили отказаться от них, сосредоточившись только на самой ALC.

В отличие от классической RDM, в нашей структуре представлена одновременно и модельная, и метамодельная часть. К метамодельной части мы относим сущности Concept и Role, которые представляют собой перечни имен концептов и ролей соответственно. С помощью сущностей-связей, которые обозначены как «=», « \sqsubseteq » и « \emptyset » для сущностей концептов и ролей, представляются аксиомы равенства, включения и непересекаемости концептов и ролей логики DL ALC. Следует отметить, что в DL не может существовать отдельно роль, как самостоятельная единица.

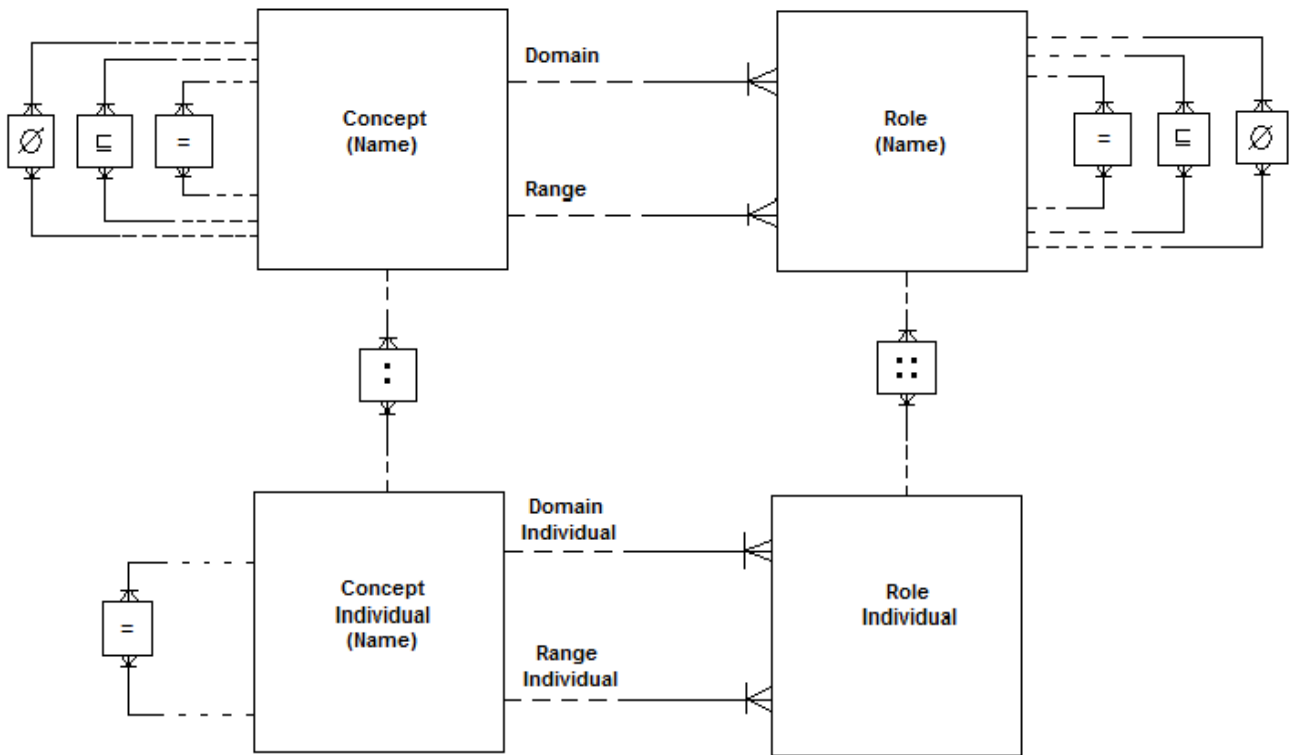


Рис. 1. Бинарная реляционная структура данных

В нашей структуре роль может существовать отдельно, будучи представлена именем самой роли, именем концепта-домена и именем концепта-диапазона.

К модельной части относятся сущности *Concept Individual* и *Role Individual*, которые представляют собой индивиды концептов и индивиды ролей. В явном виде таких элементов, как индивиды ролей нет в DL, однако мы вводим данную сущность для связывания индивидов концептов с именами ролей. Обратим внимание, что в нашей структуре есть сущность-связка, обозначенная как « \Leftarrow » для индивидов концептов, которая предназначена для представления аксиом DL равенства индивидов. Она показывает, что у одного индивида может быть множество имен, то есть не выполняется принцип уникальности. Однако по-прежнему одно имя именуется только один индивид.

Следует отметить, что для индивидов ролей мы не вводим такую аксиому, потому что на данный момент этот вопрос недостаточно исследован, и у нас нет никаких оснований утверждать, что одна и та же связь между одними и теми же индивидами может иметь несколько имен. Поэтому,

в нашей структуре верно лишь утверждение о том, что экземпляр роли существует и представляет собой пару индивидов концептов, связанную определенной ролью.

Нашу структуру мы представили в терминах RDM, обозначив её как RM^2 , где индекс 2 обозначает, что данная структура позволяет представлять не более, чем бинарные отношения. Она показана на рис. 2, к которому следует дать дополнительные объяснения.

Поскольку один концепт может иметь множество индивидов, а один и тот же индивид может принадлежать нескольким концептам, то для разрешения связи многие-ко-многим мы вводим дополнительную сущность *LinkConceptIndividual*. Аналогично, сущность *LinkRoleIndividual* разрешает связь многие-ко-многим для роли и индивидов, которые она связывает. Каждая роль имеет много экземпляров, связывающих два индивида. С другой стороны, каждая конкретная пара индивидов может быть связана несколькими ролями.

На модели оставлены аксиомы DL ALC для наглядности. В данной работе мы не рассматриваем подробно эти связи, а

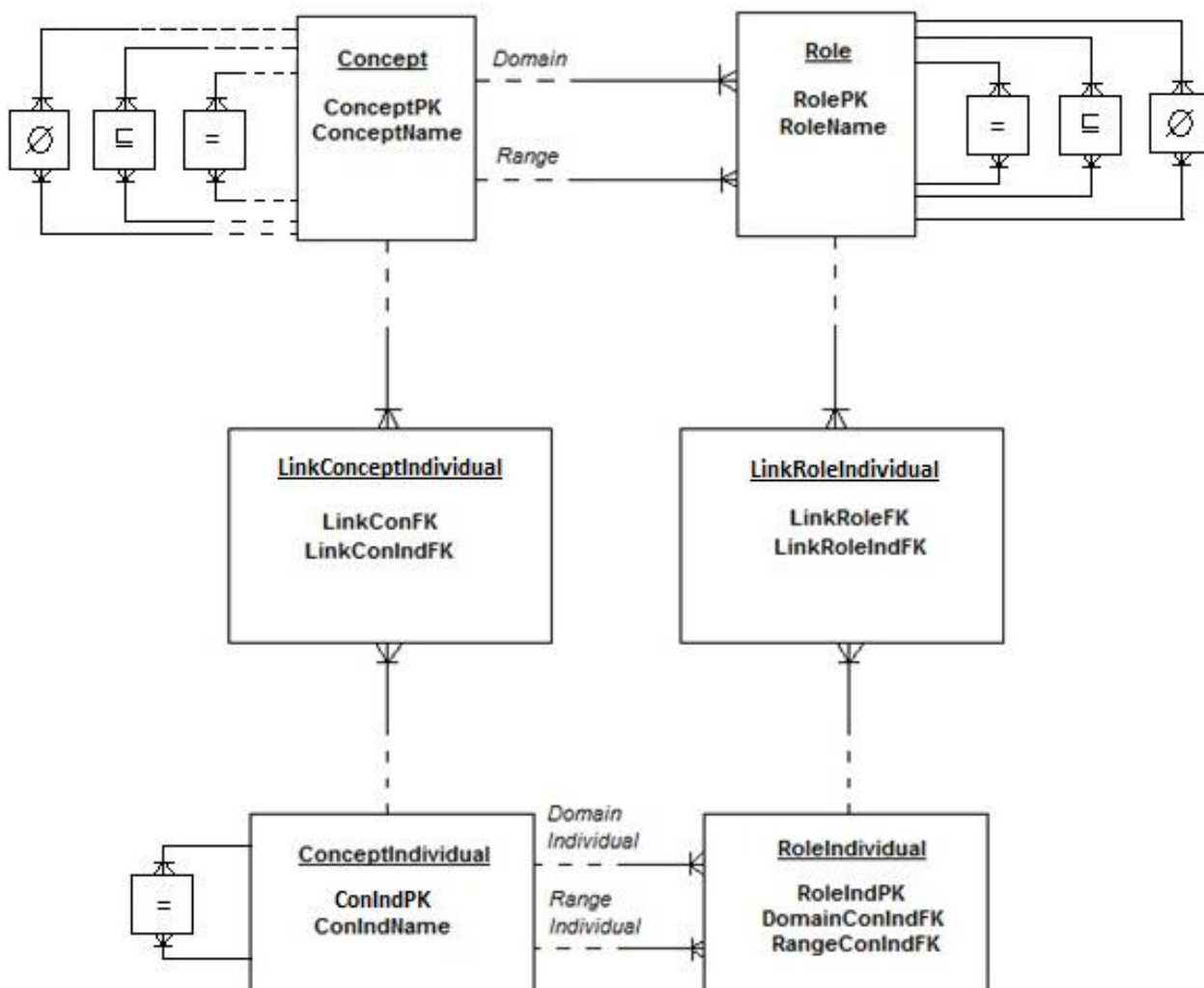


Рис. 2. Структура данных RM^2

лишь показываем таким образом, что они существуют и представимы в RM^2 . Кроме того, мы не представляем связи *Domain* и *Range* между концептами и ролями, так как они нам не потребуются.

Следует отметить, что RM представляет собой разновидность онтолого-ориентированной базы данных согласно классификации, приведенной в разделе 2.

4. Основные определения RDM

Прежде чем переходить к описанию отображений, приведем ряд определений и обозначений, имеющих отношение к реляционной модели данных, которые будут использоваться в статье.

Реляционная модель данных (RDM) – это совокупность реляционных структуры, алгебры и ограничений целостности. Что касается реляционных

ограничений целостности, то они не используются в статье, поэтому мы ничего о них не говорим.

Реляционная структура данных – это совокупность реляционных отношений. Реляционное отношение (так как в рамках данной статьи мы говорим только о реляционных отношениях, то для краткости будем использовать термин «отношение») – это пара (R^S, R^E) , где R^S – схема (или интенционал) отношения, а R^E – экземпляр (или экстенционал) отношения.

Схема отношения R^S – это выражение вида $R(A_{i1}:D_{j1}, \dots, A_{in}:D_{jk})$, где:

- R – имя отношения;
- A_i – имена атрибутов отношения;
- D_j – имена доменов отношения, определяющих области допустимых значений атрибутов.

Если указание имен доменов не существенно, то схема записывается так $R(A_1, \dots, A_n)$. Иногда при записи схемы, например, $R(A, B, C)$, под A, B, C будут подразумеваться наборы упорядоченных атрибутов, а не отдельные атрибуты.

Имена отношений являются уникальными в реляционной структуре. Имена атрибутов являются уникальными в схеме отношения, но могут повторяться в различных схемах. Имена доменов являются уникальными в реляционной структуре. На одном домене может определяться множество атрибутов.

Экземпляр отношения – это подмножество декартового произведения доменов, на которых определены атрибуты: $R^E \subseteq D_{j1} \times \dots \times D_{jk}$. Множество значений i -го столбца экземпляра отношения представляет собой множество значений i -го атрибута в схеме отношения.

Реляционная алгебра – это совокупность операций, определенных на отношениях. Она является замкнутой в том смысле, что результат выполнения любой операции также является отношением. Кратко рассмотрим их.

Именованное/переименование (ρ). Как правило, схема результирующего выражения реляционной алгебры не может быть выведена из схем исходных отношений. Оператор именованного позволяет такую схему задать. Было предложено множество различных вариантов этого оператора. Мы будем использовать следующие три:

- $\rho_R(E)$ – отношению, полученному в результате вычисления выражения реляционной алгебры E , присваивается имя R ;
- $\rho_{R(A_1, \dots, A_n)}(E)$ – n -арному отношению, полученному в результате вычисления выражения реляционной алгебры E , присваивается схема $R(A_1, \dots, A_n)$;
- $\rho_{A/B}(E)$ – в отношении, полученном в результате вычисления выражения реляционной алгебры E , атрибут A переименовывается в атрибут B .

Определение 1. Два отношения со схемами $R_1(A_1, \dots, A_n)$ и $R_2(B_1, \dots, B_k)$ являются совместимыми (по объединению),

если $n=k$ и пары атрибутов (A_i, B_i) определены на одном и том же домене. Над совместимыми отношениями определяются следующие три множественные операции.

Объединение (\cup). Объединением двух совместимых отношений является отношение, экземпляр которого является объединением экземпляров исходных отношений:

$$R_1 \cup R_2 = \{r \mid r \in R_1 \vee r \in R_2\}.$$

Пересечение (\cap). Пересечением двух совместимых отношений является отношение, экземпляр которого является пересечением экземпляров исходных отношений:

$$R_1 \cap R_2 = \{r \mid r \in R_1 \wedge r \in R_2\}.$$

Разность ($-$). Разностью двух совместимых отношений является отношение, экземпляр которого содержит все кортежи первого отношения, которых нет среди кортежей второго отношения:

$$R_1 - R_2 = \{r \mid r \in R_1 \wedge r \in S_2\}.$$

Схемы результирующих отношений всех трех операций определяются операцией именованного.

Проекция (π). Пусть задано отношение со схемой $R(A, B)$, где A и B – атрибуты или наборы атрибутов. Проекцией R на A , обозначаемой $\pi_A(R)$, называется отношение, экземпляр которого содержит только A -компоненты кортежей отношения R :

$$\pi_A(R) = \{r[A] \mid r \in R\}.$$

Здесь запись $r[A]$ обозначает выделение из кортежа r компонент, принадлежащих значениям атрибутов из набора A . Схема результирующего отношения наследует имена атрибутов A , а имя отношения определяется операцией именованного.

Определение 2. Пусть θ – любой из следующих предикатов сравнения: $=, \neq, <, \leq, >, \geq$. Атрибуты A и B одного и того же или различных отношений называются θ -сравнимыми, если для любых пар значений $a \in A$ и $b \in B$ определено выражение $a \theta b$. Наборы атрибутов $M = (A_1, \dots, A_k)$ и $N =$

$= (B_1, \dots, B_n)$ называются θ -сравнимыми, если $k = n$ и пары атрибутов (A_i, B_i) являются θ -сравнимыми. В этом случае $M \theta N$ подразумевает следующее: $M \theta N = (A_1 \theta B_1) \wedge \dots \wedge (A_k \theta B_k)$.

Селекция (σ). Пусть M и N – θ -сравнимые атрибуты или наборы атрибутов отношения R . Селекцией отношения R по условию $M \theta N$, обозначаемой $\sigma_{M\theta N}(R)$, называется отношение, экземпляр которого содержит те кортежи R , на которых истинно условие $M \theta N$:

$$\sigma_{M\theta N}(R) = \{r \mid r \in R \wedge r[M] \theta r[N]\}.$$

Схема результирующего отношения наследует имена атрибутов отношения R , а имя отношения определяется операцией именованя.

“Декартово” произведение (\times). “Декартовым” произведением отношений R_1 и R_2 , обозначаемым $R_1 \times R_2$, называется отношение, экземпляр которого содержит все возможные соединения (конкатенации) кортежей отношений R_1 и R_2 :

$$R_1 \times R_2 = \{(r_1, r_2) \mid r_1 \in R_1 \wedge r_2 \in R_2\}.$$

Схема результирующего отношения наследует имена атрибутов отношений R_1 и R_2 , а имя отношения определяется операцией именованя.

Соединение (\bowtie). Пусть M и N – θ -сравнимые атрибуты или наборы атрибутов отношений R_1 и R_2 соответственно. Соединением отношения R_1 с отношением R_2 по условию $M \theta N$, обозначаемым $R_1 \bowtie_{M\theta N} R_2$, называется отношение, экземпляр которого содержит все те пары кортежей отношений R_1 и R_2 , на которых выполняется условие $M \theta N$:

$$R_1 \bowtie_{M\theta N} R_2 = \{(r_1, r_2) \mid r_1 \in R_1 \wedge r_2 \in R_2 \wedge r_1[M] \theta r_2[N]\}.$$

Соединение следующим образом выражается через произведение и селекцию:

$$R_1 \bowtie_{M\theta N} R_2 = \sigma_{M\theta N}(R_1 \times R_2).$$

Схема результирующего отношения наследует имена атрибутов отношений R_1

и R_2 , а имя отношения определяется операцией именованя.

Определение 3. Пусть задано отношение $R(M, N)$, где M и N – отдельные атрибуты или наборы атрибутов. Образом кортежа $r_m \in \pi_M(R)$, записываемым как $Ir_m(R)$, называется такое множество кортежей $r_n \in \pi_N(R)$, соединение которых с r_m принадлежит R :

$$Ir_m(R) = \{r_n \mid r_n \in \pi_N(R) \wedge (r_m, r_n) \in R\}.$$

Деление (\div). Пусть заданы отношения $R_1(M, N)$ и $R_2(K, L)$, где M, N, K, L – отдельные атрибуты или наборы атрибутов, причем N и K – сравнимы по предикату равенства. Делением R_1 на R_2 по N и K , обозначаемым $R_1[N \div K]R_2$, называется отношение, которое состоит из таких кортежей $r_m \in \pi_M(R_1)$, образы которых $Ir_m(R_1)$ содержат все кортежи проекции $\pi_K(R_2)$:

$$R_1[N \div K]R_2 = \{r_m \mid r_m \in \pi_M(R_1) \wedge Ir_m(R_1) \supseteq \pi_K(R_2)\}.$$

Схема результирующего отношения наследует имена атрибутов набора M из R_1 , а имя отношения определяется операцией именованя.

Деление следующим образом выражается через произведение и разность:

$$R_1[N \div K]R_2 = \pi_M(R_1) - \pi_M((\pi_M(R_1) \times \pi_K(R_2)) - R_1).$$

5. Отображения концептов логики ALC в RDM

При создании отображений следует начать с описания некоторых базовых составляющих синтаксиса логики ALC. Для этого приведем его полностью.

$$\top \mid \perp \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$$

Согласно определению [27]:

– синтаксис DL описывает, какие выражения (концепты, роли, аксиомы и т. п.) считаются правильно построенными в данной логике;

– семантика указывает, как интерпретировать синтаксические выра-

жения, т. е. придать им формальный смысл.

На основании описания реляционной структуры RM^2 , приведенного в разделе 3:

- синтаксические концепты ALC будут считаться правильно построенными в RM^2 ;
- классическая семантика полностью сохраняется в ALC;
- **интерпретация синтаксиса ALC в терминах структуры RM^2 и будет являться искомым набором отображений**, описывающим интерпретацию синтаксиса DL ALC в RDM.

Введем набор обозначений, необходимых для описания отображений.

В таблице приведены все базовые составляющие синтаксиса логики ALC. Она показывает, как обозначаются синтаксические конструкции в семантике логики и в интерпретации нашей структуры. Индекс DL добавлен для отличия классического семантического обозначения с обозначениями отображений структуры RM^2 . Индекс RM^2 указывает на интерпретацию в реляционной структуре. Для введения отображений из DL в RDM мы будем пользоваться верхним индексом E (экстенционал) в обозначении синтаксических конструкций, интерпретируемых в RM^2 .

Как уже упоминалось, в DL не существует роль как самостоятельная единица, поэтому для нее не приводится семантическое обозначение в строке ALC. Однако, она существует в описываемой нами реляционной структуре, потому для ее обозначения вводится специальный символ.

Синтаксис ALC представляет собой набор концептов, сформированных из произвольных концептов с помощью специальных операций, которые называются конструкторами концептов и ролей. К

ним относятся дополнение, объединение, пересечение, существование и общность. В интерпретации данных выражений неизменно принимают участие обычные концепты и роли. Поэтому, следует четко разделять отображения базовых составляющих синтаксиса и концептов, сформированных с помощью конструкторов. Разберемся, что же представляет из себя каждый из этих компонент.

5.1. Отображения базовых составляющих синтаксиса ALC.

Согласно [27], все выражения в семантике ALC интерпретируются индукцией по построению концепта, с помощью интерпретирующей функции. Следуя этому правилу:

- каждому атомарному концепту A соответствует произвольное подмножество $A^I \subseteq \Delta$;
- каждой атомарной роли R соответствует произвольное подмножество $R^I \subseteq \Delta \times \Delta$;
- концепт *истина* (*Thing*) представляет собой всю область определения

$$T^I = \Delta;$$

- концепт *ложь* (*Nothing*) представляет собой пустое множество $\perp^I = \emptyset$.

В нашей структуре данные выражения будут интерпретироваться следующим образом:

- каждый концепт (атомарный и обычный) будут представлять собой экземпляр сущности Concept;
- каждая роль будет представлять собой экземпляр сущности Role;
- концепт *истина* (*Thing*) представляет собой сущность ConceptIndividual;
- концепт *ложь* (*Nothing*) останется пустым множеством.

Таблица. Набор обозначений для описания отображений

	C	D	R	$\neg C$	$C \cap D$	$C \cup D$	$\exists R.C$	$\forall R.C$
ALC	C_{DL}^I	D_{DL}^I	–	$(\neg C)_{DL}^I$	$(C \cap D)_{DL}^I$	$(C \cup D)_{DL}^I$	$(\exists R.C)_{DL}^I$	$(\forall R.C)_{DL}^I$
RM^2	$C_{RM^2}^E$	$D_{RM^2}^E$	$R_{RM^2}^E$	$(\neg C)_{RM^2}^E$	$(C \cap D)_{RM^2}^E$	$(C \cup D)_{RM^2}^E$	$(\exists R.C)_{RM^2}^E$	$(\forall R.C)_{RM^2}^E$

К базовыми составляющим синтаксиса ALC мы относим концепты C, D, а также роль R. С классической семантикой этих выражений все предельно ясно: каждый концепт представляет собой некоторое подмножество области определения, а роль – подмножество декартового произведения области определения. Следует разобраться, что представляют собой эти выражения в структуре RM^2 , для чего будем пользоваться рис. 2 и операциями реляционной алгебры.

Отображение концептов C и D.

Описание этого концепта определяется через его отображение. Создаётся оно следующим образом:

1) производим соединение таблиц ConceptIndividual, LinkConceptIndividual и Concept. В результате создается таблица, содержащая пары концептов и индивидов, принадлежащих концептам;

2) выполняем селекцию по “С”, то есть выбираем такие пары, которые принадлежат концепту с именем C;

3) наконец, делаем проекцию по столбцу ConIndName.

Тем самым, в результате данных шагов мы получаем таблицу, содержащую все индивиды, принадлежащие концепту с именем C. Данные этой таблицы – это данные из таблицы ConceptIndividual, которая в нашем случае является областью определения.

Далее данные шаги представлены в виде выражений реляционной алгебры.

$$C_{RM^2}^E = \pi_{ConIndName} (\sigma_{ConceptName='C'}$$

$$(ConceptIndividual \bowtie_{ConIndPK=LinkConIndFK} (LinkConceptIndividual \bowtie_{LinkConFK=ConceptPK} Concept))).$$

Описание концепта D происходит аналогично концепту C, но на шаге 2 селекция производится не по «C», а по «D».

Отображение роли R. Отображение роли R несколько сложнее, чем концептов, поэтому требует дополнительных объяснений.

1. Соединяем таблицы RoleIndividual, LinkRoleIndividual и Role. В резуль-

тате создается таблица, которая содержит все имена ролей и соответствующие им пары внешних ключей, ссылающихся на соответствующие им индивиды (домен и диапазон).

2. Делаем селекцию по “R”, то есть, выбираем такие пары внешних ключей на индивиды, которые соединены ролью R.

3. Теперь, чтобы получить собственно пары индивидов, а не их внешние ключи, производим два последовательных соединения с отношением ConceptIndividual, причем после каждого соединения переименовываем атрибут ConIndName на Domain и Range в зависимости от того, с каким внешним ключом было произведено соединение (DomainConIndFK или RangeConIndFK).

4. Делаем проекцию по Domain и Range и получаем экстенционал роли R.

Вот как выглядит это отображение в реляционной алгебре:

$$R_{RM^2}^E = \pi_{Domain, Range} (\rho_{ConIndName/Domain} (ConceptIndividual \bowtie_{ContIndPK=DomainConIndFK} (\rho_{ConIndName/Range} (ConceptIndividual \bowtie_{ContIndPK=RangeConIndFK} (\sigma_{RoleName='R'} (RoleIndividual \bowtie_{RoleIndPK=LinkRoleIndFK} (LinkRoleIndividual \bowtie_{LinkRoleFK=RolePK} Role))))))).$$

5.2. Отображения концептов, использующих конструкторы концептов и ролей.

Как уже было сказано, к данным концептам мы относим: дополнение, объединение, пересечение, экзистенциальное ограничение и ограничение значения. Их отображения включают в себя отображения базовых составляющих. При их описании мы будем пользоваться обозначениями, приведенными в таблице, подразумевая отображения, описанные в разделе 5.1.

Отображение дополнения концепта $\neg C$. Семантика ALC интерпретирует дополнение концепта следующим образом [27]:

$$(\neg C)_{DL}^I = \Delta - C_{DL}^I.$$

Это означает, что в результат входят все элементы области определения, которые не вошли в подмножество, которым интерпретируется концепт C . Опираясь на определение раздела 5, отображение для дополнения концепта C создается следующим образом:

1) выполняем проекцию ConceptIndividual по столбцу ConIndName . В результате мы получаем таблицу, содержащую имена всех индивидов области определения (которая задается в RM^2 экземплярами таблицы ConceptIndividual);

2) вычитаем из результата интерпретацию концепта C , заданную его отображением.

$$(\neg C)_{RM^S}^E = \pi_{\text{ConIndName}}(\text{ConceptIndividual}) - C_{RM^S}^E.$$

Отображение объединения концептов ($C \cup D$). Семантика ALC интерпретирует объединение концептов следующим образом [27]:

$$(C \sqcup D)_{DL}^I = C_{DL}^I \cup D_{DL}^I.$$

Это означает, что в результат входит объединение всех элементов из интерпретирующего множества концепта C и интерпретирующего множества концепта D . Соответственно, отображение объединения концептов будет представлять собой таблицу, в которой содержатся все элементы из результирующей таблицы для $C_{RM^2}^E$ и результирующей таблицы для $D_{RM^2}^E$, а именно:

$$(C \sqcup D)_{RM^2}^E = C_{RM^2}^E \cup D_{RM^2}^E.$$

Отображение пересечения концептов ($C \cap D$). Семантика ALC интерпретирует пересечение концептов следующим образом [27]:

$$(C \sqcap D)_{DL}^I = C_{DL}^I \cap D_{DL}^I.$$

Это означает, что в результат входят только те элементы, которые присутствуют одновременно и в интерпретирующем множестве концепта C и в интерпретирующем множестве концепта D . Соответственно, отображение пересечения концептов будет

представлять собой таблицу, в которой содержатся только те элементы, которые присутствуют одновременно в результирующих таблицах для $C_{RM^2}^E$ и $D_{RM^2}^E$:

$$(C \sqcap D)_{RM^2}^E = C_{RM^2}^E \cap D_{RM^2}^E.$$

Отображение конструктора экзистенциального ограничения ($\exists R.C$). Семантика ALC интерпретирует концепт экзистенциального ограничения следующим образом [27]:

$$(\exists R.C)_{DL}^I = \{ e \in \Delta \mid \text{существует } d \in \Delta \text{ такой, что } \langle e, d \rangle \in R^I \text{ и } d \in C^I \}.$$

Это означает, что результатом будет являться множество таких элементов, у которых хотя бы один R-последователь принадлежит интерпретирующему множеству концепта C .

Опираясь на определения раздела 5 и ранее описанные отображения:

1) соединяем $R_{RM^2}^E$ и $C_{RM^2}^E$ по атрибуту Range;

2) делаем проекцию по Domain.

Далее представлены формулы для данного отображения:

$$(\exists R.C)_{RM^2}^E = \pi_{\text{Domain}}(R_{RM^2}^E \bowtie_{\text{Range}} \pi_{\text{ConIndName}} C_{RM^2}^E).$$

Отображение конструктора ограничения значения ($\forall R.C$). Семантика ALC интерпретирует концепт ограничения значения следующим образом [27]:

$$(\forall R.C)_{DL}^I = \{ e \in \Delta \mid \text{для всех } d \in \Delta \text{ таких, что } \langle e, d \rangle \in R^I \text{ выполнено } d \in C^I \}.$$

Это означает, что результатом будет являться множество таких элементов, у которых все R-последователи принадлежат интерпретирующему множеству концепта C .

Существует два варианта ограниченного квантора общности:

1) $\forall y(P(y) \supset Q(x, y))$ по заданному множеству найти все его надмножества, удовлетворяющие заданному свойству;

2) $\forall y(Q(x, y) \supset P(y))$ по заданному множеству найти все его подмножества, удовлетворяющие заданному свойству.

В РА была введена дополнительная операция деления, которая имеет семантику, описанную в пункте 1. Было доказано, что она является излишней, так как выражимо через произведение и разность (см. раздел 4).

Что касается ALC, то конструктор ограничения значения имеет семантику, описанную в пункте 2. Для создания соответствующего отображения мы можем ввести в РА некую новую операцию деления с указанной семантикой, которая бы выглядела, например, так: $R_1[N/K]R_2$. Тогда отображение для конструктора $\forall R.C$ будут выглядеть следующим образом:

$$(\forall R.C)_{RM^2}^E = \pi_{\text{Domain}}(R_{RM^2}^E [\text{Range/ConIndName}] C_{RM^2}^E).$$

Но здесь возникает вопрос, а является ли эта новая операция принципиально новой? Если да, то ALC не является сводимой к классической РА. Но оказалось, что такой вариант представим через другие, стандартные операции РА, а именно:

$$(\forall R.C)_{RM^2}^E = \pi_{\text{Domain}} R_{RM^2}^E - \pi_{\text{Domain}}(R_{RM^2}^E \cap (\pi_{\text{Domain}} R_{RM^2}^E \times (\pi_{\text{Range}} R_{RM^2}^E - \pi_{\text{ConIndName}} C_{RM^2}^E))).$$

Покажем это.

Суть доказательства заключается в следующем. Среди всех пар индивидов, связанных отношением R возьмем множество, составляющее диапазон (Range) и найдем подмножество всех индивидов, не принадлежащих интерпретирующему подмножеству C. Далее, для данных индивидов возьмем множество их R-преемников, которое является подмножеством домена (Domain) отношения R, и вычтем данное подмножество из всех тех индивидов, которые интерпретируют R-домен. Результатом этих операций и будет являться искомое подмножество. Наглядность показана на рис. 3.

- 1) $\pi_{\text{ConIndName}} C_{RM^2}^E$ – множество имен индивидов, принадлежащих интерпретирующему множеству C;
- 2) $\pi_{\text{Range}} R_{RM^2}^E$ – множество имен индивидов, составляющих диапазон отношения R;
- 3) $(\pi_{\text{Range}} R_{RM^2}^E - \pi_{\text{ConIndName}} C_{RM^2}^E)$ – подмножество имен индивидов, входящих в диапазон отношения R и не принадлежащих интерпретирующему множеству C;
- 4) $\pi_{\text{Domain}} R_{RM^2}^E$ – множество имен индивидов, составляющих домен отношения R;
- 5) $(\pi_{\text{Domain}} R_{RM^2}^E \times (\pi_{\text{Range}} R_{RM^2}^E - \pi_{\text{ConIndName}} C_{RM^2}^E))$ – множество имен индивидов, принадлежащих домену R и имеющих в диапазоне R имена, принадлежащие множеству $(\pi_{\text{Range}} R_{RM^2}^E - \pi_{\text{ConIndName}} C_{RM^2}^E)$.

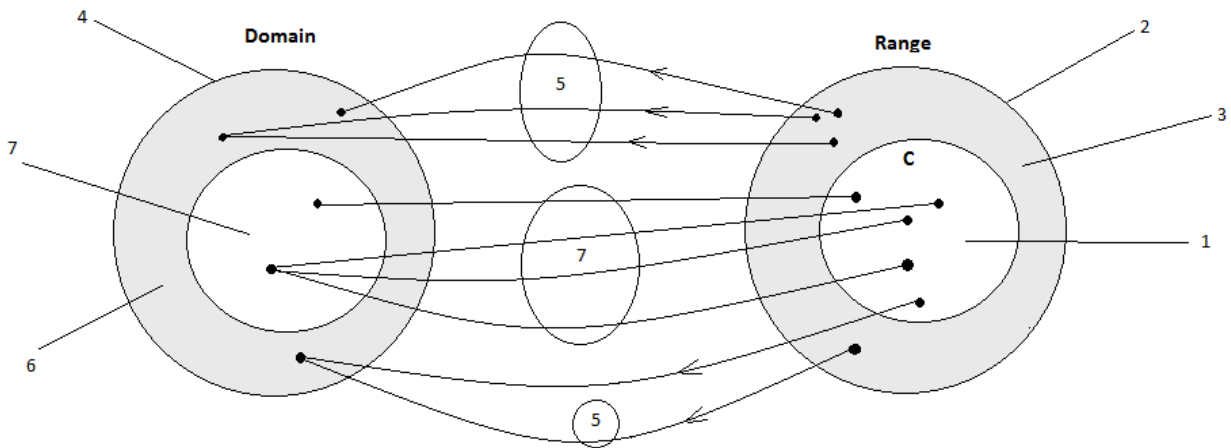


Рис. 3. Иллюстрация доказательства

$\pi_{\text{ConIndName}} C_{RM^2}^E$)) – подмножество пар индивидов, связанных отношением R, у которых R-последователи не принадлежат интерпретирующему множеству C;

б) $\pi_{\text{Domain}(R_{RM^2}^E)} \cap (\pi_{\text{Domain}(R_{RM^2}^E \times (\pi_{\text{Range}} R_{RM^2}^E - \pi_{\text{ConIndName}} C_{RM^2}^E))})$ – подмножество имен индивидов, которые составляют диапазон для пар, описанных в шаге 5.

Данное множество является подмножеством домена отношения R. Оно включает в себя все те индивиды, которые не входят в искомый результат. Его необходимо вычесть из общего множества, интерпретирующего домен отношения R, что мы и делаем на следующем шаге:

7) $\pi_{\text{Domain}(R_{RM^2}^E)} - \pi_{\text{Domain}(R_{RM^2}^E)} \cap (\pi_{\text{Domain}(R_{RM^2}^E \times (\pi_{\text{Range}} R_{RM^2}^E - \pi_{\text{ConIndName}} C_{RM^2}^E))})$ – подмножество индивидов домена отношения R, у которых любой R-последователь принадлежит такому подмножеству диапазона отношения R, все индивиды которого составляют подмножество интерпретирующего множества C.

Таким образом, мы доказали следующее утверждение: DL ALC сводима к реляционной структуре RM^2 .

Заключение

Мы сделали первый шаг на пути установления отображения между дескриптивной логикой и реляционной моделью, а именно, специфицировали специальный вариант реляционной структуры – бинарную реляционную структуру RM^2 . Привели отображение всех конструкций ALC – индивидов, концептов, ролей и конструкторов концептов – в операции над структурой RM^2 . Следующий шаг, который необходимо предпринять – это четко специфицировать множество операций модели RM^2 и привести их отображение в конструкторы концептов ALC, если это возможно будет сделать.

Как было отмечено в статье, ALC – это базис, на котором строятся другие DL путем введения дополнительных кон-

структоров концептов и ролей. В связи с этим представляют интерес исследования, которые установили бы связь RM^2 с этими расширениями.

1. *Чистякова И.С.* Онтолого-ориентированная интеграция данных в семантическом вебе // Проблемы програмування. – 2014. – № 2. – 3. – С. 188–196.
2. *Lenzerini M.* Data Integration: A Theoretical Perspective // Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2002). – N. Y.: ACM Press, 2002. – P. 233–246.
3. *Chang K.C.C., He B., et. al.* Structured databases on the web: Observations and Implications, ACM New York, NY, USA. – 2004. – 33: P. 61–70.
4. *Vysniauskas E, Nemuraite L.* Transforming ontology representation from OWL to relational database. Information Technology and Control. – 2006. – Vol. 35, N 3A. – P. 333–343.
5. *Carmen Martinez-Cruz, Ignacio J., Blanco M., Amparo Vila.* Ontologies versus relational databases: are they so different? A comparison. – Artif Intell Rev. – 2012. – 38. – P. 271–290.
6. *Brockmans S, Haase P, Hitzler P.* A meta-model and UML profile for rule-extended OWL DL ontologies // In: Sure JDY (ed) The semantic web: research and applications: 3rd European semantic web conference, ESWC. – 2006. – Vol. 4011. – P. 303–316.
7. *Colomb R., Gerber A. and Lawley M.* Issues in Mapping Metamodels in the Ontology Development Metamodel, Proceedings of the 1st International Workshop on the Model-Driven Semantic Web (MSDW). – 2004. – P. 20–24.
8. *Upadhyaya S. and Kumar P.* ERONTO: A Tool for Extracting Ontologies from Extended ER Diagrams, Proceedings of the 20th ACM Symposium on Applied Computing (SAC). – 2005. – P. 667–670.
9. *Astrova I., Korda N., Kalja A.* Storing OWL Ontologies in SQL Relational Databases. World Academy of Science, Engineering and Technology // International Journal of Computer, Control, Quantum and Information Engineering. – 2007. – Vol. 1, N 5. – P. 1246–1251.

10. *El-Ghalayini H., Odeh M., McClatchey R.* Engineering conceptual data models from domain ontologies: a critical evaluation // In: International Journal of Information Technology and Web Engineering (IJITWE), IGI Global. – 2007.
 11. *Juric D., Skocir Z.* Building owl ontologies by analyzing relational database schema concepts and wordnet semantic relations // In: Proceedings of the 9th international conference on telecommunications. ConTEL – 2007.
 12. *Stojanovic L., Stojanovic N., Volz R.* Migrating data-intensive web sites into the semantic web // In: SAC '02: Proceedings of the 2002 ACM symposium on applied computing. – ACM, New York, NY, USA. – 2002. – P. 1100–1107.
 13. *Champin P.A., Houben G., Thiran P.* Cross: an owl wrapper for reasoning on relational databases // In: Proceedings of the 26th international conference on Conceptual modeling. Springer, Auckland, New Zealand, 2007. – P. 502–517.
 14. *Doan A., Madhavan J., Domingos P., Halevy A.* Learning to map between ontologies on the semantic web // In: Proceedings of the eleventh international WWW conference. Hawaii. – 2002.
 15. *Sonia D., Khan S.* R2o transformation system: relation to ontology transformation for scalable data integration // In: IDEAS '08: proceedings of the 2008 international symposium on database engineering and applications. – ACM, New York, NY, USA. – 2008. – P. 291–295.
 16. *Astrova I., Kalja A.* Mapping of SQL Relational Schemata to OWL Ontologies. Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications, Elounda, Greece, August 18–20, 2006, P. 375–380.
 17. *Astrova I.* Towards the semantic web – an approach to reverse engineering of relational databases to ontologies // In: Advances in Databases and Information Systems: 9th East-European Conference, ADBIS 2005. – P. 111–122.
 18. *Shuefeng Zhou, Haiyun Ling, Mei Han, Huaiwei Zhang.* Ontology Generator from Relational Database Based on Jena. Computer and Information Science. – 2010. – Vol. 3, N 2. – P. 263–267.
 19. *Shuefeng Zhou, Guangwu Meng, Haiyun Ling.* Ontology Acquisition from Relational Database. Computer and Information Science, 2010. – Vol. 3, N 1. – P. 185–187.
 20. *Lei Zhang, Jing Li.* Automatic Generation of Ontology Based on Database // Journal of Computational Information Systems 7:4. – 2011. – P. 1148–1154.
 21. *Kashyap V.* Design and creation of ontologies for environmental information retrieval // In: Proceedings of the twelfth workshop on knowledge acquisition, modeling and management. Voyager Inn, Banff, Alberta, Canada. – 1999.
 22. *Jean S., Pierra G., AitAmeur Y.* Domain ontologies: a database-oriented analysis // In: Proceedings of the web information systems and technologies (WEBIST'2006). – 2006.
 23. *Fankam C., Jean S., Bellatreche L., AitAmeur Y.* Extending the ANSI/SPARC architecture database with explicit data semantics: an ontology-based approach // In: ECSA '08: Proceedings of the 2nd European conference on software architecture. Springer, Berlin. – 2008. – P. 318–321.
 24. *Roldán M., Montes JFA.* A tool for storing owl using database technology // In: Proceedings of the OWLED '05 workshop on OWL: experiences and directions. – 2005, Vol. 188. – P. 1–10.
 25. *Fankam C., Jean S., Bellatreche L., AitAmeur Y.* Extending the ANSI/SPARC architecture database with explicit data semantics: an ontology-based approach // In: ECSA '08: Proceedings of the 2nd European conference on software architecture. Springer, Berlin. – 2008. – P. 318–321.
 26. *Richard Barker.* “Case*Method: Entity Relationship Modelling”// Addison-Wesley. – 1990. – P. 240.
 27. *Дескрипционная логика (лекции) [электронный ресурс].* – Режим доступа: <http://lpcs.math.msu.su/~zolin/dl/>.
1. *Chystiakova I.S.* Ontology-oriented data integration on the Semantic Web (Онтолого-ориентированная интеграция данных в семантическом вебе). Problems in programming. – 2014. – N 2–3, P.188–196.
 2. *Lenzerini M.* Data Integration: A Theoretical Perspective // Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2002). – N. Y.: ACM Press, 2002. – P. 233–246.
 3. *Chang K.C.C., He B., et. al.* Structured databases on the web: Observations and Implica-

- tions, ACM New York, NY, USA. – 2004. – 33: P. 61–70.
4. *Vysniauskas E., Nemuraite L.* Transforming ontology representation from OWL to relational database. *Information Technology and Control.* – 2006, Vol. 35, N 3A. – P 333–343.
 5. *Carmen Martinez-Cruz, Ignacio J., Blanco M., Amparo Vila.* Ontologies versus relational databases: are they so different? A comparison. – *Artif Intell Rev.* – 2012. – 38. – P. 271–290.
 6. *Brockmans S, Haase P, Hitzler P.* A meta-model and UML profile for rule-extended OWL DL ontologies // In: Sure JDY (ed) *The semantic web: research and applications: 3rd European semantic web conference, ESWC.* – 2006. – Vol. 4011. – P. 303–316.
 7. *Colomb R., Gerber A. and Lawley M.* Issues in Mapping Metamodels in the Ontology Development Metamodel, *Proceedings of the 1st International Workshop on the Model-Driven Semantic Web (MSDW).* – 2004. – P. 20–24.
 8. *Upadhyaya S. and Kumar P.* ERONTO: A Tool for Extracting Ontologies from Extended ER Diagrams, *Proceedings of the 20th ACM Symposium on Applied Computing (SAC).* – 2005. – P. 667–670.
 9. *Astrova I., Korda N., Kalja A.* Storing OWL Ontologies in SQL Relational Databases. *World Academy of Science, Engineering and Technology // International Journal of Computer, Control, Quantum and Information Engineering.* – 2007. – Vol. 1, N 5. – P. 1246–1251.
 10. *El-Ghalayini H., Odeh M., McClatchey R.* Engineering conceptual data models from domain ontologies: a critical evaluation // In: *International Journal of Information Technology and Web Engineering (IJITWE), IGI Global.* – 2007.
 11. *Juric D., Skocir Z.* Building owl ontologies by analyzing relational database schema concepts and wordnet semantic relations // In: *Proceedings of the 9th international conference on telecommunications. ConTEL – 2007.*
 12. *Stojanovic L., Stojanovic N., Volz R.* Migrating data-intensive web sites into the semantic web // In: *SAC '02: Proceedings of the 2002 ACM symposium on applied computing.* – ACM, New York, NY, USA. – 2002. – P. 1100–1107.
 13. *Champin P.A., Houben G., Thiran P.* Cross: an owl wrapper for reasoning on relational databases // In: *Proceedings of the 26th international conference on Conceptual modeling.* Springer, Auckland, New Zealand, 2007. – P. 502–517.
 14. *Doan A., Madhavan J., Domingos P., Halevy A.* Learning to map between ontologies on the semantic web // In: *Proceedings of the eleventh international WWW conference.* Hawaii. – 2002.
 15. *Sonia D., Khan S.* R2o transformation system: relation to ontology transformation for scalable data integration // In: *IDEAS '08: proceedings of the 2008 international symposium on database engineering and applications.* – ACM, New York, NY, USA. – 2008. – P. 291–295.
 16. *Astrova I., Kalja A.* Mapping of SQL Relational Schemata to OWL Ontologies. *Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications, Elounda, Greece, August 18–20, 2006,* P. 375–380.
 17. *Astrova I.* Towards the semantic web – an approach to reverse engineering of relational databases to ontologies // In: *Advances in Databases and Information Systems: 9th East-European Conference, ADBIS 2005.* – P. 111–122.
 18. *Shuefeng Zhou, Haiyun Ling, Mei Han, Huaiwei Zhang.* Ontology Generator from Relational Database Based on Jena. *Computer and Information Science.* – 2010, Vol. 3, N 2. – P. 263–267.
 19. *Shuefeng Zhou, Guangwu Meng, Haiyun Ling.* Ontology Acquisition from Relational Database. *Computer and Information Science,* 2010, Vol. 3, N 1. – P. 185–187.
 20. *Lei Zhang, Jing Li.* Automatic Generation of Ontology Based on Database // *Journal of Computational Information Systems* 7:4. – 2011. – P. 1148–1154.
 21. *Kashyap V.* Design and creation of ontologies for environmental information retrieval // In: *Proceedings of the twelfth workshop on knowledge acquisition, modeling and management.* Voyager Inn, Banff, Alberta, Canada. – 1999.
 22. *Jean S., Pierra G., AitAmeur Y.* Domain ontologies: a database-oriented analysis // In: *Proceedings of the web information systems and technologies (WEBIST'2006).* – 2006.
 23. *Fankam C., Jean S., Bellatreche L., AitAmeur Y.* Extending the ANSI/SPARC architecture database with explicit data semantics: an ontology-based approach // In: *ECSA '08: Proceedings of the 2nd European conference on software architecture.* Springer, Berlin, – 2008. – P. 318–321.
 24. *Roldán M., Montes JFA.* A tool for storing owl using database technology // In: *Proceedings of the OWLED '05 workshop on OWL:*

- experiences and directions. – 2005, Vol. 188. – P. 1–10.
25. *Fankam C., Jean S., Bellatreche L., Aït-Ameur Y.* Extending the ANSI/SPARC architecture database with explicit data semantics: an ontology-based approach // In: ECSA '08: Proceedings of the 2nd European conference on software architecture. Springer, Berlin. – 2008. – P. 318–321.
26. *Richard Barker.* “Case*Method: Entity Relationship Modelling” // Addison-Wesley. – 1990. – P. 240.
27. *Description logic* (lectures) [digital resource]. – Access mode: <http://lpcs.math.msu.su/~zolin/dl/>.

Получено 20.04.2015

Об авторах:

Резниченко Валерий Анатольевич, кандидат физико-математических наук, старший научный сотрудник Института программных систем НАН Украины.
Количество научных публикаций в украинских изданиях – 57.
Количество научных публикаций в иностранных изданиях – 3.
ORCID orcid.org/0000-0002-4451-8931.

Чистякова Инна Сергеевна, аспирантка Института программных систем НАН Украины.
Количество научных публикаций в украинских изданиях – 7.
ORCID orcid.org/0000-0001-7946-3611.

Место работы авторов:

Институт программных систем
НАН Украины
03187, Киев,
Проспект Академика Глушкова, 40.
Тел.: +38(044) 526 5139; +38(044) 526 6249.
E-mail: vreznichenko_47@mail.ru,
inna_islyamova@ukr.net.