УДК 004.75

А. А. Бездушный

Московский физико-технический институт (ГУ)
Институтский пер., 9, Долгопрудный, 141700, Россия
E-mail: alexb@ccas.ru

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ СИСТЕМЫ ИНТЕГРАЦИИ ДАННЫХ НА ОСНОВЕ ОНТОЛОГИЙ

В статье предлагается формальная модель системы интеграции данных на основе онтологий и рассматривается вопрос переформулировки запросов в такой системе. Предлагается выбор диалектов дескриптивной логики и языков запросов, при которых построение требуемой переформулировки возможно, а также алгоритм построения переформулировки для важного класса систем интеграции данных.

Ключевые слова: интеграция данных, онтологии, дескриптивная логика, OWL.

Цель работы

Данная работа посвящена интеграции данных распределенных гетерогенных информационных источников в единую систему, которая позволила бы эффективно исполнять запросы на выборку взаимосвязанных данных из таких источников. Такого рода системы в литературе известны как *системы интеграции данных* (Enterprise Information Integration, EII [1; 2]), или *федеративные информационные системы* (FIS), *системы на основе механизма посредников* (MBIS).

Задача интеграции данных заключается в соединении данных из различных источников и предоставлении пользователю унифицированного представления этих данных. Система интеграции данных позволяет освободить пользователя от необходимости самостоятельно отбирать источники, в которых находится интересующая пользователя информация, обращаться к каждому источнику по отдельности и вручную сопоставлять и объединять данные из различных источников. При решении задачи интеграции данных возникают проблемы технического обеспечения взаимодействия с информационными источниками, синтаксической неоднородности форматов данных и, наконец, семантической неоднородности данных. В данной работе акцентируется внимание на последней, наиболее существенной проблемной области, которая затрагивает вопрос спецификации соответствия смысла (семантики) информации в разных информационных источниках.

Централизованная система интеграции данных предполагает следующую архитектуру. Имеется некоторое множество информационных источников, данные источников считаются представленными (или динамически представимыми) в некоторой единой модели данных, структура данных каждого источника регламентируется его собственной локальной схемой данных. Предполагается, что каждый информационный источник «обернут» промежуточным компонентом-адаптером, который отвечает за выборку сведений из источника в рамках единой модели данных, а также за предоставление стандартного технического интерфейса для обращения к источнику (сетевой протокол, язык запросов). Пользователь не взаимодействует с источниками напрямую, а обращается к выделенному компоненту-посреднику, который отвечает за обслуживание пользовательских запросов и взаимодействие с источниками. Пользователь формулирует свои запросы в терминах глобальной схемы данных (схемы данных посредника), которая проектируется для системы интеграции исходя из интересующих пользователя аспектов предметной области. Помимо глобальной схемы данных, в системе имеется набор описаний источников, задающих семантическое отображение между терминами глобальной схемы и терминами различных схем источников. Данные, соответствующие глобальной схеме, к которым

пользователь формулирует запросы, являются виртуальными в том смысле, что они не хранятся в системе физически. Вместо этого компонент-посредник использует описания источников для того, чтобы переформулировать пользовательский запрос в ряд запросов к источникам данных и на основе этих запросов динамически получить интересующий пользователя результат.

Целью настоящей работы является применение к задаче интеграции данных семантически богатых технологий, таких как онтологии и дескриптивная логика [3; 4]. Интеграция данных достаточно хорошо проработана для реляционных БД, однако для онтологий остается лишь слабо изученной актуальной областью исследований. Использование онтологий вместо реляционной модели данных позволяет повысить выразительные возможности системы интеграции данных, расширив ее сложными ограничениями целостности, логическим выводом, более гибкими механизмами спецификации отображения между глобальной схемой данных и схемами данных источников.

Применение онтологий к задаче интеграции распределенных данных целесообразно также в связи с тем, что онтологии в последнее время стали предметом стандартизации World Wide Web консорциума (W3C) в рамках проекта семантического веб (Semantic Web). Предложенные унифицированная модель данных RDF (Resource Decription Framework [5]) и язык веб-онтологий OWL (Web Ontology Language [3]) определили стандартный способ для семантически богатого описания распределенной в Интернет информации. Недавно была утверждена также спецификация языка запросов SPARQL [6], предназначенного для выборки информации из RDF-источников данных. Применение указанных технологий семантического веб в системе интеграции данных представляется важным и перспективным направлением.

В данной работе рассматривается класс систем интеграции данных, которые мы будем называть *системами интеграции данных на основе онтологий*. Применение в системе интеграции данных аппарата дескриптивной логики вместо реляционной модели данных ставит новые актуальные вопросы и задачи и приводит в ряде случаев к трудноразрешимости или неразрешимости задачи ответа на запросы в такой системе.

Основной задачей, которую мы будем рассматривать, является *переформулировка запросов* в системе интеграции данных на основе онтологий, т. е. построение на основе пользовательского запроса, заданного в терминах глобальной схемы данных, распределенного запроса к источникам данных. Сформулируем ключевые параметры рассматриваемого в данной работе класса задач о переформулировке запросов в системе интеграции данных.

Глобальная модель данных. В системе интеграции данных на основе онтологий будем исходить из унифицированной модели данных RDF.

Модель данных источников. Будем считать, что данные источников представлены (или представимы) в унифицированной модели данных RDF.

Язык спецификации глобальной схемы данных. Будем представлять глобальную схему данных онтологией на языке веб-онтологий OWL, используя некоторое подмножество конструкций OWL (выбор которого мы будем рассматривать ниже). Иначе говоря, формально глобальная схема представляется онтологией на некотором диалекте дескриптивной логики $\mathcal{L}_\Gamma$.

Язык спецификации схем данных источников. Будем считать, что схемы данных источников также представлены с помощью некоторого фрагмента языка OWL, т. е. на некотором диалекте дескриптивной логики $\mathcal{L}_\Delta$.

Язык спецификации отображения схем (описания источников). Отображение онтологий задается в общем случае парами запросов на некотором языке запросов $\mathcal{QL}_\mathcal{F}$ в терминах глобальной онтологии и в терминах онтологий источников соответственно.

Язык пользовательских запросов. Будем рассматривать запросы на языке SPARQL, однако для сохранения разрешимости задачи ответа на запросы в системе интеграции данных мы будем допускать лишь ограниченное число конструкций SPARQL. Формально мы рассматриваем некоторый язык запросов $\mathcal{QL}_\Gamma$ над глобальной онтологией.

Язык переформулированных запросов. Будем рассматривать возможность переформулировки запроса в некоторое подмножество SPARQL, т. е. формально некоторый язык запросов $\mathcal{QL}_\mathrm{R}$ над онтологиями источников (выбор которого мы будем рассматривать ниже).

Выбор конкретных диалектов дескриптивной логики и языков запросов определяет наличие решения задачи, т. е. возможность переформулировки запроса в рассматриваемой системе интеграции данных. В данной работе предлагается формальная математическая модель системы интеграции данных на основе онтологий и рассматривается вопрос выбора параметров задачи переформулировки запросов в такой системе. Предлагается выбор диалектов дескриптивной логики и языков запросов, при которых построение требуемой переформулировки возможно. Рассматриваются условия, при которых переформулированный запрос может быть транслирован в язык запросов SQL, что означает важную на практике возможность оптимизированного исполнения запросов к источникам, представленным реляционными базами данных, непосредственно реляционной СУБД, и предлагается соответствующий алгоритм переформулировки.

Актуальность и новизна

Технологии семантического веб (Semantic Web) являются молодым и перспективным направлением развития современной информационной индустрии. Стандартизация World Wide Web консорциумом (W3C) в 2004 г. модели описания информационных ресурсов RDF (Resource Description Framework [5]) и языка веб-онтологий OWL (Web Ontology Language [3]) положила начало интенсивному развитию и внедрению семантических технологий. В январе 2008 г. была утверждена также спецификация языка SPARQL [6], предназначенного для формулировки запросов на выборку информации из информационных источников, данные которых могут быть представлены в унифицирующей модели данных RDF.

Применение семантически богатых технологий, таких как OWL-онтологии и дескриптивная логика [3; 4], к задаче интеграции данных является весьма перспективным подходом. Онтологии позволяют специфицировать структуру и семантику терминов системы интеграции и источников данных, выразить различные формы ограничений целостности в системе интеграции данных. Благодаря применению конструкций дескриптивной логики механизм спецификации соответствия глобальных терминов и терминов источников данных приобретает бóльшие выразительные возможности, нежели доступные в реляционных системах интеграции данных. Это, в частности, позволяет расширить спектр возможных источников, которые могут быть интегрированы в единую систему, а также сделать спецификацию соответствия терминов более краткой. Другим стимулом для применения технологий семантического веб в контексте интеграции данных распределенных гетерогенных информационных источников является модель данных RDF, которая специально спроектирована для представления распределенной в сети информации. RDF позиционируется как унифицирующая модель данных, предназначенная для описания информационных ресурсов веб таким образом, чтобы любые совместимые со стандартами RDF и OWL сторонние системы могли единым образом автоматически проинтерпретировать смысл этих описаний (принцип *семантической интероперабельности*).

Следует отметить, что в отличие от реляционных баз данных онтологии могут выражать некоторую форму неполноты информации, что вносит существенные сложности при рассмотрении в контексте онтологий задач, решенных для реляционных баз данных. Принцип «открытости» онтологий (open-world assumption) предполагает, что в высказываниях онтологии зафиксирована лишь часть сведений об объектах и терминах, при этом могут существовать и другие сведения, не указанные в онтологии явно. Более того, предполагается, что некоторые дополнительные сведения могут быть получены дедуктивным логическим выводом на основе зафиксированных в онтологии фактов и высказываний. Принцип «открытости» определяет также особенности ответа на поисковые запросы к онтологии. Для вычисления множества ответов на запрос относительно онтологии требуется провести логический вывод на основе исходного множества фактов онтологии и ее аксиом, что может приводить к трудноразрешимости или неразрешимости задачи. В частности, задача ответа даже на простые конъюнктивные запросы (вид запросов, аналогичный select-project-join запросам в SQL) для дескриптивной логики $\mathcal{SHOIN}^{(D)}$ [7], соответствующей диалекту языка веб-онтологий OWL-DL, на момент написания статьи не решена, и вопрос ее разрешимости остается открытым.

Рассматриваемая в настоящей работе задача ответа на запросы в системе интеграции данных, а также смежная с ней задача ответа на запросы с использованием представлений, достаточно детально изучена для реляционных баз данных (см. обзор [8], а также работы [9–12]) и в то же время остается в большой степени областью актуальных исследований в контексте онтологий и дескриптивной логики. Можно отметить лишь небольшой ряд работ [13–19], в той или иной степени рассматривающих эту задачу для разных диалектов дескриптивной логики, разных языков запросов и разных механизмов спецификации представлений (отображения онтологий), при этом самые актуальные работы в этой области датируются 2008 г. Ключевой проблемой при рассмотрении такого рода задачи для дескриптивной логики является ее трудноразрешимость или неразрешимость для достаточно выразительных диалектов дескриптивной логики. Соответственно важно выбрать параметры задачи таким образом, чтобы она имела практическую ценность и определенные преимущества и при этом оставалась разрешимой за полиномиальное время относительно объема данных (фактов) онтологий (класс сложности $\mathcal{P}$) или даже ограничивалась логарифмическим объемом памяти относительно объема данных (класс $\mathcal{LOGSPACE}$). Последнее позволяет сохранить возможность применения реляционных СУБД для хранения фактов онтологий и ответа на запросы, формулируемые в системе к таким множествам фактов, что важно с практической точки зрения. В данной работе делается акцент на практическом применении OWL-онтологий и дескриптивной логики в задаче интеграции данных и формулируется разрешимый класс задач переформулировки запросов в системе интеграции данных на основе онтологий.

Предварительные определения

Дескриптивная логика. Математической основой языка веб-онтологий OWL является так называемая дескриптивная (описательная) логика (Description logics, $\mathcal{DL}$ [4]). Дескриптивная логика – это семейство языков представления знаний, предназначенных для выражения терминологического знания о предметной области, семантика которого задается отображением в исчисление предикатов первого порядка.

Дескриптивная логика оперирует двумя видами отношений – унарными, называемыми *концептами* (классами), и бинарными, называемыми *ролями* (свойствами). Концепты и роли могут быть *атомарными* или сложными, определенными с помощью *конструкторов* языка дескриптивной логики на основе других концептов и ролей. Роли принято делить на *абстрактные роли*, связывающие объекты, и *атрибуты* (конкретные роли), связывающие объекты со значениями примитивного типа данных. *Онтология* в дескриптивной логике определяется как $\mathcal{O} \stackrel{\text{def}}{=} \{\mathcal{T}, \mathcal{A}\}$,

где $\mathcal{T}$ – *терминология* (англ. TBox), множество терминологических аксиом, т. е. высказываний о *терминах* онтологии – *концептах* и *ролях* [1];

$\mathcal{A}$ – *множество фактов* [2] (ABox), высказываний об объектах [3] онтологии.

Если сравнивать с реляционными базами данных, то терминология соответствует схеме базы данных, а множество фактов – данным. Однако в отличие от реляционной базы данных в онтологии имеющиеся факты рассматриваются как неполная информация об описываемых

[1] В ряде работ принято выделять отдельные множества высказываний о концептах (TBox) и о ролях (RBox), в настоящей работе мы опускаем это разделение, а концепты и роли обозначаем также понятием «термины».

[2] В ряде работ для некоторых диалектов $\mathcal{DL}$ множество фактов $\mathcal{A}$ не вводится в систему определений, в частности для вариаций $\mathcal{AL}$, а также для $\mathcal{O}$-диалектов, в которых факты могут быть выражены номиналами и вложением концептов: $C(a) \rightarrow \{a\} \sqsubseteq C$, $R(a, b) \rightarrow \{a\} \sqsubseteq \exists R.\{b\}$. Мы приводим эти теории к общей системе определений, вводимой в данной работе.

[3] В англоязычной литературе используется термин «individual», мы будем использовать термин «объект».

информационных ресурсах, а также предполагается, что на основе терминологии и исходных фактов могут быть дедуцированы производные факты.

Допустимые формы высказываний терминологии $\mathcal{T}$ и множества фактов $\mathcal{A}$ определяются используемым языком дескриптивной логики $\mathcal{L}$. В табл. 1 приведено сопоставление выразительных возможностей ряда упоминаемых в настоящей работе диалектов дескриптивной логики, начиная с языков из класса $\mathcal{AL}$ (атрибутивный язык [4]) и заканчивая диалектами языка OWL (OWL-Lite, OWL-DL [3]). Для сравнения приводятся также характеристики языка RDF Schema [20].

Для тех конструкций, которые представляются в языке OWL конкретным элементом, указаны наименования таких элементов OWL. Формальное определение упоминаемых в таблице конструкторов и аксиом приводится ниже. Кроме того, в табл. 1 приводится расшифровка принятой символьной нотации наименований диалектов дескриптивной логики, которую мы будем использовать в дальнейшем. Для каждой ключевой характеристики языка дескриптивной логики во втором столбце указана соответствующая литера символьной нотации. Наименование диалекта получается соединением соответствующих литер, например $\mathcal{ALCNR}$ = = $\mathcal{AL} + \mathcal{C} + \mathcal{N} + \mathcal{R}$, атрибутивный язык, расширенный сложным отрицанием, – ограничениями мощности и сложными ролями.

Рассмотрим формальное определение допустимых высказываний онтологии для дескриптивной логики $\mathcal{SHOIN}^{(\mathcal{D})}$, которая соответствует диалекту OWL-DL [3] (см. также [7]). Пусть $\mathbb{C}$ – бесконечное множество констант, состоящее из двух непересекающихся подмножеств $\mathbb{C}_a$ объектов и $\mathbb{C}_d$ значений примитивных типов данных. Пусть $\mathbb{N}_C$ – множество имен концептов, $\mathbb{N}_{Ra}$ – множество имен абстрактных ролей, $\mathbb{N}_{Rd}$ – множество имен атрибутов. Множество всех имен будем также называть *алфавитом* $\mathbb{N} = \mathbb{N}_C \cup \mathbb{N}_{Ra} \cup \mathbb{N}_{Rd}$. *Атомарной ролью* будем называть $P \in \mathbb{N}_{Ra} \cup \mathbb{N}_{Rd}$, *атомарным концептом* – $A \in \mathbb{N}_C$. *Абстрактной ролью* называется роль $R_a \in \mathbb{N}_{Ra}$ или *обратная* роль R_a^- для любой $R_a \in \mathbb{N}_{Ra}$. *Атрибутом* называется роль $R_d \in \mathbb{N}_{Rd}$.

Множество $\mathcal{SHOIN}^{(\mathcal{D})}$ *концептов* C определяется согласно следующей синтаксической нотации (пояснение ниже):

$$C \to \top \mid \bot \mid A \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \forall R_a.C \mid \exists R_a.C \mid \geq n\, R_s \mid \leq n\, R_s \mid \{a_1, ..., a_n\} \mid \geq n\, R_d \mid$$
$$\mid \leq n\, R_d \mid \forall R_{d1}..R_{dn}.D \mid \exists R_{d1}..R_{dn}.D$$
$$D \to d \mid \{c_1, ..., c_n\}.$$

Здесь A – атомарный концепт, $C_{1..n}$ – концепт, R_a – абстрактная роль, $R_{d1..n}$ – атрибут, R_s – абстрактная роль, не имеющая транзитивных вложенных ролей (с учетом рефлексивно-транзитивного замыкания, см. [7]), $a_i \in \mathbb{C}_a$, $c_i \in \mathbb{C}_d$, $n \geq 1$, d – допустимый примитивный тип данных (см. [7]).

Неформально, сложный концепт задает некоторый класс объектов на основе других концептов и ролей с помощью следующих конструкторов:

- верхний концепт $\top$, содержащий все объекты;
- нижний концепт $\bot$, не содержащий объектов;
- атомарный концепт $A \in \mathbb{N}_C$;
- дополнение концепта $\neg C$ – задает класс объектов, не принадлежащих к C;
- конъюнкция концептов $C_1 \sqcap C_2$ – задает класс объектов, принадлежащих одновременно к обоим концептам C_1 и C_2;
- дизъюнкция концептов $C_1 \sqcup C_2$ – задает класс объектов, принадлежащих к C_1 или C_2;
- универсальное ограничение $\forall R_a.C$ – задает класс объектов, которые роль R_a связывает только с объектами из класса C (аналогично $\forall R_{d1}..R_{dn}.D$ – со значениями типа D);

Таблица 1

Сопоставление диалектов дескриптивной логики

Характеристика	Нотация	*RDFS* [20]	$\mathcal{ALE}$ [4]	$\mathcal{ALN}$ [4]	$\mathcal{ALCNR}$ [4]	$\mathcal{DL}$-Lite$^{\mathcal{A}}$ [18; 21–22]	$\mathcal{SHIF}^{(\mathcal{D})}$ OWL-Lite [3]	$\mathcal{SHOIN}^{(\mathcal{D})}$ OWL-DL [3]	$\mathcal{SROIQ}^{(\mathcal{D})}$ OWL 1.1 [23]
Конструкторы концептов и ролей									
$\top$ – Верхний концепт (Thing)	$\mathcal{AL}$		+	+	+	+	+	+	+
$\bot$ – Нижний концепт (Nothing)	$\mathcal{AL}$		+	+	+	только $A_1 \sqcap A_2 \sqsubseteq \bot$	только $a_1 \sqcap a_2 \sqsubseteq \bot$	+	+
$\neg A$ – Атомарное отрицание	$\mathcal{AL}$		+	+	+	+	+	+	+
$\neg C$ – Сложное отрицание (complementOf)	$\mathcal{C}$				+			+	+
$C_1 \sqcap C_2$ – Конъюнкция (intersectionOf)	$\mathcal{AL}$	неявно	+	+	+	+	+	+	+
$C_1 \sqcup C_2$ – Дизъюнкция (unionOf)	$\mathcal{U}$							+	+
$\forall R.C$ – Универсальные ограничения (allValuesFrom)	$\mathcal{AL}$	неявно range, domain	+	+	+		+	+	+
$\exists R$ ($\exists R.\top$) – Простые экзистенциальные ограничения	$\mathcal{AL}$		+	+	+	+	+	+	+
$\exists R.C$ – Экзистенциальные ограничения (someValuesFrom)	$\mathcal{E}$		+			+	+	+	+
$\geq n\ R_s, \leq n\ R_s$ – Ограничения мощности (min/maxCardinality)	$\mathcal{N}$			+	+		только 0 или 1	+	+
$\geq n\ R_s.C, \leq n\ R_s.C$ – Сложные ограничения мощности	$\mathcal{Q}$								+
$\{a_1, ..., a_n\}$ – Номиналы, перечисление класса (oneOf, hasValue)	$\mathcal{O}$							+	+
Примитивные типы данных (DatatypeProperty)	($\mathcal{D}$)	+				+	+	+	+
R^- – Обратные роли (inverseOf)	$\mathcal{I}$					+	+	+	+

Окончание табл. 1

Характеристика	Нотация	*RDFS* [20]	$\mathcal{ALE}$ [4]	$\mathcal{ALN}$ [4]	$\mathcal{ALCNR}$ [4]	$\mathcal{DL}$-*Lite*$^{\mathcal{A}}$ [18; 21–22]	$\mathcal{SHIF}^{(\mathcal{D})}$ OWL-Lite [3]	$\mathcal{SHOIN}^{(\mathcal{D})}$ OWL-DL [3]	$\mathcal{SROIQ}^{(\mathcal{D})}$ OWL 1.1 [23]
Сложные роли, рефлексивность, отрицание ролей	$\mathcal{R}$				только $R_1 \sqcap R_2$	только $\neg R$			
Высказывания терминологии									
$C_1 \sqsubseteq C_2$ – Вложение концептов, GCI (subClassOf)	$\mathcal{AL}$	+	+	+	+	+ огранич.	+	+	+
$R_1 \sqsubseteq R_2$ – Вложение ролей (subPropertyOf)	$\mathcal{H}$	+				+	+	+	+
Funct(R), или $\top \sqsubseteq \leq 1\ R$ – Функциональные роли (FunctionalProperty)	$\mathcal{F}$					+ огранич.	+	+	+
Trans(R) – Транзитивные роли (TransitiveProperty)	$\mathcal{S}$ *						+	+	+
Высказывания множества фактов									
$C(a)$, $R(a, b)$ – Принадлежность к концепту, роли	–	+	+	+	+	+	+	+	+

* Класс $\mathcal{S}$ объединяет $\mathcal{AL}$, $\mathcal{C}$ и поддержку обратных ролей.

• экзистенциальное ограничение $\exists R_a.C$ – задает класс объектов, связываемых ролью R_a с объектом из класса C (аналогично $\exists R_{d1}..R_{dn}.D$ – со значением типа данных D);

• ограничения мощности $\geq n\ R_s$, $\geq n\ R_d$ (и соответственно $\leq n\ R_s$, $\leq n\ R_d$) – задают классы объектов, связываемых заданной ролью не менее чем (не более чем) с n объектами (значениями);

• перечисление $\{a_1, ..., a_n\}$ – задает класс перечислением множества объектов.

Теперь мы готовы дать определение допустимых видов высказываний онтологии. Терминология для $\mathcal{SHOIN}^{(\mathcal{D})}$ (OWL-DL) может включать следующие аксиомы:

• $C_1 \sqsubseteq C_2$, где C_1, C_2 – концепты (аксиома вложения концептов);

• $R_{a1} \sqsubseteq R_{a2}$, где R_{a1}, R_{a2} – абстрактные роли (аксиома вложения абстрактных ролей);

• $R_{d1} \sqsubseteq R_{d2}$, где R_{d1}, R_{d2} – атрибуты (аксиома вложения атрибутов);

• Trans(R_a), где R_a – абстрактная роль (аксиома транзитивности роли).

Множество фактов онтологии $\mathcal{A}$ содержит высказывания формы $C(a)$ или $R(a, b)$, где C – концепт, R – роль, $a \in \mathbb{C}_a$, $b \in \mathbb{C}$.

В данной работе нам понадобится также диалект дескриптивной логики $\mathcal{DL}$-*Lite* [18; 21–22]. Его основное отличие от $\mathcal{SHOIN}^{(\mathcal{D})}$ (т. е. OWL-DL) заключается в запрете ряда конструкто-

ров, а также в ограничении формы аксиом вложения концептов. Рассмотрим определение языка $\mathcal{DL}$-$\mathcal{Lite}^A$, эквивалентное [18]. Как и для $\mathcal{SHOIN}^{(\mathcal{D})}$, абстрактная роль $R_a \in \mathbb{N}_{Ra} \cup \{R_a^- \mid R_a \in \mathbb{N}_{Ra}\}$, атрибут $R_d \in \mathbb{N}_{Rd}$, атомарный концепт $A \in \mathbb{N}_C$. Концепты в $\mathcal{DL}$-$\mathcal{Lite}^A$ определяются согласно следующей синтаксической нотации (C_L – простой концепт, C – концепт):

$C_L \rightarrow A \mid \exists R_a \mid \exists R_d$;

$C \rightarrow \top \mid A \mid \neg A \mid C_1 \sqcap C_2 \mid \exists R_a \mid \exists R_d \mid \exists R_a.C \mid \neg\exists R_a \mid \neg\exists R_d$.

Терминология $\mathcal{DL}$-$\mathcal{Lite}^A$ может включать только следующие ограниченные формы аксиом:

• $C_L \sqsubseteq C$, где C_L – простой концепт, C – концепт, согласно приведенной нотации (ограниченная аксиома вложения концептов);

• $R_{a1} \sqsubseteq R_{a2}$, $R_{a1} \sqsubseteq \neg R_{a2}$, где R_{a1}, R_{a2} – абстрактные роли (аксиомы вложения и различия абстрактных ролей);

• $R_{d1} \sqsubseteq R_{d2}$, $R_{d1} \sqsubseteq \neg R_{d2}$, где R_{d1}, R_{d2} – атрибуты (аксиомы вложения и различия атрибутов);

• $\rho(R_d) \sqsubseteq d$, где R_d – атрибут, d – допустимый примитивный тип данных, ρ задает множество всех значений R_d (аксиома типа значений атрибута);

• Funct(R) (аксиома функциональной роли при отсутствии конструктора $\leq 1\ R$).

При этом в $\mathcal{DL}$-$\mathcal{Lite}^A$-терминологиях для функциональных ролей не допускается определение вложенных ролей, а также их использование в конструкциях $\exists R_a.C$. На основе $\mathcal{DL}$-$\mathcal{Lite}^A$ определяются диалекты $\mathcal{DL}$-$\mathcal{Lite}^R$ и $\mathcal{DL}$-$\mathcal{Lite}^F$. Язык $\mathcal{DL}$-$\mathcal{Lite}^R$ получаем из $\mathcal{DL}$-$\mathcal{Lite}^A$ запретом аксиом Funct(R), язык $\mathcal{DL}$-$\mathcal{Lite}^F$ – запретом аксиом вложения ролей.

Теперь перейдем от рассмотрения синтаксиса высказываний онтологии к их семантике. Введем ключевые понятия интерпретации и модели онтологии, которые формализуют семантику онтологий (см. также [7]). *Интерпретацией* $\mathcal{I}$ онтологии $\mathcal{O} = \{\mathcal{T}, \mathcal{A}\}$ называется пара $(\Delta^I, \cdot^I)$, где Δ^I – непустое множество объектов, называемое областью определения интерпретации $\mathcal{I}$, а $\cdot^I$ – функция интерпретации, которая сопоставляет каждому концепту терминологии $\mathcal{T}$ некоторое подмножество Δ^I, а каждой роли – подмножество декартова произведения $\Delta^I \times \Delta^I$. Функция интерпретации определяется индуктивно, в зависимости от конструкций используемого диалекта дескриптивной логики. Следующее определение функции интерпретации вводится для $\mathcal{SHOIN}^{(\mathcal{D})}$, т. е. OWL-DL (обозначения те же, что и выше):

• $P^I \subseteq \Delta^I \times \Delta^I$ (каждой атомарной роли P сопоставляется подмножество $\Delta^I \times \Delta^I$);

• $(a, b) \in P^I \leftrightarrow (b, a) \in (P^-)^I$ (обратные роли P^-);

• $(a, b) \in R^{+I} \wedge (b, c) \in R^{+I} \rightarrow (a, c) \in R^{+I}$ (для любой транзитивной роли R^+ : Trans(R^+));

• $A^I \subseteq \Delta^I$ (каждому атомарному концепту сопоставляется подмножество Δ^I);

• $\top^I = \Delta^I$ (верхний концепт представляет Δ^I);

• $\bot^I = \varnothing$ (нижний концепт – пустое множество);

• $(\neg C)^I = \Delta^I \setminus C^I$ (дополнение C^I в Δ^I);

• $(C_1 \sqcap C_2)^I = C_1^I \cap C_2^I$ (пересечение);

• $(C_1 \sqcup C_2)^I = C_1^I \cup C_2^I$ (объединение);

• $(\forall R.C)^I = \{a \in \Delta^I \mid \forall b \in \Delta^I : (a, b) \in R^I \rightarrow b \in C^I\}$ (аналогично $\forall R_{d1}..R_{dn}.D$);

• $(\exists R.C)^I = \{a \in \Delta^I \mid \exists b \in C^I : (a, b) \in R^I\}$ (аналогично $\exists R_{d1}..R_{dn}.D$);

• $(\geq n\ R_s)^I = \{a \in \Delta^I \mid |\{b \mid (a, b) \in R_s^I\}| \geq \mathrm{n}\}$ (аналогично $\geq n\ R_d$);

• $(\leq n\ R_s)^I = \{a \in \Delta^I \mid |\{b \mid (a, b) \in R_s^I\}| \leq n\}$ (аналогично $\leq n\ R_d$);

• $(\{a_1, ..., a_n\})^I = \{a_1^I, ..., a_n^I\}$ (перечисление класса).

Аналогично функция интерпретации вводится и для других диалектов дескриптивной логики (для $\mathcal{DL}$-$\mathcal{Lite}^A$ также $(\rho(R_d))^I = \{b \in \Delta^I \mid (a, b) \in R_d^I\}$).

Интерпретация $\mathcal{I}$ называется *моделью* онтологии $\mathcal{O} = \{\mathcal{T}, \mathcal{A}\}$, если она удовлетворяет всем высказываниям в $\mathcal{T}$ и $\mathcal{A}$. Для $\mathcal{SHOIN}^{(\mathcal{D})}$ (OWL-DL) это означает, что:

- для любой аксиомы $C_1 \sqsubseteq C_2$ верно $C_1^I \subseteq C_2^I$;
- для любой аксиомы $R_1 \sqsubseteq R_2$ верно $R_1^I \subseteq R_2^I$;
- для любой аксиомы $C(a)$ верно $a^I \in C^I$;
- для любой аксиомы $R(a, b)$ верно $(a^I, b^I) \in R^I$.

Множество моделей онтологии $\mathcal{O}$ будем обозначать $\mathcal{M}_\mathcal{O}$. Онтология $\mathcal{O}$ называется *выполнимой*, если она имеет модели, т. е. $\mathcal{M}_\mathcal{O} \neq \varnothing$. Интерпретация $\mathcal{I} \in \mathcal{M}_\mathcal{O}$ называется *моделью* концепта C относительно $\mathcal{O}$, если $C^I \neq \varnothing$. Концепт называется *выполнимым*, если он имеет модели.

Отношение *логического следствия* $\models$ определяется в соответствии с функцией интерпретации следующим образом:

- $\mathcal{I} \models C(a) \Leftrightarrow a^I \in C^I$;
- $\mathcal{I} \models R(a, b) \Leftrightarrow (a^I, b^I) \in R^I$;
- $\mathcal{I} \models R_1 \sqsubseteq R_2 \Leftrightarrow R_1^I \subseteq R_2^I$;
- $\mathcal{I} \models C_1 \sqsubseteq C_2 \Leftrightarrow C_1^I \subseteq C_2^I$.

Индуктивно отношение логического следствия вводится для любых высказываний α ($\mathcal{I} \models \alpha$), являющихся логическими комбинациями указанных видов простых высказываний. Например, $\mathcal{I} \models C(a) \wedge \exists b\, R(a, b) \Leftrightarrow a^I \in C^I \wedge \exists b^I: (a^I, b^I) \in R^I$. Высказывание α *логически следует* из онтологии $\mathcal{O}$ ($\mathcal{O} \models \alpha$), если $\mathcal{I} \models \alpha\ \forall \mathcal{I} \in \mathcal{M}_\mathcal{O}$.

Будем говорить, что объект $a \in \mathbb{C}_a$ *принадлежит* к концепту C относительно $\mathcal{O}$, если $\mathcal{O} \models C(a)$ (т. е. $a^I \in C^I\ \forall \mathcal{I} \in \mathcal{M}_\mathcal{O}$). Будем говорить, что концепт C_1 *содержится* в концепте C_2 относительно $\mathcal{O}$ ($C_1 \sqsubseteq_\mathcal{O} C_2$), если $\mathcal{O} \models C_1 \sqsubseteq C_2$ (т. е. $C_1^I \subseteq C_2^I\ \forall \mathcal{I} \in \mathcal{M}_\mathcal{O}$). Концепт C_1 *эквивалентен* концепту C_2 относительно $\mathcal{O}$ ($C_1 \equiv_\mathcal{O} C_2$), если $C_1 \sqsubseteq_\mathcal{O} C_2$ и $C_2 \sqsubseteq_\mathcal{O} C_1$ (т. е. $C_1^I = C_2^I\ \forall \mathcal{I} \in \mathcal{M}_\mathcal{O}$). Будем говорить, что некоторый язык дескриптивной логики *принимает уникальность имен*, если различные константы представляют различные сущности предметной области, т. е. не допускается эквивалентность объектов, представленных различными константами: $a \neq b \rightarrow a^I \neq b^I$. Уникальность имен принята в ряде диалектов $\mathcal{DL}$, в том числе в $\mathcal{DL}$-$\mathcal{Lite}$, в OWL уникальность имен не принята.

Язык запросов. Для описания запросов в соответствии с моделью данных RDF было предложено достаточно много диалектов языков, на основе пересмотра которых сформирован утвержденный недавней рекомендацией W3C язык SPARQL [6] (от 15 января 2008 г.). В роли языка запросов в системе интеграции данных на основе онтологий мы будем рассматривать именно язык SPARQL, так как он является фактическим стандартом и специально спроектирован для унифицирующей модели данных RDF, а также предполагает совместное использование с языком веб-онтологий OWL.

В то же время использование полного набора конструкций SPARQL в контексте рассматриваемой задачи может привести к ее неразрешимости. В дальнейшем нам понадобится рас-

сматривать более ограниченные языки запросов. В этой связи введем определение класса запросов *SPARQL-BGP*, допускающего только базисную форму запросов SPARQL – простой шаблон подграфа (Basic Graph Pattern, BGP [6]). Ограничивая возможности SPARQL поиском по таким шаблонам и выборкой значений соответствующих вершин RDF-графа, мы получим примерный аналог класса запросов SPJ (selection-projection-join) в реляционной модели данных, который является основой для построения алгоритмов планирования запросов в реляционных системах интеграции данных. Ниже приведен пример запроса из класса SPARQL-BGP, предназначенного для выборки информации о публикациях всех докторов наук в отделении математических наук (ОМН) за 2008 г. (определение синтаксиса SPARQL приведено в [6]):

```
prefix : <http://umeta.ru/...>
select ?x, ?y
where {
  ?x :academicDegree ?d.
  ?d :academicDegreeLevel
            <urn:degree:doctorate>.
  ?x :worksFor ?o.
  ?o :superOrganization <urn:orgid:omn>.
  ?y :author ?x.
  ?y :year '2008'.
}
```

Формальным эквивалентом класса запросов SPARQL-BGP являются конъюнктивные запросы. Введем формальные определения для конъюнктивных запросов на выборку данных в контексте $\mathcal{DL}$-онтологий, которые нам понадобятся в дальнейшем.

Конъюнктивным запросом $Q(\underline{X})$ [4] над терминологией $\mathcal{T}$, выраженной на языке дескриптивной логики $\mathcal{L}$, называется формула $Q(\underline{X}) \leftarrow B(\underline{X}, \underline{Y})$, где $B(\underline{X}, \underline{Y}) = \wedge_{i=1..k} p_i(\underline{Z}_i)$ – тело запроса, представленное конечным числом конъюнктов. Переменные из вектора $\underline{X}$ называются *свободными (различимыми)* переменными, $\underline{Y}$ – *экзистенциальными (неразличимыми)* переменными. Конъюнкты p_i, называемые *предикатами*, могут иметь форму $C(a)$, где C – концепт в терминологии $\mathcal{T}$, либо $R(a, b)$, где R – роль в терминологии $\mathcal{T}$. Параметры предикатов могут быть переменными из $\underline{X}$ или $\underline{Y}$ либо константами ($\underline{Z}_i \subset \underline{X} \cup \underline{Y} \cup \mathbb{C}$, при этом $\underline{X} \subseteq \underline{Z}_1 \cup \ldots \cup \underline{Z}_k$). Конъюнктивные запросы над $\mathcal{L}$-терминологией будем обозначать $\mathcal{CQ}$-$\mathcal{L}$, запросы в форме конечного объединения (дизъюнкции) таких конъюнктивных запросов – $\mathcal{UCQ}$-$\mathcal{L}$. Рассмотренная форма запросов допускает указание в предикатах сложных концептов или ролей с помощью конструкторов дескриптивной логикой $\mathcal{L}$. Конъюнктивные запросы относительно пустой терминологии, допускающие только атомарные концепты или роли в конъюнктах, будем обозначать $\mathcal{CQ}$ (их объединения – соответственно $\mathcal{UCQ}$). Такие языки запросов будем называть реляционными. Реляционный язык конъюнктивных запросов, допускающий также конъюнкты в форме неравенств $Z_1 \neq Z_2$, будем обозначать $\mathcal{CQ}^{\neq}$, конечные объединения (дизъюнкции) таких запросов – $\mathcal{UCQ}^{\neq}$.

[4] Здесь и ниже подчеркнутый символ обозначает вектор переменных или констант (конечный упорядоченный набор переменных или констант).

Множеством ответов на $\mathcal{CQ}$-$\mathcal{L}$ запрос Q *относительно интерпретации* $\mathcal{I}$ называется множество векторов констант $\underline{t}$, такое, что при подстановке их вместо свободных переменных тела запроса, формула $\exists Y\, B(\underline{t}, \underline{Y})$ является истинной в $\mathcal{I}$:

$$Q(\mathcal{I}) \stackrel{\text{def}}{=} \{\underline{t} \mid \mathcal{I} \vDash \exists Y\, B(\underline{t}, \underline{Y}), \underline{t} = (c_1, \ldots, c_n), c_i \in \mathbb{C}\}.$$

Когда $\underline{t}$ является ответом на запрос Q относительно интерпретации $\mathcal{I}$ ($\underline{t} \in Q(\mathcal{I})$), будем также писать $\mathcal{I} \vDash Q(\underline{t})$.

Множеством ответов на $\mathcal{CQ}$-$\mathcal{L}$ запрос Q *относительно онтологии* $\mathcal{O} = \{\mathcal{T}, \mathcal{A}\}$ называется множество векторов констант $\underline{t}$, которые являются ответами на запрос Q относительно любой интерпретации $\mathcal{I}$, являющейся моделью онтологии $\mathcal{O}$:

$$Q(\mathcal{O}) \stackrel{\text{def}}{=} \{\underline{t} \mid \mathcal{I} \vDash Q(\underline{t})\ \forall \mathcal{I} \in \mathcal{M}_{\mathcal{O}}, \underline{t} = (c_1, \ldots, c_n), c_i \in \mathbb{C}\}.$$

Отметим, что приведенное определение существенно отличается от аналогичного определения для реляционной модели данных ввиду принципа «открытости» онтологий (open-world assumption). В отличие от реляционной базы данных, содержащей фиксированный набор данных, онтология может иметь множество моделей, каждая из которых представляет собой набор данных, совместимый с онтологией. Ответами же на запрос относительно онтологии считаются только такие факты, которые верны для всех совместимых с онтологией интерпретаций, т. е. которые неизбежно логически следуют из фактов и высказываний онтологии.

Математическая модель системы интеграции данных на основе онтологий

Формализуем математическую модель центральной системы интеграции данных на основе онтологий. Пусть задача заключается в исполнении согласованных распределенных запросов над m источниками данных $\{\Delta\}_{1..m}$, которые представлены онтологиями $\{\mathcal{O}_\Delta\}_{1..m}$.

Система интеграции данных на основе онтологий [5] формально определяется нами как

$$\Psi \stackrel{\text{def}}{=} \{\mathcal{O}_\Gamma, \{\mathcal{O}_\Delta\}_{1..m}, \{\mathcal{F}\}_{1..m}\},$$

где $\mathcal{O}_\Gamma = \{\mathcal{T}_\Gamma, \mathcal{A}_\Gamma\}$ – *глобальная онтология*, выраженная на языке дескриптивной логики $\mathcal{L}_\Gamma$ над алфавитом $\mathbb{N}_\Gamma$ (алфавит содержит символы, соответствующие всем концептам и ролям онтологии), будем считать, что онтология $\mathcal{O}_\Gamma$ выполнима ($\mathcal{M}_{\mathcal{O}\Gamma} \neq \varnothing$), а также имеет пустое множество исходных фактов ($\mathcal{A}_\Gamma = \varnothing$);

$\{\mathcal{O}_\Delta\}_{1..m}$ – конечное множество *онтологий источников данных* $\mathcal{O}_\Delta = \{\mathcal{T}_\Delta, \mathcal{A}_\Delta\}$, каждая из которых выражена на языке дескриптивной логики $\mathcal{L}_\Delta$ над собственным алфавитом $\mathbb{N}_\Delta$ (без ограничения общности принимаем все алфавиты взаимно непересекающимися), будем считать, что все онтологии $\mathcal{O}_\Delta$ выполнимы ($\mathcal{M}_{\mathcal{O}\Delta} \neq \varnothing$);

$\{\mathcal{F}\}_{1..m}$ – конечное множество *отображений* $\mathcal{F}$ между $\mathcal{O}_\Gamma$ и множеством онтологий источников $\{\mathcal{O}_\Delta\}$, задаваемых в общем случае в форме $q_\Delta \rightsquigarrow q_\Gamma$, где q_Γ и q_Δ – запросы на некотором языке $\mathcal{QL}_\mathcal{F}$, сформулированные в терминах $\mathcal{O}_\Gamma$ и $\{\mathcal{O}_\Delta\}$ соответственно, а знак $\rightsquigarrow$ обозначает один из операторов $\{\subseteq, \supseteq, \equiv\}$; отображения в форме $q_\Delta \subseteq q_\Gamma$ называются *корректными*, в форме $q_\Delta \supseteq q_\Gamma$ – *полными*, в форме $q_\Delta \equiv q_\Gamma$ *эквивалентными*.

[5] В дальнейшем систему интеграции данных на основе онтологий будем называть для краткости системой интеграции данных или просто Ψ.

Будем говорить, что отображение $q_\Delta \rightsquigarrow q_\Gamma$ *определяет глобальные* термины, если запрос q_Γ относительно глобальной онтологии имеет форму $A(a)$, где A – атомарный концепт в терминологии $\mathcal{T}_\Gamma$, либо $P(a, b)$, где P – атомарная роль в терминологии $\mathcal{T}_\Gamma$, a и b – свободные переменные (запросы такой формы будем называть *атомарными терминологическими*). Такое отображение сопоставляет отдельному атомарному концепту или роли глобальной онтологии $\mathcal{O}_\Gamma$ запрос в терминах онтологий источников $\{\mathcal{O}_\Delta\}$. Данный подход известен в литературе как Global-as-view, GAV [1; 2; 12], система интеграции данных, допускающая только отображения в форме GAV, называется GAV-системой.

Аналогично, будем говорить, что отображение $q_\Delta \rightsquigarrow q_\Gamma$ *определяет локальные* термины, если запрос q_Δ относительно онтологий источников является атомарным терминологическим, то есть имеет форму $A(a)$, где A – атомарный концепт в терминологии $\mathcal{T}_\Delta$, либо $P(a, b)$, где P – атомарная роль в терминологии $\mathcal{T}_\Delta$, a и b – свободные переменные. Такое отображение сопоставляет отдельному атомарному концепту или роли онтологии источника $\mathcal{O}_\Delta$ запрос в терминах глобальной онтологии $\mathcal{O}_\Gamma$. Данный подход известен в литературе как Local-as-view, LAV [1–2; 11], система интеграции данных, допускающая только отображения в форме LAV, называется LAV-системой. Если же в системе интеграции данных допускаются отображения произвольной формы $q_\Delta \rightsquigarrow q_\Gamma$, т. е. не ставится условие представления первого или второго запроса в атомарной терминологической форме, то такая система называется GLAV-системой (Global-local-as-view [1–2]).

Глобальной моделью для системы интеграции Ψ будем называть интерпретацию $\mathcal{I}_\Gamma$, которая является моделью для глобальной онтологии $\mathcal{O}_\Gamma$ ($\mathcal{I}_\Gamma \in \mathcal{M}_{\mathcal{O}\Gamma}$). Пусть $\mathcal{O}_U$ – онтология, полученная объединением онтологий источников, т. е. $\mathcal{O}_U \stackrel{\text{def}}{=} \{\bigcup_{\Delta\in\Psi} \perp_\Delta, \bigcup_{\Delta\in\Psi} \mathcal{A}_\Delta\}$. Будем говорить, что глобальная модель $\mathcal{I}_\Gamma$ *удовлетворяет корректному отображению* $\mathcal{F}$, заданному формулой $q_\Delta \subseteq q_\Gamma$, если $q_\Delta(\mathcal{O}_U) \subseteq q_\Gamma(\mathcal{I}_\Gamma)$, т. е. $\forall \underline{X}\,(\underline{X} \in q_\Delta(\mathcal{O}_U) \rightarrow \underline{X} \in q_\Gamma(\mathcal{I}_\Gamma))$. Аналогично, $\mathcal{I}_\Gamma$ *удовлетворяет полному отображению* $q_\Delta \supseteq q_\Gamma$, если $q_\Delta(\mathcal{O}_U) \supseteq q_\Gamma(\mathcal{I}_\Gamma)$, т. е. $\forall \underline{X}\,(\underline{X} \in q_\Gamma(\mathcal{I}_\Gamma) \rightarrow \underline{X} \in q_\Delta(\mathcal{O}_U))$. Наконец, $\mathcal{I}_\Gamma$ *удовлетворяет эквивалентному отображению* $q_\Delta \equiv q_\Gamma$, если $q_\Delta(\mathcal{O}_U) = q_\Gamma(\mathcal{I}_\Gamma)$, т. е. $\forall \underline{X}\,(\underline{X} \in q_\Delta(\mathcal{O}_U) \leftrightarrow \underline{X} \in q_\Gamma(\mathcal{I}_\Gamma))$. Будем говорить, что глобальная модель $\mathcal{I}_\Gamma$ *корректна* относительно Ψ, если $\mathcal{I}_\Gamma$ удовлетворяет всем отображениям $\{\mathcal{F}\}$ в Ψ.

Запросы к системе интеграции данных Ψ формулируются на некотором языке запросов $\mathcal{QL}_\Gamma$ в терминах глобальной онтологии $\mathcal{T}_\Gamma$, выраженной на языке дескриптивной логики $\mathcal{L}_\Gamma$. *Множеством точных ответов* на $\mathcal{QL}_\Gamma$ запрос Q *относительно системы интеграции данных* Ψ называется множество векторов констант $\underline{t}$, которые являются ответами на запрос Q относительно любой корректной глобальной модели $\mathcal{I}_\Gamma$ для Ψ:

$$Q(\Psi) \stackrel{\text{def}}{=} \{\underline{t} \mid \mathcal{I}_\Gamma \vDash Q(\underline{t})\ \forall \mathcal{I}_\Gamma \in \mathcal{M}_{\mathcal{O}\Gamma} : \mathcal{I}_\Gamma \text{ корректна относительно } \Psi\}.$$

Это определение по сути означает, что такие ответы логически следуют из фактов и высказываний онтологий источников, отображений, а также высказываний глобальной онтологии. Ответ является точным (является логическим следствием), если он имеет место в любой глобальной интерпретации, совместимой с высказываниями глобальной онтологии и отображениями ее в онтологии источников данных

В большинстве случаев вычисление множества точных ответов на основе логического вывода непосредственно над множествами фактов распределенных онтологий источников не-

приемлемо ввиду высокой сетевой нагрузки и низкой производительности, поэтому мы будем рассматривать смежную задачу, предполагающую предварительную переформулировку исходного запроса в отдельные запросы к источникам данных, которые могли бы быть исполнены непосредственно источниками. Результатом переформулировки запроса является так называемый *логический план* его распределенного исполнения – запрос, в котором используются исключительно термины онтологий источников. На основе логического плана запроса производится построение физического плана исполнения запроса, определяющего последовательность операций по извлечению данных из источников и их обработке. Вопрос построения физического плана исполнения запроса в данной статье не затрагивается.

Дадим формальные определения, связанные с переформулировкой запросов. Нам потребуются следующие определения содержимости и эквивалентности запросов. Будем говорить, что запрос Q_1 *содержится* в запросе Q_2 относительно системы интеграции данных на основе онтологий Ψ, и обозначать это $Q_1 \sqsubseteq_\Psi Q_2$, если для любых корректных глобальных моделей $\mathcal{I}_\Gamma$ множество ответов на запрос Q_1 содержится в множестве ответов на Q_2, т. е. $Q_1(\mathcal{I}_\Gamma) \subseteq Q_2(\mathcal{I}_\Gamma)$ $\forall \mathcal{I}_\Gamma \in \mathcal{M}_{\mathcal{O}\Gamma}$ такой, что $\mathcal{I}_\Gamma$ корректна относительно Ψ. Запросы Q_1 и Q_2 *эквивалентны* относительно Ψ ($Q_1 \equiv_\Psi Q_2$), если $Q_1 \sqsubseteq_\Psi Q_2$ и $Q_2 \sqsubseteq_\Psi Q_1$.

Теперь рассмотрим виды переформулировок в системе интеграции данных на основе онтологий. Пусть Q – запрос на языке $\mathcal{QL}_\Gamma$ в терминах глобальной онтологии $\mathcal{O}_\Gamma$. Запрос Q' на языке $\mathcal{QL}_R$ (над $\mathcal{L}_\Delta$-терминологией) называется *эквивалентной переформулировкой* запроса Q на основе системы интеграции данных Ψ, если:

1) все концепты или роли, используемые в запросе Q', являются терминами онтологий источников $\mathcal{O}_\Delta$ системы интеграции данных Ψ;

2) запрос Q' эквивалентен запросу Q относительно Ψ ($Q' \equiv_\Psi Q$).

Пусть Q – запрос на языке $\mathcal{QL}_\Gamma$ в терминах глобальной онтологии $\mathcal{O}_\Gamma$. Запрос Q' на языке $\mathcal{QL}_R$ (над $\mathcal{L}_\Delta$-терминологией) называется *максимальной переформулировкой* запроса Q на основе системы интеграции данных Ψ относительно языка запросов $\mathcal{QL}_R$, если:

1) все концепты или роли, используемые в запросе Q', являются терминами онтологий источников $\{\mathcal{O}_\Delta\}$ системы интеграции данных Ψ;

2) запрос Q' содержится в запросе Q относительно Ψ ($Q' \sqsubseteq_\Psi Q$);

3) не существует такого запроса P на языке $\mathcal{QL}_R$ (над $\mathcal{L}_\Delta$-терминологией), в котором все используемые концепты или роли являются терминами онтологий источников $\{\mathcal{O}_\Delta\}$ системы интеграции данных Ψ такого, что $Q' \sqsubseteq_\Psi P$ и $P \sqsubseteq_\Psi Q$, но Q' не эквивалентен запросу P.

Если запрос Q' удовлетворяет первым двум условиям, но не является максимальной переформулировкой, будем называть его просто переформулировкой. Переформулировку Q' запроса Q на основе системы интеграции данных Ψ будем называть *точной* относительно системы интеграции данных Ψ, если множество ответов на запрос Q' относительно объединения всех онтологий источников $\mathcal{O}_U=\{\cup_{\Delta\in\Psi} \mathcal{T}_\Delta, \cup_{\Delta\in\Psi} \mathcal{A}_\Delta\}$ совпадает с множеством точных ответов на запрос Q относительно системы интеграции данных Ψ: $Q'(\mathcal{O}_U) = Q(\Psi)$.

Следует отметить, что одним из ключевых параметров задачи построения переформулировки является целевой язык переформулированного запроса, этот параметр непосредственно влияет на разрешимость задачи. Так, требуемая переформулировка может не существовать для некоторого ограниченного целевого языка и при этом существовать для более выразительного целевого языка. Отметим, что, в принципе, точная переформулировка запроса всегда возможна при условии, что на язык переформулированного запроса $\mathcal{QL}_R$ не накла-

дывается никаких ограничений и что задача ответа на исходный запрос относительно отдельной онтологии для используемого диалекта дескриптивной логики в принципе разрешима. Действительно, если в $\mathcal{QL}_R$ может быть заложено произвольное вычисление машины Тьюринга, то такая машина может полностью произвести логический вывод на основе исходного запроса, терминологий и множеств фактов всех онтологий источников и выдать в качестве результата множество точных ответов на исходный запрос.

Анализ разрешимости задачи по переформулировке запросов

Рассмотрим условия, при которых возможна переформулировка запросов в системе интеграции данных на основе онтологий. Для того чтобы выявить случаи, в которых точная переформулировка исходного запроса невозможна, рассмотрим леммы, связывающие сложность вычисления множества ответов на запросы (далее также будем называть ее сложностью ответа на запросы) с фактом существования точной переформулировки.

Лемма 1. Пусть система интеграции данных Ψ такова, что в ней всегда возможна точная переформулировка запроса Q на языке $\mathcal{QL}_\Gamma$ над $\mathcal{L}_\Gamma$-терминологией в запрос Q' на языке $\mathcal{QL}_R$ над $\mathcal{L}_\Delta$-терминологией. В таком случае сложность ответа на $\mathcal{QL}_R$-запросы над $\mathcal{L}_\Delta$-онтологиями относительно объема фактов онтологии $|\mathcal{A}|$ не ниже сложности вычисления множества точных ответов на $\mathcal{QL}_\Gamma$-запросы для системы интеграции данных Ψ относительно суммарного объема фактов всех источников в системе $|\cup_{\Delta\in\Psi} \mathcal{A}_\Delta|$.

Лемма 2. Сложность вычисления множества точных ответов на запросы на языке $\mathcal{QL}_\Gamma$ над $\mathcal{L}_\Gamma$-терминологией в системе интеграции данных Ψ, выраженная относительно суммарного объема фактов всех источников в системе $|\cup_{\Delta\in\Psi} \mathcal{A}_\Delta|$, не ниже сложности ответа на запросы для языка $\mathcal{QL}_\Gamma$ над $\mathcal{L}_\Gamma$-онтологиями относительно объема фактов онтологии $|\mathcal{A}|$.

Лемма 3. Сложность ответа на конъюнктивные запросы $\mathcal{CQ}$-$\mathcal{L}$ над онтологиями на языке дескриптивной логики $\mathcal{L}$, выраженная относительно объема фактов онтологии $|\mathcal{A}|$ или объема терминологии $|\mathcal{T}|$, не ниже сложности проверки принадлежности объекта к концепту, а также сложности проверки содержимости одного концепта в другом для дескриптивной логики $\mathcal{L}$ относительно $|\mathcal{A}|$ или $|\mathcal{T}|$ соответственно.

Таким образом, для возможности точной переформулировки, сложность ответа на запросы для языка $\mathcal{QL}_R$ переформулированных запросов над дескриптивной логикой $\mathcal{L}_\Delta$ должна быть не ниже аналогичной сложности вычисления точных ответов на $\mathcal{QL}_\Gamma$-запросы в системе интеграции Ψ, которая в свою очередь ограничена снизу сложностью ответа на $\mathcal{QL}_\Gamma$-запросы над дескриптивной логикой $\mathcal{L}_\Gamma$.

Эти простые леммы позволяют на основе анализа сложности ответов на запросы сделать выводы относительно невозможности точной переформулировки запросов для достаточно выразительных диалектов дескриптивной логики на недостаточно выразительных языках запросов, в том числе таких, которые могли бы быть напрямую транслированы в SQL.

Пусть $\mathcal{QL}_\Gamma = \mathcal{CQ}$-$\mathcal{L}_\Gamma$, т. е. пользовательские запросы к системе интеграции данных на основе онтологий Ψ формулируются только в виде простых конъюнктивных запросов над дескриптивной логикой $\mathcal{L}_\Gamma$ (математический эквивалент введенного выше класса запросов

SPARQL-BGP). В реляционных LAV-системах интеграции данных с корректными конъюнктивными отображениями схем всегда возможна максимальная переформулировка конъюнктивного запроса ($\mathcal{CQ}$) в конечное объединение конъюнктивных запросов ($\mathcal{UCQ}$) в терминах источников данных [8; 10]. Рассмотрим, для каких $\mathcal{L}_\Gamma$ в системе интеграции данных на основе онтологий Ψ возможна переформулировка $\mathcal{CQ}$-$\mathcal{L}_\Gamma$ запроса в реляционный $\mathcal{UCQ}$-запрос или хотя бы в произвольную формулу реляционного исчисления.

Как известно, сложность вычисления для некоторого набора данных ответов на запрос, заданный формулой реляционного исчисления (в том числе $\mathcal{UCQ}$), относительно объема данных, лежит в классе $\mathcal{LOGSPACE}$, т. е. ограничивается логарифмическим объемом памяти относительно объема данных (см. [24]). Непосредственным следствием этого утверждения и приведенных лемм 1 и 2 является следующая теорема

Теорема 1. Пусть дана система интеграции данных на основе онтологий $\Psi = \{\mathcal{O}_\Gamma, \{\mathcal{O}_\Delta\}, \{\mathcal{F}\}\}$, где $\mathcal{O}_\Gamma$ выражена на диалекте дескриптивной логики $\mathcal{L}_\Gamma$. В системе Ψ невозможна точная переформулировка $\mathcal{CQ}$-$\mathcal{L}_\Gamma$ запроса в формулу реляционного исчисления (в том числе конечное объединение конъюнктивных запросов $\mathcal{UCQ}$), если диалект дескриптивной логики $\mathcal{L}_\Gamma$ таков, что сложность ответа на $\mathcal{CQ}$-$\mathcal{L}_\Gamma$ запросы для онтологий на языке дескриптивной логики $\mathcal{L}_\Gamma$ относительно объема фактов онтологии $|\mathcal{A}|$ лежит вне класса $\mathcal{LOGSPACE}$.

В табл. 2 приведена информация о рассмотренных в литературе характеристиках вычислительной сложности типовых задач для ряда распространенных диалектов дескриптивной логики [24–27]. Для различных диалектов дескриптивной логики $\mathcal{L}$ приведена вычислительная сложность ответа на конъюнктивные запросы ($\mathcal{CQ}$-$\mathcal{L}$), проверки принадлежности объекта к концепту ($a \in C$), проверки содержимости одного концепта в другом ($C_1 \sqsubseteq_\mathcal{O} C_2$), а также проверки выполнимости онтологии и выполнимости концепта. В столбцах таблицы приводятся характеристики вычислительной сложности от объема множества фактов онтологии $|\mathcal{A}|$, терминологии $|\mathcal{T}|$, размера запроса $|Q|$ (для $\mathcal{CQ}$-$\mathcal{L}$) и от $|\mathcal{A}| + |\mathcal{T}| + |Q|$ в совокупности.

Таблица 2

Сопоставление вычислительной сложности задач для разных диалектов дескриптивной логики *

Диалект $\mathcal{L}$	Сложность от объема данных $\|\mathcal{A}\|$	Сложность от размера терминологии $\|\mathcal{T}\|$	Сложность от размера запроса $\|Q\|$	Совокупная сложность
Сложность ответа на конъюнктивный запрос ($\mathcal{CQ}$-$\mathcal{L}$)				
RDF Schema [20]	$\mathcal{LOGSPACE}$	$\in \mathcal{P}$	Неизвестна	Неизвестна
$\mathcal{DL}$-$Lite^A$ [18; 21–22]	$\mathcal{LOGSPACE}$	$\in \mathcal{P}$	$\mathcal{NP}$-полная	$\mathcal{NP}$-полная
$\mathcal{EL}$++ [28]	$\mathcal{P}$-сложная	Неизвестна	Неизвестна	Неизвестна
DLP [29]	$\mathcal{P}$-полная	$\in \mathcal{EXPTIME}$	$\in \mathcal{EXPTIME}$	$\in \mathcal{EXPTIME}$

Продолжение табл. 2

Диалект $\mathcal{L}$	Сложность от объема данных $\|\mathcal{A}\|$	Сложность от размера терминологии $\|\mathcal{T}\|$	Сложность от размера запроса $\|Q\|$	Совокупная сложность
Horn-$\mathcal{SHIQ}$ [30]	$\in \mathcal{P}$	Неизвестна	Неизвестна	Неизвестна
$\mathcal{SHIF}^{(\mathcal{D})}$ (OWL-Lite)	*co*-$\mathcal{NP}$-полная	$\mathcal{EXPTIME}$-полная	$\in 2\mathcal{EXPTIME}$	$\in 2\mathcal{EXPTIME}$
$\mathcal{SHOIN}^{(\mathcal{D})}$ (OWL-DL)	Разрешимость неизвестна	Разрешимость неизвестна	Разрешимость неизвестна	Разрешимость неизвестна
Сложность проверки принадлежности объекта к концепту ($a \in C$)				
RDF Schema	$\in \mathcal{LOGSPACE}$	$\in \mathcal{P}$	–	$\in \mathcal{P}$
$\mathcal{DL\text{-}Lite}^{\mathcal{A}}$	$\in \mathcal{LOGSPACE}$	$\in \mathcal{P}$	–	$\in \mathcal{P}$
$\mathcal{EL}$++	$\mathcal{P}$-полная	$\mathcal{P}$-полная	–	$\mathcal{P}$-полная
Horn-$\mathcal{SHIQ}$	$\mathcal{P}$-полная	$\mathcal{EXPTIME}$-полная	–	$\mathcal{EXPTIME}$-полная
$\mathcal{AL}$, $\mathcal{ALN}$	–	$\mathcal{P}$	–	$\mathcal{P}$
$\mathcal{ALE}$	–	$\mathcal{NP}$-сложная, *co*-$\mathcal{NP}$-сложная	–	$\mathcal{NP}$-сложная, *co*-$\mathcal{NP}$-сложная
$\mathcal{ALR}$	–	$\mathcal{NP}$	–	
$\mathcal{ALU}$	–	*co*-$\mathcal{NP}$	–	
$\mathcal{ALC}$	–	$\mathcal{PSPACE}$	–	$\mathcal{PSPACE}$
$\mathcal{SHIF}^{(\mathcal{D})}$ (OWL-Lite)	*co*-$\mathcal{NP}$-полная	$\mathcal{EXPTIME}$-полная	–	$\mathcal{EXPTIME}$-полная
$\mathcal{SHOIN}^{(\mathcal{D})}$ (OWL-DL)	$\mathcal{NP}$-сложная	$\mathcal{NEXPTIME}$-полная	–	$\mathcal{NEXPTIME}$-полная
Сложность проверки содержимости одного концепта в другом ($C_1 \sqsubseteq_{\mathcal{O}} C_2$)				
RDF Schema	$\mathcal{LOGSPACE}$	$\in \mathcal{P}$	–	$\in \mathcal{P}$
$\mathcal{DL\text{-}Lite}^{\mathcal{A}}$	$\mathcal{LOGSPACE}$	$\in \mathcal{P}$	–	$\in \mathcal{P}$
$\mathcal{EL}$++	$\mathcal{P}$-полная	$\mathcal{P}$-полная	–	$\mathcal{P}$-полная
DLP	$\mathcal{P}$-полная	$\in \mathcal{EXPTIME}$	–	$\in \mathcal{EXPTIME}$
Horn-$\mathcal{SHIQ}$	$\mathcal{P}$-полная	$\mathcal{EXPTIME}$-полная	–	$\mathcal{EXPTIME}$-полная
$\mathcal{AL}$, $\mathcal{ALN}$	–	$\mathcal{P}$	–	$\mathcal{P}$
$\mathcal{ALE}$	–	$\mathcal{NP}$	–	$\mathcal{NP}$
$\mathcal{ALR}$	–	$\mathcal{NP}$	–	$\mathcal{NP}$

Окончание табл. 2

Диалект $\mathcal{L}$	Сложность от объема данных $\|\mathcal{A}\|$	Сложность от размера терминологии $\|\mathcal{T}\|$	Сложность от размера запроса $\|Q\|$	Совокупная сложность
$\mathcal{ALU}$	–	*co-*$\mathcal{NP}$	–	*co-*$\mathcal{NP}$
$\mathcal{ALC}$	–	$\mathcal{PSPACE}$	–	$\mathcal{PSPACE}$
$\mathcal{SHIF}^{(\mathcal{D})}$ (OWL-Lite)	*co-*$\mathcal{NP}$-полная	$\mathcal{EXPTIME}$-полная	–	$\mathcal{EXPTIME}$-полная
$\mathcal{SHOIN}^{(\mathcal{D})}$ (OWL-DL)	$\mathcal{NP}$-сложная	$\mathcal{NEXPTIME}$-полная	–	$\mathcal{NEXPTIME}$-полная
Сложность проверки выполнимости онтологии, выполнимости концепта				
RDF Schema	Тривиально	Тривиально	–	Тривиально
$\mathcal{DL\text{-}Lite}^{\mathcal{A}}$	$\in \mathcal{LOGSPACE}$	$\in \mathcal{P}$	–	$\in \mathcal{P}$
$\mathcal{EL}$++	$\mathcal{P}$-полная	$\mathcal{P}$-полная	–	$\mathcal{P}$-полная
DLP	$\mathcal{P}$-полная	$\in \mathcal{EXPTIME}$	–	$\in \mathcal{EXPTIME}$
Horn-$\mathcal{SHIQ}$	$\mathcal{P}$-полная	$\mathcal{EXPTIME}$-полная	–	$\mathcal{EXPTIME}$-полная
$\mathcal{AL}$, $\mathcal{ALN}$	–	$\mathcal{P}$	–	$\mathcal{P}$
$\mathcal{ALE}$	–	*co-*$\mathcal{NP}$	–	*co-*$\mathcal{NP}$
$\mathcal{ALR}$	–	*co-*$\mathcal{NP}$	–	*co-*$\mathcal{NP}$
$\mathcal{ALU}$	–	$\mathcal{NP}$	–	$\mathcal{NP}$
$\mathcal{ALC}$	–	$\mathcal{PSPACE}$	–	$\mathcal{PSPACE}$
$\mathcal{SHIF}^{(\mathcal{D})}$ (OWL-Lite)	$\mathcal{NP}$-полная	$\mathcal{EXPTIME}$-полная		$\mathcal{EXPTIME}$-полная
$\mathcal{SHOIN}^{(\mathcal{D})}$ (OWL-DL)	$\mathcal{NP}$-сложная	$\mathcal{NEXPTIME}$-полная		$\mathcal{NEXPTIME}$-полная

* В таблице используются следующие обозначения классов сложности: $\mathcal{LOGSPACE}$ – разрешимо детерминированной машиной Тьюринга (ДМТ) с логарифмическим объемом памяти; $\mathcal{P}$ – разрешимо ДМТ за полиномиальное время; $\mathcal{NP}$ – разрешимо за полиномиальное время недетерминированной машиной Тьюринга (НДМТ); *co-*$\mathcal{NP}$ – дополнение задачи лежит в классе $\mathcal{NP}$ (например, дополнением задачи содержимости концепта является несодержимость концепта); $\mathcal{PSPACE}$ – разрешимо ДМТ с полиномиальным объемом памяти; $\mathcal{EXPTIME}$ – разрешимо ДМТ за экспоненциальное время; $\mathcal{NEXPTIME}$ – разрешимо НДМТ за экспоненциальное время; $2\mathcal{EXPTIME}$ – разрешимо ДМТ за время $_2 2^{p(n)}$. Известно включение $\mathcal{LOGSPACE} \subseteq \mathcal{P} \subseteq \mathcal{NP} \subseteq \mathcal{PSPACE} \subseteq \mathcal{EXPTIME} \subseteq$ $\subseteq \mathcal{NEXPTIME} \subseteq \in\mathcal{EXPTIME}$. Полной в классе называется задача, к которой полиномиально сводимы ДМТ все задачи класса. Сложной в классе называется задача, к которой полиномиально сводима ДМТ некоторая полная задача в классе.

Из теоремы 1 и приведенных характеристик следует, что даже диалект OWL-Lite (дескриптивная логика $\mathcal{SHIF}^{(\mathcal{D})}$) обладает слишком высокими выразительными способностями для того, чтобы простейшие конъюнктивные запросы относительно OWL-Lite онтологии всегда можно было переформулировать в реляционное исчисление, т. е. в SQL (точнее, в то подмножество SQL, которое соответствует реляционному исчислению). Это означает невозможность в полной мере задействовать оптимизированные механизмы исполнения запросов современных реляционных СУБД в том случае, когда интегрируемые системой источники данных представлены, полностью или частично, реляционными базами данных (т. е. когда множества фактов $\mathcal{A}_\Delta$ онтологий источников хранятся в реляционных базах данных). В то же время возможность задействовать SQL для вычисления переформулированных запросов весьма принципиальна с практической точки зрения, поскольку это позволяет эффективно работать с большими объемами данных.

Переформулировка запросов для дескриптивной логики $\mathcal{DL}$-$\mathcal{L}$ite

Наиболее выразительные языки дескриптивной логики, для которых сложность ответа на конъюнктивные запросы относительно объема данных все еще лежит в классе $\mathcal{LOGSPACE}$, принадлежат к семейству $\mathcal{DL}$-$\mathcal{L}ite$ [18; 21–22]. Детальный анализ показывает, что для языка $\mathcal{DL}$-$\mathcal{L}ite^R$ (см. определение выше) действительно можно предложить корректный алгоритм переформулировки конъюнктивных запросов в реляционное исчисление (конкретнее, $\mathcal{UCQ}$), и имеет место следующая теорема.

Теорема 2. Пусть система интеграции данных $\Psi = \{\mathcal{O}_\Gamma, \{\mathcal{O}_\Delta\}, \{\mathcal{F}\}\}$ такова, что:

• глобальная онтология $\mathcal{O}_\Gamma=\{\mathcal{T}_\Gamma, \mathcal{A}_\Gamma\}$ выполнима и выражена на языке дескриптивной логики $\mathcal{L}_\Gamma = \mathcal{DL}$-$\mathcal{L}ite^R$, причем $\mathcal{A}_\Gamma = \varnothing$;

• онтологии источников данных $\mathcal{O}_\Delta=\{\mathcal{T}_\Delta, \mathcal{A}_\Delta\}$ выполнимы и выражены на языке дескриптивной логики $\mathcal{L}_\Delta = \mathcal{DL}$-$\mathcal{L}ite^R$;

• отображения $\mathcal{F}$ заданы в форме $q_\Delta \subseteq q_\Gamma$, где q_Δ и q_Γ – конъюнктивные запросы над $\mathcal{DL}$-$\mathcal{L}ite^R$-терминологиями $\{\mathcal{T}_\Delta\}$ и $\mathcal{T}_\Gamma$ соответственно, т. е. рассматривается GLAV-система с корректными $\mathcal{CQ}$-$\mathcal{DL}$-$\mathcal{L}ite^R$ отображениями.

Пусть пользовательский запрос Q к системе Ψ задается в форме объединения конъюнктивных запросов $\mathcal{QL}_\Gamma = \mathcal{UCQ}$-$\mathcal{DL}$-$\mathcal{L}ite^R$ над терминологией $\mathcal{T}_\Gamma$. Тогда всегда существует максимальная переформулировка запроса Q на основе системы интеграции данных Ψ, представимая в виде объединения реляционных конъюнктивных запросов над терминологиями $\{\mathcal{T}_\Delta\}$ (на языке $\mathcal{QL}_R = \mathcal{UCQ}$).

Вкратце рассмотрим основные принципы предлагаемого алгоритма [6] построения переформулированного запроса для рассматриваемого в теореме 2 класса систем интеграции данных. Предварительным условием является *нормализация* $\mathcal{DL}$-$\mathcal{L}ite^R$ терминологии $\mathcal{T}_\Gamma$ исчерпывающим применением к аксиомам следующих правил нормализации (обозначения даны выше):

[6] Подробная спецификация алгоритма переформулировки и исследование характеристик и сложности алгоритма приводится в диссертационной работе автора.

• $C_L \sqsubseteq C_1 \sqcap C_2 \Rightarrow C_L \sqsubseteq C_1, C_L \sqsubseteq C_2$;

• $C_L \sqsubseteq \exists R_a.C \Rightarrow C_L \sqsubseteq \exists R_\pi, R_\pi \sqsubseteq R_a, \exists R_\pi^- \sqsubseteq C$, где R_π – новая абстрактная роль;

• $\exists R_1 \sqsubseteq \exists R_2 \Rightarrow \exists R_1 \sqsubseteq A_\pi, A_\pi \sqsubseteq \exists R_2$, где A_π – новый атомарный концепт;

• $\exists R \sqsubseteq \neg A \Rightarrow \exists R_1 \sqsubseteq A_\pi, A_\pi \sqsubseteq \neg A$, где A_π – новый атомарный концепт;

• $A \sqsubseteq \neg\exists R \Rightarrow A \sqsubseteq \neg A_\pi, \exists R \sqsubseteq A_\pi$, где A_π – новый атомарный концепт;

• $\exists R_1 \sqsubseteq \neg\exists R_2 \Rightarrow \exists R_1 \sqsubseteq A_\pi, A_\pi \sqsubseteq \neg B_\pi, \exists R_2 \sqsubseteq B_\pi$, где A_π, B_π – новые атомарные концепты;

• $A \sqsubseteq \exists R_a^- \Rightarrow A \sqsubseteq \exists R_\pi, R_\pi \sqsubseteq R_a^-$, где R_π – новая абстрактная роль;

• $R_{a1}^- \sqsubseteq R_{a2}^- \Rightarrow R_{a1} \sqsubseteq R_{a2}$;

• $R_{a1}^- \sqsubseteq \neg R_{a2}^- \Rightarrow R_{a1} \sqsubseteq \neg R_{a2}$;

• $R_{a1} \sqsubseteq R_{a2}^- \Rightarrow R_{a1}^- \sqsubseteq R_{a2}$;

• $R_{a1} \sqsubseteq \neg R_{a2}^- \Rightarrow R_{a1}^- \sqsubseteq \neg R_{a2}$;

• $R_{a1}^- \sqsubseteq \neg R_{a2} \Rightarrow R_{a1}^- \sqsubseteq R_\pi, R_\pi \sqsubseteq \neg R_{a2}$, где R_π – новая абстрактная роль.

После нормализации $\mathcal{DL}$-$\mathcal{L}ite^R$ терминология $\mathcal{T}_\Gamma$ будет содержать только аксиомы в форме:

• $A_1 \sqsubseteq \neg A_2, R_{a1} \sqsubseteq \neg R_{a2}, R_{d1} \sqsubseteq \neg R_{d2}$ – непересекаемость атомарных концептов, ролей;

• $A_1 \sqsubseteq A_2, R_{a1} \sqsubseteq R_{a2}, R_{d1} \sqsubseteq R_{d2}$ – иерархия вложения атомарных концептов, ролей;

• $R_{a1}^- \sqsubseteq R_{a2}$ – определение обратной роли;

• $\exists R \sqsubseteq A$ – ограничение домена роли (определение концепта субъектов роли);

• $\exists R_a^- \sqsubseteq A$ – ограничение значений роли (определение концепта объектов роли);

• $\rho(R_d) \sqsubseteq d$ – указание типа данных атрибута;

• $A \sqsubseteq \exists R$ – экзистенциальное ограничение.

Перейдем к рассмотрению этапов алгоритма переформулировки. На первом этапе алгоритма производится построение промежуточной переформулировки Q' пользовательского запроса Q с учетом ограничений целостности глобальной онтологии, т. е. аксиом $\mathcal{T}_\Gamma$. Основная идея этапа заключается в том, чтобы «закодировать» в запрос необходимые аксиомы терминологии $\mathcal{T}_\Gamma$. Исходно ответ на запрос относительно онтологии предполагает предварительное поведение логического вывода на основе фактов онтологии и ее терминологических аксиом, в результате которого вычисляются производные факты. Предлагаемый алгоритм позволяет исключить необходимость проведения такого логического вывода, вместо этого переформулировав запрос таким образом, чтобы он учитывал все необходимые производные факты. По сути построение такого переформулированного запроса является своего рода логическим выводом относительно исходного запроса и аксиом терминологии.

Краткий листинг предлагаемого алгоритма переформулировки запроса относительно $\mathcal{DL}$-$\mathcal{L}ite^R$ терминологии приводится ниже. Алгоритм производит построение альтернативных формулировок запроса на основе аксиом терминологии исчерпывающим применением к запросу правил замены предикатов, приводимых в табл. 3. Помимо расширения запроса альтернативными формулировками алгоритм производит отсеивание пустых подзапросов, согласно аксиомам непересекаемости, и минимизацию запросов.

На следующем этапе вычисляется переформулировка Q'' полученного запроса Q' относительно правил отображения $q_\Delta \subseteq q_\Gamma$. Для этой цели может быть использован модифицированный Bucket-алгоритм или алгоритм обратных правил (см. [8]), используемые в реляционных системах интеграции данных. На третьем этапе алгоритма производится, при необходимости, переформулировка запроса Q'' относительно терминологических аксиом $\mathcal{T}_\Delta$ онтологий источников, задействованных в запросе. Эта операция аналогична этапу 1. На последнем этапе производится исключение избыточных операций и оптимизация запроса.

Таблица 3

Правила переформулировки запроса относительно $\mathcal{DL}$-$\mathcal{Lite}^R$ терминологии

Исходный предикат запроса	Аксиома терминологии	Заменяющий предикат
$A(x)$	$A_2 \sqsubseteq A$	$A_2(x)$
$R(x, y)$	$R_2 \sqsubseteq R$	$R_2(x, y)$
$R(x, y)$	$R_2^- \sqsubseteq R$	$R_2(y, x)$
$A(x)$	$\exists R \sqsubseteq A$	$R(x, y_\pi)$, где y_π – новая экзистенциальная переменная запроса
$A(x)$	$\exists R^- \sqsubseteq A$	$R(y_\pi, x)$, где y_π – новая экзистенциальная переменная запроса
$R(x, y)$, где y – экзистенциальная переменная	$A \sqsubseteq \exists R$	$A(x)$

Алгоритм 1. Переформулировка запроса относительно $\mathcal{DL}$-$\mathcal{Lite}^R$ терминологии

Входные параметры: Q – запрос на языке $\mathcal{UCQ}$-$\mathcal{DL}$-$\mathcal{Lite}^R$

$\mathcal{T}$ – терминология на языке $\mathcal{DL}$-$\mathcal{Lite}^R$

Результат: Q' – запрос на языке $\mathcal{UCQ}$-$\mathcal{DL}$-$\mathcal{Lite}^R$

‣ $Q' = Q$
Цикл { *// цикл поиска возможных переформулировок*
  $Q_{врем} := Q'$
  Цикл ∀ дизъюнкта q в $Q_{врем}$ {
    // переформулировка с учетом аксиом терминологии:
    Цикл ∀ конъюнкта p в q {
      Цикл ∀ аксиомы T терминологии $\mathcal{T}$ {

        Если для p и T есть правило замены на p' (см. табл. 3){
          $q' := (q - p) \wedge p'$ *// заменить p на p' согласно правилу*
          $Q' := Q' \vee q'$ */ добавить в Q' переписанный дизъюнкт*
        }
      }
    } *// конец цикла по конъюнктам*
    Цикл ∀ пары конъюнктов одинаковой «арности» p_1, p_2 в q {
    // учет аксиом непересекаемости:
      Если параметры предикатов p_1 и p_2 идентичны (одинаковые переменные и константы) {
    Цикл ∀ аксиомы непересекаемости терминологии $\mathcal{T}$ в форме $t_1 \sqsubseteq \neg t_2$, где $t_1, t_2 \in \{A, R_a, R_d\}$ {
      Если t_1 сводимо к p_1, а t_2 к p_2 (либо t_1 к p_2, t_2 к p_1) путем рекурсивного применения правил замены по табл. 3
        $Q' := Q' - q$ *// убрать невыполнимый дизъюнкт*
      }
    }
    // минимизация запросов:

Если p_1, p_2 представлены одним и тем же концептом или ролью, причем приравниванием переменных может быть достигнута полная идентичность конъюнктов p_1 и p_2 {
$q' := q - p_2$ *// убрать дублирующий предикат* p_2
q' := ren_vars(q') *// переобозначить в* q' *переменные с учетом их приравнивания для идентичности* p_1 *и* p_2
$Q' := Q' \vee q'$ *// добавить в* Q' *переписанный дизъюнкт*
}
} *// конец цикла по конъюнктам*
} *// конец цикла по дизъюнктам*
Если $Q' = Q_{врем}$, прервать цикл *// нет переформулировок*
} *// конец цикла поиска переформулировок*
Вернуть Q' ◂

В результате выполнения всех этапов алгоритма вычисляется максимальная переформулировка исходного $\mathcal{UCQ}$-$\mathcal{DL}$-$\mathcal{L}ite^R$ запроса к системе интеграции Ψ, представляющая собой объединение конъюнктивных запросов ($\mathcal{UCQ}$) к информационным источникам, в которых используются исключительно термины онтологий источников. На основе переформулированного запроса производится построение физического плана исполнения запроса, определяющего последовательность операций по извлечению данных из источников и их обработке. Вопрос построения физического плана исполнения запроса в данной статье не затрагивается, здесь может быть непосредственно применена методика, используемая в реляционных системах интеграции данных.

Смежные работы

Задача ответа на запросы в системе интеграции данных, в том числе смежная задача ответа на запросы с использованием представлений (вариант для LAV-систем интеграции), в той или иной степени рассмотрена в литературе для различных моделей данных, включая реляционную модель данных, объектную модель данных, слабоструктурированные данные, XML, а также некоторые диалекты дескриптивной логики. При этом для разных моделей данных, разных языков запросов и разных механизмов отображения схем (спецификации представлений) для решения задачи используются существенно различные алгоритмы, и имеют место различные условия разрешимости задачи.

Наибольшее внимание рассматриваемая проблема получила в контексте реляционных баз данных, прежде всего в связи тем, что задача ответа на запросы с использованием представлений (LAV) непосредственно применима также к оптимизации запросов в реляционной СУБД с материализованными представлениями. В этом случае рассматривается возможность переформулировки исходного пользовательского запроса таким образом, чтобы снизить время его исполнения за счет использования предварительно подготовленных материализованных представлений.

Обзор основных работ, посвященных задаче ответа на запросы с использованием представлений (и переформулировки запросов с использованием представлений) в *реляционной* модели данных приведен в [8]. В этом обзоре рассматриваются также три алгоритма (Bucket-алгоритм, алгоритм обратных правил и алгоритм MiniCon), предназначенных для переформулировки конъюнктивных запросов (в том числе с арифметическими сравнениями) к отношениям глобальной реляционной схемы в объединение конъюнктивных запросов ($\mathcal{UCQ}$) к реляционным схемам источников данных, для конъюнктивных LAV-отображений. Эти алгоритмы были разработаны для первых систем интеграции данных: Bucket-алгоритм был реализован в системе Information Manifold [11], а алгоритм обратных правил – в системе

InfoMaster. В обзоре [8] приводится также перечень работ, рассматривающих расширение стандартной реляционной задачи переформулировки запросов с использованием представлений, в частности агрегацией в запросах или отображениях, ограничениями целостности (см. также [31; 32]), ограничениями доступа и пр. Кроме того, приводятся ссылки на результаты для других моделей данных, в том числе для слабоструктурированных данных.

В работе [10] приводятся характеристики вычислительной сложности ответа на запросы с использованием представлений в реляционной модели данных. Рассматриваются корректные ($r_\Delta \subseteq q_\Gamma$) и эквивалентные ($r_\Delta \equiv q_\Gamma$) LAV-отображения, задаваемые конъюнктивными запросами $\mathcal{CQ}$ (в том числе с неравенствами $\mathcal{CQ}^{\neq}$), их объединениями ($\mathcal{UCQ}$), рекурсивными Datalog-запросами или произвольными формулами реляционного исчисления. Аналогично рассматриваются различные формы пользовательских запросов к системе. Среди основных выводов: задача в принципе неразрешима, когда запросы либо отображения представлены произвольными формулами реляционного исчисления, а также неразрешима в ряде случаев при использовании рекурсивных представлений. Кроме того, задача трудноразрешима (*co*-$\mathcal{NP}$) за исключением случая корректных отображений, задаваемых конъюнктивными запросами $\mathcal{CQ}$ (в том числе $\mathcal{CQ}^{\neq}$), при формулировке пользовательских запросов на языках $\mathcal{CQ}$, $\mathcal{UCQ}$ или Datalog. Отсюда следует, в частности, что класс полиномиально разрешимых задач ответа на запросы даже в реляционных LAV-системах интеграции данных крайне ограничен.

Первая попытка рассмотреть задачу переформулировки запросов относительно LAV-системы интеграции данных в контексте *дескриптивной логики* была представлена в работе [13] (1997). В работе показывается, что максимальная переформулировка *терминологических* (задаваемых описанием концепта или ролью) запросов для дескриптивной логики $\mathcal{ALN}$ в $\mathcal{UCQ}$ относительно корректных $\mathcal{CQ}$-$\mathcal{ALN}$ отображений возможна только при наложении жестких ограничений на форму отображений (отсутствие экзистенциальных переменных), что неприемлемо на практике. Помимо этого, в [13] доказывается разрешимость более простой задачи, в которой пользовательские запросы, отображения и переформулировки задаются терминологическими запросами в дескриптивной логике $\mathcal{ALCNR}$ (т. е. описанием концепта или ролью на языке $\mathcal{ALCNR}$).

Задача чисто терминологической переформулировки для $\mathcal{ALN}$ и $\mathcal{ALE}$ также детально рассмотрена в работе [33]. В работах [14–16] рассматривается переформулировка запросов относительно корректных терминологических отображений.

В [14] рассматривается GAV-система интеграции данных с ограничениями целостности, в которой пользовательские запросы формулируются на языке $\mathcal{UCQ}$-$\mathcal{ALN}$, а отображения задаются $\mathcal{ALN}$-терминами (терминологическими запросами на языке $\mathcal{ALN}$). Жесткое ограничение выразительных возможностей отображений принимается в связи с доказанной в [13] невозможностью максимальной переформулировки запросов для конъюнктивных отображений $\mathcal{CQ}$-$\mathcal{ALN}$. Примечательно, что в работе рассматривается также применение в онтологии правил логического вывода (horn rules) помимо стандартных возможностей дескриптивной логики. Предложенный в статье [14] алгоритм был частично реализован в системе интеграции данных PICSEL.

В [15] приводится алгоритм максимальной $\mathcal{UCQ}$-переформулировки $\mathcal{CQ}$-$\mathcal{ALE}$ запросов в LAV-системе интеграции данных относительно корректных терминологических отображений $\mathcal{ALE}$. Также показывается, что при замене $\mathcal{ALE}$ на $\mathcal{ALNE}$ получить конечное число максимальных $\mathcal{UCQ}$-переформулировок не всегда возможно.

Наконец, в работе [16] рассматривается LAV-система интеграции с терминологическими отображениями на собственном диалекте дескриптивной логики $\mathcal{AL}$+ (расширение $\mathcal{AL}$ совмещением конструктора $\forall$ с $\exists$, при ограничении формы аксиом вложения концептов). Для такой системы приводится алгоритм точной переформулировки $\mathcal{CQ}$-$\mathcal{AL}$+ запросов в $\mathcal{UCQ}$.

Относительно всех работ [13–16; 33] следует отметить, что с практической точки зрения терминологические запросы или отображения для $\mathcal{AL}$*-диалектов дескриптивной логики не представляют интереса в контексте интеграции данных, поскольку реляционные конъюнктивные запросы имеют больше необходимых на практике выразительных возможностей. Ввод в систему интеграции данных конструкций дескриптивной логики при отказе от конъюнктивных запросов в отображениях не делает ее более выразительной, чем аналогичные реляционные системы интеграции данных, даже наоборот. Кроме того, сами по себе $\mathcal{AL}$*-диалекты дескриптивной логики являются устаревшими. Современные экспрессивные языки дескриптивной логики, в том числе положенные в основу языка веб-онтологий OWL, имеют существенно более высокие выразительные возможности, которые теперь являются неотъемлемой частью понятия онтологии. Прежде всего это касается различных характеристик ролей, их взаимосвязи, ролевых аксиом. В настоящей работе, напротив, делается попытка «обменять» проблемные конструкции $\mathcal{AL}$* на более важные на практике выразительные возможности современных диалектов дескриптивной логики. При этом делается выбор такого языка дескриптивной логики, для которого требуемая переформулировка запросов в системе интеграции данных будет возможна даже при использовании конъюнктивных отображений вместо терминологических. В итоге мы получаем сбалансированную по функциональным возможностям систему интеграции данных, вобравшую в себя допустимый максимум от реляционных систем интеграции данных и от дескриптивной логики при сохранении разрешимости задачи.

В работе [17] проводится теоретический анализ вычислительной сложности ответа на запросы в LAV-системе интеграции данных, в которой пользовательские запросы и отображения задаются объединениями конъюнктивных запросов над дескриптивной логикой $\mathcal{DLR}$ ($\mathcal{UCQ}$-$\mathcal{DLR}$). $\mathcal{DLR}$ является экспрессивным диалектом дескриптивной логики, допускающим n-арные отношения. Сложность рассматривается для корректных, полных и эквивалентных отображений. Показывается, что для корректных отображений задача проверки, принадлежит ли заданный кортеж множеству ответов на заданный запрос к системе, разрешима для $\mathcal{UCQ}$-$\mathcal{DLR}$ запросов и отображений. Однако полученные результаты имеют исключительно теоретический характер и, по замечанию автора работы, не ясно, как предложенная методика может быть использована на практике

Наиболее актуальные результаты, касающиеся ответа на запросы в системах интеграции данных с дескриптивной логикой, отражены в работах 2008 г. В работах [18; 19; 22] впервые рассматривается в этом контексте дескриптивная логика $\mathcal{DL}$-$\mathcal{Lite}$ [21; 24], в которой предпринята попытка выделить максимальный $\mathcal{LOGSPACE}$-разрешимый фрагмент OWL. $\mathcal{DL}$-$\mathcal{Lite}$ включает наиболее практически значимые для построения онтологий конструкции OWL и отбрасывает проблемные конструкции, приводящие к трудноразрешимости.

В работах [18; 22] рассматривается GAV-система интеграции данных на основе дескриптивной логики $\mathcal{DL}$-$\mathcal{Lite}^{A}$, и показывается, что для корректных GAV-отображений возможно раскрытие пользовательских $\mathcal{UCQ}$-$\mathcal{DL}$-$\mathcal{Lite}^{A}$ запросов в формулы реляционного исчисления. Предложенный алгоритм реализован в прототипе системы интеграции данных MASTRO-I. Однако в этих работах рассмотрена лишь достаточно простая задача раскрытия запросов относительно Global-as-view системы интеграции данных. GAV-отображения обладают определенными недостатками с практической точки зрения: термины глобальной схемы в GAV-

системе на практике выбираются «снизу вверх», т. е. отталкиваясь от терминов источников, и нуждаются в постоянном пересмотре при добавлении новых источников. Это приемлемо для интеграции фиксированного набора устаревших источников информации, но неприемлемо для построения «сверху вниз» крупной динамической интеграционной системы. В настоящей работе рассматривается Global-local-as-view система, лишенная указанных недостатков, и показывается возможность GLAV переформулировки $\mathcal{UCQ}$-$\mathcal{DL}$-$\mathcal{Lite}^{R}$ запросов в $\mathcal{UCQ}$. Полученный в настоящей работе результат существенно расширяет сферу применения дескриптивной логики в системах интеграции данных и впервые позволяет построить достаточно выразительную GLAV-систему интеграции данных, расширенную дескриптивной логикой.

Наконец, в работе [19] рассматривается теоретический вопрос определения вычислительной сложности ответа на конъюнктивные запросы относительно онтологий с использованием материализованных представлений. В частности, показывается, что сложность лежит в пределах $\mathcal{LOGSPACE}$ для $\mathcal{DL}$-$\mathcal{Lite}^{R}$, и вне $\mathcal{LOGSPACE}$ для $\mathcal{DL}$-$\mathcal{Lite}^{F,A}$. Для $\mathcal{ALC}$, $\mathcal{ALCQI}$, $\mathcal{SHIQ}$ доказывается *co*-$\mathcal{NP}$-полнота задачи. Система определений в [19] существенно отличается от используемой в настоящей работе, поскольку рассматривается именно ответ на запросы с использованием материализованных представлений, а не система интеграции данных. Кроме того, в этой работе делается попытка рассматривать альтернативную семантику представлений, не имеющую отношения к системам интеграции данных.

Выводы

В настоящей работе формализована математическая модель системы интеграции данных на основе онтологий и рассмотрен выбор параметров задачи переформулировки запросов в такой системе.

Произведен выбор диалекта дескриптивной логики, для которого переформулировка запросов в системе возможна при использовании конъюнктивных отображений и запросов. В итоге впервые предложена сбалансированная по функциональным возможностям GLAV-система интеграции данных на основе онтологий, вобравшая в себя допустимый максимум от реляционных систем интеграции данных и от дескриптивной логики, при сохранении разрешимости задачи и предложен алгоритм переформулировки запросов в такой системе.

Областью непосредственного применения результатов работы является проект построения Единого научного информационного пространства РАН (ЕНИП РАН [34; 35]). Полученные результаты позволяют обеспечить динамическое исполнение распределенных запросов в среде ЕНИП РАН.

Список литературы

1. *Halevy A. Y.*, *Rajaraman A.*, *Ordille J.* Data Integration: The Teenage Years // Proc. of the 32nd International Conference on Very Large Data Bases (VLDB'06). N. Y.: ACM Press, 2006. P. 9–16.

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. OWL Web Ontology Language, http://www.w3.org/TR/owl-features/

4. *Baader F.*, *Calvanese D.*, *McGuinness D.*, *Nardi D.*, *Patel-Schneider P. F.*, The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2003.

5. RDF Primer, http://www.w3.org/TR/rdf-primer/

6. SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/

7. *Horrocks I.*, *Sattler U.* A Tableaux Decision Procedure for SHOIQ // J. of Automated Reasoning. 2007. Vol. 39. No. 3. P. 249–276.

8. *Halevy A. Y.* Answering Queries Using Views. A Survey // VLDB Journal: Very Large Data Bases. 2001. Vol. 10. No. 4. P. 270–294.

9. *Ullman J. D.* Information Integration Using Logical Views // Proc. of the 6th International Conference on Database Theory (ICDT'97): Lecture Notes in Computer Science. Berlin, Germany: Springer, 1997. Vol. 1186. P. 19–40.

10. *Abiteboul S.*, *Duschka O.* Complexity of Answering Queries Using Materialized Views // Proc. of the 17th ACM SIGACT-SIGMOD-SIGART Conference on Principles of Database Systems (PODS'98). Seattle, WA, 1998. P. 254–265.

11. *Levy A. Y.*, *Rajaraman A.*, *Ordille J. J.* Querying Heterogeneous Information Sources Using Source Descriptions // Proc. of the International Conference on Very Large Databases (VLDB). N. Y.: ACM Press, 1996. P. 251–262.

12. *Chawathe S.*, *Garcia-Molina H.*, *Hammer J.*, *Ireland K.*, *Papakonstantinou Y.*, *Ullman J.*, *Widom J.*, The TSIMMIS Project: Integration of Heterogeneous Information Sources // Proc. of IPSJ. Tokyo, 1994.

13. *Beeri C.*, *Levy A. Y.*, *Rousset M.-C.* Rewriting Queries Using Views in Description Logics // Proc. of the Sixteenth ACM SIG-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'97). N. Y.: ACM Press, 1997. P. 99–108.

14. *Goasdoué F.*, *Lattes V.*, *Rousset M.-C.* The Use of CARIN Language and Algorithms for Information Integration: The PICSEL Project // International Journal of Cooperative Information Systems (IJCIS). 2000. Vol. 9. No. 4. P. 383–401.

15. *Goasdoué F., Rousset M.-C.* Rewriting Conjunctive Queries Using Views in Description Logics with Existential Restrictions // Description Logics (DL 2000). Aachen, Germany, 2000. P. 113–122.

16. *Goasdoué F.*, *Rousset M.-C.* Answering Queries Using Views: a KRDB Perspective for the Semantic Web // ACM Journal – Transactions on Internet Technology (TOIT). 2004. Vol. 4. No. 3. P. 255–288.

17. *Calvanese D.*, *De Giacomo G.*, *Lenzerini M.* Answering Queries Using Views in Description Logics // Proc. of the 17th Nat. Conf. on Artificial Intelligence. AAAI 2000. P. 386–391.

18. *Poggi A.*, *Lembo D.*, *Calvanese D.*, *De Giacomo G.*, *Lenzerini M.*, *Rosati R.* Linking data to ontologies // J. on Data Semantics. 2008. P. 133–173.

19. *Calvanese D.*, *De Giacomo G.*, *Lenzerini M.*, *Rosati R.* View-Based Query Answering Over Description Logic Ontologies // Proc. of the 11th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2008). Berlin: Springer, 2008.

20. RDF Vocabulary Description Language 1.0: RDF Schema, http://www.w3.org/TR/rdf-schema/

21. *Calvanese D.*, *De Giacomo G.*, *Lembo D.*, *Lenzerini M.*, *Rosati R.* Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family // J. of Automated Reasoning. 2007. Vol. 39. No. 3. P. 385–429.

22. *Calvanese D.*, *De Giacomo G.*, *Lembo D.*, *Lenzerini M.*, *Poggi A.*, *Rosati R.*, *Ruzzi M.* Data Integration Through DL-LiteA Ontologies // Proc. of the Int. Workshop on Semantics in Data and Knowledge Bases (SDKB 2008): Lecture Notes in Computer Science. Berlin: Springer, 2008.

23. *Patel-Schneider P. F.*, *Horrocks I.*, *Motik B.* OWL 1.1 Web Ontology Language: Structural Specification and Functional-Style Syntax., eds., 2006. http://www.w3.org/2007/OWL/wiki/Syntax

24. *Calvanese D.*, *De Giacomo G.*, *Lembo D.*, *Lenzerini M.*, *Rosati R.*, Data Complexity of Query Answering in Description Logics // Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'06). Berlin: Springer, 2006. P. 260–270.

25. *Tobies S.* Complexity Results and Practical Algorithms for Logics in Knowledge Representation: Ph.D. Dissertation. Aachen, Germany, 2002.

26. *Donini F. M*, *Lenzerini M.*, *Nardi D.*, *Schaerf A.* Deduction in Concept Languages: From Subsumption to Instance Checking // J. Logic Computat. 1994. Vol. 4. No. 4. P. 423–452.

27. *Grau B.C.*, OWL 1.1 Web Ontology Language Tractable Fragments. http://www.w3.org/Submission/owl11-tractable/

28. *Baader F.*, *Brandt S.*, *Lutz C.* Pushing the EL Envelope // Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005). San Francisco: Morgan Kaufmann, 2005. P. 364–369.

29. *Grosof B.*, *Volz R.*, *Horrocks I. Decker S.* Description Logic Programs: Combining Logic

Programs with Description Logics // Proc. of the 12th International World Wide Web Conference (WWW 2003). N. Y.: ACM Press, 2003.

30. *Hustand U.*, *Motik B.*, *Sattler U.*, Data Complexity in Very Expressive Description Logics // Proc. of the 19th Joint Int. Conf. on Artificial Intelligence (IJCAI 2005). Menlo Park, CA: AAAI Press, 2005.

31. *Gryz J.* Query Rewriting Using Views in the Presence of Functional and Inclusion Dependencies // J. Information Systems. 1999. Vol. 24. No. 7. P. 597–612.

32. *Calì A.*, *Lembo D.*, *Rosati R.* Query Rewriting and Answering under Constraints in Data Integration Systems // Proc. of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003). Berlin: Springer, 2003. P. 16–21.

33. *Baader F.*, *Kusters R.*, *Molitor R.* Rewriting Concepts Using Terminologies // Proc. of the 17th International Conference on Knowledge Representation and Reasoning (KR 2000). San Francisco: Morgan Kaufmann Publishers, 2000. P. 297–308.

34. *Бездушный А. А.*, *Бездушный А. Н.*, *Серебряков В. А.*, *Филиппов В. И.*, Интеграция метаданных Единого научного информационного пространства РАН / Вычислительный Центр им. А. А. Дородницына РАН. М., 2006.

35. *Бездушный А. А.*, *Бездушный А. Н.*, *Жижченко А. Б.*, *Калёнов Н. Е.*, *Кулагин М. В.*, *Серебряков В. А.* Предложения по наборам метаданных для научных информационных ресурсов ЕНИП РАН // 6 Всерос. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2004). Пущино, 2004. С. 277–284.

Материал поступил в редколлегию 20.05.2008

A. A. Bezdushny

Formal Model of Ontology-Based Data Integration Systems

The paper proposes a formal theory for ontology-based data integration systems, and considers the query rewriting problem for such class of data integration systems. A choice of appropriate description logic dialects and query languages is studied, provided that the rewriting problem is decidable. A query rewriting algorithm is proposed for an important class of ontology-based data integration systems.

Keywords: data integration, ontologies, description logics, OWL.