Серия «Математика»
2011. Т. 4, № 1. С. 20—30

Онлайн-доступ к журналу:
http://isu.ru/izvestia

ИЗВЕСТИЯ
Иркутского
государственного
университета

УДК 510.62:004.82

Базы данных как онтологии *

И. А. Казаков, А. В. Манцивода
Иркутский государственный университет

Аннотация. В статье исследуется проблема интерпретации баз данных как онтологий. Представлен метод погружения баз данных в онтологии через их моделирование с помощью объектных теорий. Решается проблема построения замкнутого мира баз данных в рамках открытого мира дескриптивных логик. Результаты работы имеют как теоретическое, так и практическое значение, предоставляя метод единообразной работы с базами данных как онтологиями в логических базах знаний. В частности, данный метод применяется для работы с естественнонаучными базами данных, связанными с озером Байкал.

Ключевые слова: онтологии; базы данных; объектные теории; дескриптивные логики; Libretto.

1. Введение

В реляционных базах данных [5] накоплен огромный объем информации о самых разнообразных отраслях знаний. Вовлечение этого гигантского объема в орбиту систем автоматической обработки знаний является важнейшей задачей, и с этим связан ряд интересных проблем. В частности, из-за глубоких различий в формализмах непростой проблемой является интеграция баз данных и систем знаний. Сегодня основной тренд развития состоит в разработке гибридных систем, объединяющих разные формализмы (см. напр. [14]). Мы считаем такой подход контрпродуктивным и ведущим к большим неудачам, главная из которых состоит в том, что тяжелый и малоуправляемый гибридный формализм излишне труден для массового пользователя, что резко снижает шансы на продвижение этих технологий. В настоящей работе рассматривается подход, основанный на погружении баз данных в

* Работа выполнена при финансовой поддержке Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009-2013 гг., государственный контракт № П45 от 02.04.2010.

объектные теории [3] – простые дескриптивные логики [16] со специальными свойствами, что позволяет организовать единую среду данных и знаний. Особое внимание уделяется преодолению трудностей, связанных с концептуальными различиями в подходах. В первую очередь речь идет о несовместимости открытого мира дескриптивных логик, лежащих в основе многих проектов, связанных с обработкой знаний, включая семантический веб [17], и замкнутого мира баз данных (о проблеме открытого и замкнутого мира см. [12][15]). В данной работе предлагается решение этой проблемы на основе концепции логических архитектур [4].

Метод, разработанный в настоящей работе, используется как основа для манипулирования базами данных в рамках системы Libretto [11][2]. Интересно использовать данный метод для создания семантических гридов [13], интегрирующих гетерогенные информационные источники, среди которых имеются реляционные базы данных.

2. Базы данных и реляционная алгебра

В этом пункте дается стандартное формальное определение реляционной базы данных. Начнем с типов данных. Домен – это набор значений некоторого базового типа. В дальнейшем считается, что мы имеем фиксированный набор доменов

$$\mathbb{D} = \{D_1, \ \dots\ , \ D_m\}, \tag{2.1}$$

где каждый D_i представляет определенный тип (например, целых чисел или строк). Считаем, что каждый тип данных содержит выделенный элемент `NULL`, причем

$$D_i \cap D_j = \{\mathtt{NULL}\} \tag{2.2}$$

для каждой пары $D_i, D_j \in \mathbb{D}$. Обозначим

$$\mathcal{D} = D_1 \cup \ldots \cup D_m$$

Пусть $\mathbb{ID}$ – счетное множество имен (идентификаторов).

Определение 1. k–местное табличное отношение *есть тройка* $\mathbf{r} = \langle R, body_{\mathbf{r}}, header_{\mathbf{r}}\rangle$, *где*

— $R \in \mathbb{ID}$ – *имя отношения,*

— $body_{\mathbf{r}} \subseteq D^1 \times D^2 \times \dots \times D^k$, $|\, body_{\mathbf{r}}\,| < \infty$, *и*

— $header_{\mathbf{r}} = \langle\langle A_1, D^1\rangle, \dots, \langle A_k, D^k\rangle\rangle$, $A_i \in \mathbb{ID}, i \in \overline{1,k}$,

для некоторых $D^1, \dots, D^k \in \mathbb{D}$. *Конечное множество* $body_{\mathbf{r}}$ *называется* телом *табличного отношения, а кортеж* $header_{\mathbf{r}}$ – заголовком $\mathbf{r}$.

Табличное отношение моделирует таблицу базы данных, кортежи из $body_{\mathbf{r}}$ играют роль строк таблицы, а заголовок $header_{\mathbf{r}}$ задает имена столбцов (которые мы будем называть *атрибутами*) и привязывает атрибуты к типам данных. *Сигнатурой* $sign(\mathbf{r})$ отношения $\mathbf{r}$ назовем множесто его атрибутов:

$$sign(\mathbf{r}) = \{A_1, \dots, A_k\}$$

Для кортежа $t \in body_{\mathbf{r}}$ определим одноименную функцию

$$t(A_i) = t[i],$$

получающую значение ячейки из столбца по его атрибуту A_i.

Лемма 1. *$t[i] \in D^i$, где $\langle A_i, D^i \rangle = header_R[i]$.*

Доказательство. Непосредственно следует из определения. □

Мы готовы дать определение основного понятия данного пункта.

Определение 2. База данных $\mathbb{DB} = \{\mathbf{r}^1,\ \mathbf{r}^2, \dots, \mathbf{r}^n\}$ *– это конечное множество табличных отношений.* Сигнатурой базы данных $\mathbb{DB}$ *назовем множество всех ее атрибутов:* $sign(\mathbb{DB}) = \bigcup_{i=1}^{n} sign(\mathbf{r}^i)$.

3. Базы данных как объектные дескриптивные теории

Рассмотрим произвольную базу данных с точки зрения объектных теорий [3]. Будем считать, что онтологии основаны на тех же типах данных (доменах), что и базы данных:

$$\mathbb{D} = \{D_1, \dots, D_m\}$$

вида (2.1), за исключением выделенного элемента `NULL`.

Пусть $\mathbb{DB} = \{\mathbf{r}^1, \dots, \mathbf{r}^n\}$ – база данных, состоящая из табличных отношений с заголовками $header_{\mathbf{r}^i} = \langle\langle A_1^i, D_1^i\rangle, \dots, \langle A_k^i, D_k^i\rangle\rangle$, где A_j^i – атрибуты и $D_j^i \in \mathbb{D}$.

Определим отображение сущностей базы данных (отношений и атрибутов) в сущности онтологии (концепты и т-свойства). Чтобы отобразить компоненты базы данных в компоненты онтологии, вводится оператор $E \xrightarrow{\mathtt{tr}} \mathtt{e}$, переводящий сущность E базы данных в соответствующую ей сущность $\mathtt{e}$ онтологии: $\mathtt{tr}(E) = \mathtt{e}$. Для множества сущностей полагаем $\mathtt{tr}(\{E_1, \dots, E_k\}) = \{\mathtt{tr}(E_1), \dots, \mathtt{tr}(\mathrm{E_k})\}$.

Начнем с сопоставления каждому табличному отношению $\mathbf{r}^i = \langle R^i, body_{\mathbf{r}^i}, header_{\mathbf{r}^i}\rangle$ имени концепта C^i:

$$\mathtt{tr}(\mathbf{r}^i) = C^i$$

Атрибуты (столбцы) таблиц будут моделироваться с помощью т-свойств. Используя механизмы наследования в объектно–ориентированном стиле, мы можем построить весьма гибкую модель базы данных $\mathbb{DB}$. Для этого заведем отдельный класс для каждого столбца, чтобы затем собирать таблицы из столбцов через наследование. С каждым атрибутом базы данных A^i_j отношения $\mathbf{r}^i$ сопоставляем имя концепта B^i_j и имя т-свойства P^i_j, $i \in \overline{1,n}$; $j \in \overline{1,k}$ (совпадающим именам атрибутов из разных табличных отношений должны соответствовать разные имена концептов и т-свойств):

$$\mathtt{tr}(A^i_j) = \{B^i_j, P^i_j\}$$

Будем описывать семантику новых сущностей с помощью аксиом объектных теорий (см. [3]). Определим семантику концепта B^i_j и т–свойства P^i_j с помощью аксиомы домена:

$$\exists P^i_j \sqsubseteq B^i_j, j \in \overline{1,k} \tag{3.1}$$

и ранга

$$\exists P^{i-}_j \sqsubseteq D^i_j, j \in \overline{1,k} \tag{3.2}$$

где D^i_j – область значений для атрибута A^i_j. Поскольку в ячейке таблицы не может быть более одного значения, то P^i_j может принимать либо ни одного, либо одно значение (отсутствие значений моделирует NULL):

$$B^i_j \sqsubseteq\ \leq 1.P^i_j, j \in \overline{1,k} \tag{3.3}$$

Теперь "соберем" семантику табличного концепта C^i как наследника всех своих столбцов:

$$\begin{array}{c} C^i \sqsubseteq B^i_1 \\ \cdots \\ C^i \sqsubseteq B^i_k \end{array} \tag{3.4}$$

Нам осталось определить объекты, которые будут моделировать конкретные данные, представленные в $\mathbb{DB}$ в виде кортежей (строк таблиц). Пусть

$$body_{\mathbf{r}^i} = \{t^i_1, \ldots, t^i_l\}$$

набор кортежей отношения $\mathbf{r}^i$. Введем новую константу o^i_j, определив

$$o^i_j = \mathtt{tr}(t^i_j).$$

Введем для o^i_j аксиому принадлежности табличному концепту

$$C^i(o^i_j), \tag{3.5}$$

где $C^i = \mathtt{tr}(\mathbf{r}^i)$ и аксиомы значений т-свойств

$$P^i_s(o^i_j, t^i_j(A^i_s)) \tag{3.6}$$

для всех атрибутов $A_s^i \in sign(\mathbf{r}^i)$ таких, что $t_j^i(A_s^i) \neq \mathtt{NULL}$ и $P_s^i \in \mathtt{tr}(A_s^i)$. Этот момент является существенным: в объектных теориях отсутствие значения свойства моделируется не с помощью явной константы NULL, а через отсутствие соответствующей аксиомы.

Таким образом, определим для отношения $\mathbf{r}^i$ множество аксиом $\mathbb{O}^i$, состоящее из k аксиом вида (3.1)–(3.4), а также наборов аксиом вида (3.5) и (3.6) для каждого кортежа t_j^i и каждого непустого значения $t_j^i(A_t^i)$, соответственно.

Определение 3. *Множество аксиом*

$$\mathbb{O}_{\mathbb{DB}} = \bigcup_{i=1}^{n} \mathbb{O}^i$$

назовем объектной теорией *базы данных* $\mathbb{DB}$. *Будем говорить, что сущности объектной теории (концепты C_i, объекты o_i и т.д.)* моделируют *в $\mathbb{O}_{\mathbb{DB}}$ соответствующие сущности базы $\mathbb{DB}$ (отношения $\mathbf{r}_i$, строки t_i и т.д.)*

В случаях, когда база данных известна из контекста, вместо $\mathbb{O}_{\mathbb{DB}}$ иногда будем писать $\mathbb{O}$.

Пусть $\mathtt{tr}(A) = \{B, P\}$ для некоторого атрибута A. Обозначим $\mathtt{tr}(A)_{\mathcal{C}} = B$ и $\mathtt{tr}(A)_{\mathcal{P}} = P$.

Лемма 2. *$\mathbb{O}_{\mathbb{DB}}$ является объектной теорией [3] в словаре $\mathcal{W} = \langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$, где*

$$\mathcal{C} = \mathtt{tr}(\{\mathbf{r}^1, \dots, \mathbf{r}^n\}) \cup \mathtt{tr}(sign(\mathbb{DB}))_{\mathcal{C}} = \{C^1, \dots, C^n\} \cup \bigcup_{i=1}^{n} \{B_1^i, \dots, B_{k^i}^i\}$$

$$\mathcal{R} = \emptyset$$

$$\mathcal{P} = \mathtt{tr}(sign(\mathbb{DB}))_{\mathcal{P}} = \bigcup_{i=1}^{n} \{P_1^i, \dots, P_{k^i}^i\}$$

$$\mathcal{O} = \bigcup_{i=1}^{n} \mathtt{tr}(body_{\mathbf{r}^i}) = \bigcup_{i=1}^{n} \{o_1^i, ..., o_{l^i}^i\}$$

Доказательство. По определению объектной теории мы должны, во-первых, показать, что множество аксиом наследования ациклично, и это выполняется, поскольку аксиомы наследования применяются только к концептам C^i, и они могут наследовать только от элементов другой изолированной группы концептов B_j^i. Во-вторых, каждое свойство P_j^i обладает ровно одной аксиомой домена и одной ранга, таким образом п. 2 определения объектной теории [3] также выполняется. Осталось показать, что $\mathbb{O}_{\mathbb{DB}}$ непротиворечиво. В [3] показано, что $TBox$ объектных теорий без аксиом кардинальности всегда непротиворечив. Чтобы аксиома кардинальности (3.3) противоречила другим аксиомам, необходимо, чтобы из (3.1), (3.2) и (3.4) выводилось наличие для некоторого объекта более одного значения свойства P_i^j, что, очевидно, невозможно. Нам осталось проверить то, что аксиомы из $ABox$ не противоречат аксиомам домена, ранга и кардинальности из $TBox$. Эта проверка легко осуществляется с помощью свойств самой базы данных $\mathbb{DB}$ и метода построения объектной модели $\mathbb{O}_{\mathbb{DB}}$. $\square$

Таким образом, оператор $\stackrel{\mathtt{tr}}{\to}$ переводит структуры баз данных в корректные объектные теории.

4. Аксиомы замкнутости мира

Разберемся с парадигмой замкнутого мира баз данных. Теория $\mathbb{O}_{\mathbb{DB}}$ содержит всю позитивную информацию, связанную с базой данных $\mathbb{DB}$ (то есть, явно определяет, что *содержится* в $\mathbb{DB}$). Однако силы логики $\mathcal{OODL}$ [3], на которой построены объектные теории, недостаточно для того, чтобы выразить то, чего *нет* в $\mathbb{DB}$. Например, пусть $\mathbb{DB}$ определяет табличные отношения $\mathbf{r}_1$ и $\mathbf{r}_2$, которые переводятся отображением $\stackrel{\mathtt{tr}}{\to}$ в концепты C_1 и C_2. Получившаяся теория $\mathbb{O}_{\mathbb{DB}}$ не запрещает выбора в качестве своей модели таких интерпретаций, в которых истинно $C_1 \sqsubseteq C_2$, то есть C_1 наследует от C_2. С точки зрения баз данных это означает, что в этой интерпретации все строки таблицы, соответствующей концепту C_1, включаются в строки таблицы, соответствующей C_2, что противоречит идеологии баз данных – таблицы не могут наследовать друг от друга. Однако слабая логика $\mathcal{OODL}$ неспособна выразить это ограничение, поскольку для него нужно отрицание (например, это ограничение может быть выражено аксиомой $C_1 \sqsubseteq \neg C_2$) или конъюнкция ($C_1 \sqcap C_2 \sqsubseteq \bot$).

Эти локальные проблемы имеют глубокую концептуальную основу, связанную с различием в семантике баз данных и теорий в дескриптивных логиках. Первые построены на парадигме замкнутого мира, в то время как вторые основаны на парадигме открытого мира (см., например, [12][15]). Семантика замкнутого мира, в отличие от открытого, подразумевает, что вся информация о предметной области нам известна, и если что-то мы не знаем, то это ложно. В семантике открытого мира для того, чтобы доказать ложность какого-то утверждения, надо явно вывести противоречие. Для нас это различие имеет серьезное прикладное значение: в дескриптивной логике нельзя вывести часть информации, которая считается истинной в базе данных. В частности, необходимо явно формулировать то, чего *не может быть* в модели, описывающей базу данных. С точки зрения баз данных замкнутость мира уточняется следующим образом:

1. Нет строк, кроме явно перечисленных.

2. Нет столбцов вне таблиц.
3. Нет непустых данных (не NULL), кроме явно перечисленных.
4. Разные таблицы не содержат общих строк.

Выразительности логики $\mathcal{OODL}$ не хватает для формализации ни одного из перечисленных выше ограничений. Это как раз тот случай, когда срабатывает двухуровневая логическая архитектура, лежащая в основе системы Libretto:

$$\mathcal{OODL} \subseteq \mathcal{SHOIN}(D)$$

Если в слабой но очень эффективной логике $\mathcal{OODL}$ описывается позитивная информация о содержимом базы данных, то более тяжелая, но выразительная логика $\mathcal{SHOIN}(D)$ [9][3] может быть использована для описания негативных ограничений, позволяющих реализовать семантику замкнутого мира базы данных.

Пусть $\mathbb{O}$ – теория базы данных $\mathbb{DB}$ в словаре $\mathcal{W} = \langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$. Покажем, что все четыре условия замкнутости выразимы в логике $\mathcal{SHOIN}(D)$.

Первое свойство:

$$\top \sqsubseteq \bigsqcup_{o \in \mathcal{O}} \{o\} \tag{4.1}$$

где o пробегает по всем объектам из $\mathcal{O}$, моделирующим строки таблиц базы данных. Заметим, что по построению $\mathbb{O}$ из аксиомы (4.1) следует $\top \sqsubseteq \bigsqcup_{C \in \{C_1, \ldots, C_n\}} C$, поскольку каждый объект o должен обладать ровно одной аксиомой $C(o) \in \mathbb{O}$.

Выразим второе свойство для произвольного атрибута столбца B_j. Пусть $C_1 \sqsubseteq B_j, \ldots, C_l \sqsubseteq B_j$ – все аксиомы наследования вида (3.4) для B_j, содержащиеся в $\mathbb{O}$. Тогда добавим следующее свойство:

$$B_j \sqsubseteq C_1 \sqcup \ldots \sqcup C_l \tag{4.2}$$

Третье свойство описывает значения P_j. Пусть $P_j(o_1, v_1), \ldots, P_j(o_m, v_m) \in \mathbb{O}$ – все вхождения атомарных формул с P_j. Тогда добавляем свойство

$$\exists P_j \sqsubseteq \{o_1\} \sqcup \ldots \sqcup \{o_m\} \tag{4.3}$$

Четвертое свойство для двух таблиц C_i и C_j формализуется с помощью следующей аксиомы

$$C_i \sqcap C_j \sqsubseteq \bot \tag{4.4}$$

Множество аксиом $\mathbb{CW}$ назовем теорией замкнутого мира для $\mathbb{O}$, если оно состоит из аксиомы (4.1), аксиомы вида (4.2) для каждого концепта–столбца $B_i \in \mathcal{C}$, аксиомы (4.3) для каждого $P_j \in \mathcal{P}$, и аксиомы (4.4) для каждой пары табличных концептов $C_i, C_j \in \mathcal{C}$. Благодаря аксиомам (4.1) и (4.4), $C \in \mathcal{C}$ образуют покрытие предметной области конечной коллекцией непересекающихся множеств.

Конечно, нам необходимо обосновать гипотезу о том, что аксиом из $\mathbb{CW}$ достаточно для того, чтобы формализовать парадигму замкнутого мира. Для этого введем понятие минимальной модели $\mathbb{O}$, то есть такой модели, на которой истинны только те аксиомы баз данных, которые явно присутствуют в $\mathbb{O}$.

Определение 4 (Минимальная модель). *Пусть $\mathbb{O}$ – объектная теория базы данных $\mathbb{DB}$. Интерпретацию $I \models \mathbb{O}$ назовем минимальной моделью $\mathbb{O}$, если все объекты Δ^I выделенные и для любого утверждения $\mathcal{A}$ вида (3.1)–(3.6) из $I \models \mathcal{A}$ следует $\mathbb{O} \vdash \mathcal{A}$.*

Под выводимостью $\vdash$ понимается любое полное исчисление в дескриптивных логиках, например, табличный алгоритм [16]. Поскольку позитивную информацию о базе данных задают аксиомы (3.1)–(3.6) из объектной теории $\mathbb{O}$, то для реализации парадигмы замкнутого мира нам достаточно, чтобы на интерпретации I не выполнялись никакие выражения вида (3.1)–(3.6), кроме заданных в $\mathbb{O}$. Для формул любого другого вида такое требование замкнутости было бы некорректным, поскольку означало бы попытку распространить парадигму замкнутого мира за рамки необходимого. Именно эти положения и реализованы в минимальных моделях – любое утверждение о базе данных $\mathbb{DB}$, похожее на аксиомы (3.1)–(3.6), в них ложно, если оно явно не следует из $\mathbb{O}$.

Мы готовы сформулировать и доказать основную теорему данной работы.

Теорема 1. *Пусть $\mathbb{DB}$ – база данных с объектной теорией $\mathbb{O}$ и множеством аксиом замкнутого мира $\mathbb{CW}$. Для любой интерпретации I следующие два условия эквивалентны:*

1. $I \models \mathbb{O} + \mathbb{CW}$.

2. I является минимальной моделью $\mathbb{O}$.

Доказательство. 1. $\Rightarrow$ 2. Пусть $I \models \mathbb{O} + \mathbb{CW}$. Сразу видим, что благодаря аксиоме (4.1) выполняется свойство выделенности элементов минимальной модели (каждый элемент имеет константу, его обозначающую). Продолжаем доказательство разбором случаев по типу аксиом: нам нужно показать, что любое утверждение вида (3.1)–(3.6), истинное в I, выводимо из $\mathbb{O} + \mathbb{CW}$.

Аксиома (3.1). Пусть $I \models \exists P_i \sqsubseteq B_j$. Случай $\exists P_i^I = \emptyset$. Тогда $\mathbb{O}$ не содержит ни одной аксиомы вида $P_i(o, v)$, а соответствующая аксиома (4.3) принимает вид $\exists P_i \equiv \bot$, откуда, очевидно, следует $\exists P_i \sqsubseteq E$ для любого E. Случай $\exists P_i^I \neq \emptyset$. Пусть $o^I \in \exists P_i^I$. По правилам построения $\mathbb{O}$, и по аксиомам (4.1) и (4.4) существует ровно один концепт C_k такой, что $C_k(o) \in \mathbb{O}$. По аксиоме (4.2) для B_j и аксиоме (4.4), C_k должен содержаться в правой части (4.2). Тогда по правилам построения (4.2) теория $\mathbb{O}$ содержит аксиому $C_k \sqsubseteq B_j$. Поэтому в любой интерпретации $B_j(o)$ истинно. Поскольку o – произвольно выбранный элемент из $\exists P_i^I$, то $\exists P_i \sqsubseteq B_j$ истинна в любой интерпретации теории $\mathbb{O} + \mathbb{CW}$, и поскольку $\vdash$ – полный вывод, то $\mathbb{O} + \mathbb{CW} \vdash \exists P_i \sqsubseteq B_j$.

Аксиома (3.2). Пусть $I \models \exists P_j^- \sqsubseteq D_i$. Случай $(\exists P_j^-)^I = \emptyset$ рассматривается так же, как для аксиомы (3.1), поскольку пустота $\exists P_j^-$ влечет, по определению, пустоту $\exists P_j$, после чего применяется аксиома (4.3). Пусть $d \in (\exists P_j^-)^I$. Тогда по аксиоме (3.2) для P_j: $d \in D_j$, что в соответствии со свойством (2.2) возможно только в том случае, когда $i = j$ и $(\exists P_j^- \sqsubseteq D_i) \in \mathbb{O}$.

Аксиома (3.3). Пусть $I \models B_i \sqsubseteq \ \leq 1.P_j$. Из аксиом (3.1) и (3.3) следует $\mathbb{O} + \mathbb{CW} \vdash \top \sqsubseteq \ \leq 1.P_j$, а значит $\mathbb{O} + \mathbb{CW} \vdash B_i \sqsubseteq \ \leq 1.P_j$.

Аксиома (3.4). Пусть $I \models C_i \sqsubseteq B_j$. Случай $C_i^I = \emptyset$. Тогда из аксиом (3.5), (4.2), (4.4) выводимо $C_i \sqsubseteq \bot$, что сильнее $C_i \sqsubseteq B_j$. Случай $C_i^I \neq \emptyset$. Тогда, в силу аксиомы (4.4), C_i должен присутствовать в правой части аксиомы (4.2) для B_j. Отсюда, по правилам построения (4.2), следует, что $C_i \sqsubseteq B_j$ является аксиомой $\mathbb{O}$.

Случай (3.5). Пусть $I \models C_i(o)$. По правилам построения аксиом (3.5) и аксиоме (4.1), $C_j(o) \in \mathbb{O}$ для некоторого C_j. По (4.4) $i = j$ и, таким образом, $C_i(o) \in \mathbb{O}$.

Случай (3.6). Пусть $I \models P_j(o, v)$. Тогда o должна присутствовать в правой части аксиомы (4.3). Следовательно, по правилам построения (4.3) $P_j(o, v') \in \mathbb{O}$ для некоторого v'. В силу $\mathbb{O} + \mathbb{CW} \vdash \top \sqsubseteq \ \leq 1.P_j$ (см. выше), очевидно, что $v = v'$.

2. $\Rightarrow$ 1. Пусть I – минимальная модель $\mathbb{O}$. Нам нужно показать, что $I \models \mathbb{CW}$. Это доказывается разбором случаев для каждой аксиомы из $\mathbb{CW}$. Непосредственно проверяется, что каждая из этих аксиом истинна на I. Например, аксиома (4.1) истинна в силу свойства минимальной модели, что каждый ее элемент должен быть выделенным. Остальные проверяются аналогично. $\square$

Если вернуться к проблеме наследования одной таблицы от другой $(C_1 \sqsubseteq C_2)$, упомянутой в начале этого пункта, то теперь это невозможно для любой непустой C_1 в силу аксиомы (4.4).

5. Заключение

В данной работе была построена модель реляционных баз данных в терминах объектных теорий [3] в рамках логической архитектуры $\mathcal{OODL} \subseteq \mathcal{SHOIN}(D)$. Показано, что предлагаемый нами подход позволяет смоделировать не только структуру реляционных баз данных, но и парадигму замкнутого мира, действующую в этих базах. Разработанный метод ориентирован на реализацию в системе Libretto [2], поскольку объектные теории описывают логическую семантику хранилища знаний в системе Libretto, а на логике $\mathcal{SHOIN}(D)$ основан язык запросов системы. В планы дальнейших исследований входит построение алгебры Кодда [5] и моделирование ER–моделей [6][7] в терминах объектных теорий. В частности, в данной работе не были затронуты вопросы моделирования внешних ключей и связей объектов друг с другом, связей один-ко-многим, что легко реализуется в объектных теориях с помощью о-свойств (ролей). Планируется использовать разработанный метод как для создания интегральных баз знаний, включающих реляционные базы данных, так и для разработки технологии программирования, объединяющей реляционные хранилища данных с объектно–ориентированными языками программирования. Этот подход будет, в частности, ориентирован на те задачи, для решения которых сегодня используется технология Hibernate [10]. В дальнейшем также планируется использовать для моделирования этих интегрированных формализмов теорию наследственно–конечных списочных надстроек [1][8]. На прикладном уровне рассматриваемый в данной работе метод используется для построения междисциплинарной естественнонаучной базы знаний об озере Байкал – решается задача по включению в базу знаний ряда баз данных, содержащих информацию об озере и окружающей территории.

Список литературы

1. Гончаров С. С. Σ–программирование / С. С. Гончаров, Д. И. Свириденко // Вычислительные системы. – Новосибирск, 1985. – Вып. 107. – С.3–29.
2. Libretto: объектно-итерационный язык и система управления объектными базами знаний [Электронный ресурс]. – URL: http://ontobox.org.
3. Малых А. А. Объектно-ориентированная дескриптивная логика / А. А. Малых, А. В. Манцивода // Изв. Иркут. гос. ун-та. Сер.: Математика. – 2011.– Т. 4, № 1.
4. Малых А. А. Логические архитектуры и объектно-ориентированный подход / А.А. Малых, А.В. Манцивода, В.С. Ульянов // Вестн. НГУ. Сер.: Математика, механика, информатика. – 2009. – Т. 9, № 3. – С. 64–85.
5. Abiteboul S. Foundations of Databases / Serge Abiteboul, Richard B. Hull, and Victor Vianu. – Addison-Wesley, 1995. – 685 p.
6. Chen P. P. The Entity-Relationship model: Towards a unified view of data / P. P. Chen // ACM Transaction of Database Systems.– 1976. – Vol. 1(1). – P.9–36.
7. Calvanese D. Unifying class based representation formalizms / D. Calvanese, M. Lenzerini, D. Nardi. // Journal of AI Research. – 1999. - Vol. 11. – P. 199–204.
8. Ershov Yu.L. Semantic Programming / Yu. L. Ershov, S. S. Goncharov, D. I. Sviridenko // Information processing, Proc. IFIP 10th World Comput. Congress, Dublin. – 1986. – Vol. 10. – P.1093–1100.
9. Horrocks I. Patel-Schneider. Reducing OWL entailment to description logic satisfiability / Ian Horrocks and F. Peter // : Proc. of the 2003 International Semantic Web Conference (ISWC 2003)/ eds. D. Fensel, K. Sycara, J. Mylopoulos. – Springer, 2003. – Lecture Notes in Computer Science 2870. – P. 17-29.

10. Hibernate: Mapping an Object-Oriented Domain Model to a Relational Database [Electronic resource]. – URL: https://www.hibernate.org/.
11. Malykh A. A Query Language for Logic Architectures / A. Malykh, A. Mantsivoda // Proceedings of 7th International Conference «Perspectives of System Informatics». – Springer-Verlag Berlin Heidelberg, Lecture Notes in Computer Science 5947. – 2010. – P. 294–305.
12. Mazzocchi S. Closed World vs. Open World: the First Semantic Web Battle [Electronic resource] / Stefano Mazzocchi. – URL: http://www.betaversion.org/ stefano/linotype/news/91/.
13. Murphy M. J. Towards the Semantic Grid: A State of the Art Survey of Semantic Web Services and their Applica0bility to Collaborative Design, Engineering, and Procurement / M. J. Murphy, M. Dick, T. Fischer // Communications of the IIMA. - 2008. – Vol.8, Issue 3. – P. 11–24.
14. Rosati R. On Combining Description Logic Ontologies and Nonrecursive Datalog Rules / R. Rosati // Lecture Notes in Computer Science. – 2008. – Vol. 341. – Web Reasoning and Rule Systems. – P. 13–27.
15. Supporting Open and Closed World Reasoning on the Web / Carlos Viegas Damasio, Anastasia Analyti, Grigoris Antoniou and Gerd Wagner // Lecture Notes in Computer Science. – 2006. – Vol. 4187. – P.149–163.
16. The Description Logic Handbook: Theory, Implementation, Applications / F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P.F. Patel-Schneider. – Cambridge, 2003. – P.574.
17. W3C Semantic Web Activity [Electronic resource]. – URL: http://www.w3.org/2001/sw/.

A. V. Mantsivoda, I. A. Kazakov

Databases as Ontologies

Abstract. In this paper the problem of the interpretation of databases as ontologies is investigated. A method for embedding databases in ontologies via their simulation within object theories is established. The problem of modeling the DB's closed world within the open world of description logics is considered. The results of the paper have both theoretical and practical impact, because they offer a homogenuous and coherent method for databases-as-ontologies manipulation in logical knowledge bases. In particular, this method has been applied to handling natural science databases describing Lake Baikal.

Keywords: ontology; database; object theory; description logic, Libretto

Казаков Илья Анатольевич, Институт математики, экономики и информатики, Иркутский государственный университет, 664003, Иркутск, ул. К. Маркса, 1 тел.: (3952)242210 (kazakov@baikal.ru)

Манцивода Андрей Валерьевич, Институт математики, экономики и информатики, Иркутский государственный университет, 664003, Иркутск, ул. К. Маркса, 1 тел.: (3952)242210 (andrei@baikal.ru)

Ilya Kazakov, Irkutsk State University, 1, K. Marks St., Irkutsk, 664003 Phone: (3952)242210 (kazakov@baikal.ru)

Andrei Mantsivoda, Irkutsk State University, 1, K. Marks St., Irkutsk, 664003 Phone: (3952)242210 (andrei@baikal.ru)