Серия «Математика»
2011. Т. 4, № 1. С. 57—72

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

ИЗВЕСТИЯ
Иркутского государственного университета

УДК 510.62:004.82

Объектно-ориентированная дескриптивная логика *

А. А. Малых, А. В. Манцивода
Иркутский государственный университет

Аннотация. В работе развиваются логические формализмы, связанные с объектно–ориентированным моделированием. В частности, анализируется опыт работы с дескриптивными логиками, ориентированными на построение объектных моделей логическими средствами (т.н. ОО–проекциями), и рассматриваются вопросы дальнейшего развития этих формализмов. Вводится понятие объектной теории как расширение определения ОО–проекций.

Ключевые слова: дескриптивная логика, объектная модель, ОО–проекция, объектная теория, база знаний, язык программирования Libretto

1. Введение

ОО–проекции [2] представляют собой простую дескриптивную логику [8], моделирующую структуру хранилища данных и знаний в рамках системы Libretto [3]. Эта система базируется на концепции логических архитектур [2]. Логические архитектуры состоят из набора вложенных друг в друга логик, каждая из которых обладает своим соотношением эффективность/выразительность и моделирует определенный набор функций системы. В частности, система Libretto обладает двухуровневой логической архитектурой, в которой легкая логика, моделирующая концепцию объектных моделей (ОО–проекция), определяет структуру хранилища данных и знаний, а более тяжелая, но выразительная логика ($\mathcal{SHOIN}(D)$ [5]) лежит в основе языка запросов (язык Libretto), а также позволяет накладывать более сложные ограничения (аксиомы) на базы знаний.

* Работа выполнена при финансовой поддержке Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009-2013 гг., государственный контракт № 16.740.11.0137 от 01.09.2010.

В процессе работы был накоплен список проблем, связанный с использованием ОО–проекций. Стало очевидным, что данный формализм нуждается в дальнейшей настройке для того, чтобы более адекватно и эффективно решать свои задачи. Данная работа посвящена развитию концепции ОО–проекции как логической модели хранилища данных и знаний.

2. Объектно-ориентированные проекции

В этом пункте формулируется базовое определение ОО–проекции. Пусть

$$\mathfrak{D} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_k \rangle$$

некоторая область данных с попарно непересекающимися непустыми типами данных $\mathcal{D}_1, \ldots, \mathcal{D}_k$. Примерами $\mathcal{D}_i$ могут служить целые и вещественные числа, строки, даты и т.д. В дальнейшем константы v, v_i будут обозначать элементы типов данных – все элементы будем считать выделенными. Обозначим $|\mathfrak{D}| = \bigcup_{j=1}^{k} \mathcal{D}_j$.

Словарь дескриптивной логики включает символы концептов, ролей, атрибутов и объектов. Концепты выделяют в множестве объектов предметной области совокупности объектов, обладающих определенными свойствами. В дальнейшем концепты обозначаются через B, C, E, причем B, C обозначают именованные (атомарные) концепты, а E – произвольные концепты (формулы). Роли (о-свойства) определяют отношения между объектами предметной области и обозначаются через R, R_i. Атрибуты (т-свойства), обозначаемые через P, P_i, привязывают элементы области данных к объектам. Через O, O_i обозначаем имена объектов предметной области. Словарь – это четверка $\langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$, где $\mathcal{C}$ – множество именованных концептов, $\mathcal{R}$ – множество ролей, $\mathcal{P}$ – множество атрибутов, $\mathcal{O}$ – множество имен объектов. Область данных $\mathfrak{D}$ считается фиксированной, поэтому ее компоненты в словари не входят.

ОО–проекции – это подлогики выразительной дескриптивной логики $\mathcal{SHOIN}(D)$, лежащей в основе языка семантического веба OWL DL [9]. Основные структуры и семантика $\mathcal{SHOIN}(D)$ представлены на рис 1. В таблице на этом рисунке F могут обозначать как произвольные концепты E_i, так и имена типов данных D_i, а T – могут быть как ролями R_i, так и атрибутами P_i.

Описание предметной области в $\mathcal{SHOIN}(D)$, как и в любой другой дескриптивной логике, состоит из двух блоков - A-бокса и T-бокса. TBox содержит описание знаний концептуального (терминологическо-

Концепт	Пример	Семантика	
C	$Woman$	$C(x)$	имен. концепт
$E_1 \sqcap E_2$	$Woman \sqcap Manager$	$E_1(x) \wedge E_2(x)$	конъюнкция
$E_1 \sqcup E_2$	$Woman \sqcup Man$	$E_1(x) \vee E_2(x)$	дизъюнкция
$\neg C$	$\neg Man$	$\neg C(x)$	отр. атом. концептов
$\{o\}$	$\{john\}$	$x = o$	объекты
$\{v\}$	$\{29\}$	$x = v$	значения типов
$\forall T.F$	$\forall hasChild.Woman$	$\forall y.(T(x,y) \to F(y))$	огр. $\forall$
$\exists T.F$	$\exists Age.\{29\}$	$\exists y.(T(x,y) \wedge F(y))$	огр. $\exists$
$\forall T^{-}.F$	$\forall hasChild^{-}.Man$	$\forall y.(T(y,x) \to F(y))$	обрат. $\forall$
$\exists T^{-}.F$	$\exists hasChild^{-}.Man$	$\exists y.(T(y,x) \wedge F(y))$	обрат. $\exists$
$\geq n.T$	$\geq 1.hasChild$	$\exists y^{\geq n}.T(x,y)$	min–кардинальность
$\leq n.T$	$\leq 2.hasChild$	$\exists y^{\leq n}.T(x,y)$	max–кардинальность
$\top$	$\exists hasChild.\top$	true	все объекты
$\bot$	$\forall hasChild.\bot$	false	пусто

Рис. 1. Логика $\mathcal{SHOIN}(D)$

го) уровня и состоит из аксиом включения:

$$\begin{aligned} E_1 \sqsubseteq E_2, &\quad \forall x.(E_1(x) \to E_2(x)) \\ R_1 \sqsubseteq R_2, &\quad \forall x \forall y.(R_1(x,y) \to R_2(x,y)) \\ P_1 \sqsubseteq P_2, &\quad \forall x \forall v.(P_1(x,v) \to P_2(x,v)) \end{aligned}$$

$E_1 \equiv E_2$ – сокращенная запись двух аксиом ($E_1 \sqsubseteq E_2$) и ($E_2 \sqsubseteq E_1$). ABox описывает предметную область на уровне конкретных данных. А-бокс может содержать аксиомы следующих типов: $C(O)$ – принадлежность объекта концепту, $R(O_1, O_2)$ – отношение между объектами, $P(O, v)$ – значение атрибута объекта для $C \in \mathcal{C}$, $R \in \mathcal{R}$ и $P \in \mathcal{P}$.

Обратимся теперь к определению ОО–проекций. Язык ОО–проекций включает атомарные концепты C_i, роли R_i, конъюнкцию $\sqcap$ и квантор всеобщности $\forall R.C$, где C – атомарный концепт. Вместе с ролями R_i мы можем определять атрибуты объектов P_i, которые привязывают к объектам значения типов данных из $\mathcal{D}$, например, $\forall P_i.D_j$. Здесь $D_j, j \in \overline{1,k}$, – имена типов данных из $\mathcal{D}$. Также у нас имеются две логические константы $\top$ (универсальный концепт) и $\bot$ (нижний концепт). ОО–проекции – это дескриптивные логики, поэтому любая интерпретация I этих конструкций на множестве объектов Δ должна удовлетворять правилам:

$$\begin{array}{rl} \top^I = \Delta & \quad \bot^I = \emptyset \\ D_i^I = \mathcal{D}_i \in \mathcal{D} & \quad C_i^I \subseteq \Delta \\ R_i^I \subseteq \Delta \times \Delta & \quad P_i^I \subseteq \Delta \times |\mathcal{D}| \end{array}$$

$$(F \sqcap G)^I = F^I \cap G^I$$
$$(\forall R.C)^I = \{o \mid o \in \Delta \text{ and } \forall o'.(R(o, o') \to C(o'))\}$$
$$(\forall P.D_i)^I = \{o \mid o \in \Delta \text{ and } \forall v.(R(o, v) \to v \in \mathcal{D}_i)\}$$

Для типизации атомарных ролей и атрибутов вводится оператор типизации θ, который:

1. привязывает роли $R \in \mathcal{R}$ к парам $\theta(R) = [C_d, C_r]$, где $C_d, C_r \in \mathcal{C}$;
2. привязывает атрибуты P к парам $\theta(P) = [C_d, D_r]$, где $C_d \in \mathcal{C}$ и $D_r \in \mathcal{D}$.

C_d называется областью определения роли R (атрибута P), C_r (D_r) называется областью значений роли R (атрибута P).

Будем говорить, что интерпретация I согласована с θ, если

$$\begin{array}{l} \forall R \in \mathcal{R}.(\theta(R) = [C_d, C_r] \ \to \ R^I \subseteq C_d^I \times C_r^I) \\ \forall P \in \mathcal{P}.(\theta(P) = [C_d, D_r] \ \to \ P^I \subseteq C_d^I \times D_r^I) \end{array} \tag{2.1}$$

Интерпретация I согласована со словарем $\langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$, если для любого объекта $o \in \Delta$ существует имя $O \in \mathcal{O}$ такое, что $o = O^I$ (т.е. все объекты выделенные). В дальнейшем будем рассматривать только интерпретации, согласованные с θ и словарем. $\mathcal{K} \models E$ будет обозначать, что E истинна в любой согласованной интерпретации $\mathcal{K}$. В отличие от этого $\mathcal{K} \vdash E$ означает, что E выводима из $\mathcal{K}$ в некотором полном исчислении (например, с помощью табличных алгоритмов), а значит, истинна в любой интерпретации.

Пусть $\mathcal{W} = \langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$ – словарь.

Определение 1 (ОО-концепт)**.** *Концепт вида*

$$C_1 \sqcap \ldots \sqcap C_f \sqcap \forall R_1.E_1 \sqcap \ldots \sqcap \forall R_g.E_g \sqcap \forall P_1.D_1 \sqcap \ldots \sqcap \forall P_h.D_h$$

называется ОО-концептом, где E_i – ОО-концепты, $C, C_i \in \mathcal{C}$, $R_i \in \mathcal{R}$, $P_i \in \mathcal{P}$ и $\mathcal{D}_i \in \mathcal{D}$. Если все $E_i \in \mathcal{C}$, то ОО–концепт называется примитивным.

Определение 2 (ОО-определение)**.** *Аксиома*

$$C \sqsubseteq E$$

где E – примитивный ОО-концепт, называется объектно-ориентированным определением (ОО–определением).

В ОО-определении

$$C \sqsubseteq C_1 \sqcap \ldots \sqcap C_n \sqcap \forall R_1.B_1 \sqcap \ldots \sqcap \forall R_k.B_k \sqcap \forall P_1.D_1 \sqcap \ldots \sqcap \forall P_l.D_l \tag{2.2}$$

атомарные концепты C, C_i, B_i играют роль объектно ориентированных классов, $C_1 \sqcap \ldots \sqcap C_n$ обозначает наследование, и если $n > 1$, тогда мы имеем множественное наследование. Каждое выражение $\forall R_i.B_i$ обозначает поле R_i, значения которого являются объектами класса B_i. Каждое выражение $\forall P_i.D_i$ соответствует полю, значения которого являются элементами предопределенного типа данных с именем D_i.

Определение 3 (наследование)**.** *Концепт C в ОО–определении (2.2)* непосредственно наследует *от $C_i, i \in \overline{1,n}$, (обозначаем $C \prec C_i$).*

Пусть $\mathcal{A} = \{C_i \sqsubseteq E_i\}$ – множество ОО–определений. C_i наследует *от C_j ($C_i \prec^* C_j$), если либо $C_i \prec C_j$, либо существует C_l такой, что $C_i \prec^* C_l$ и $C_l \prec^* C_j$. $\mathcal{A}$ называется* ацикличным*, если для каждого C_i, выполняется $C_i \not\prec^* C_i$.*

Схожим образом, если концепт C определен с помощью ОО–определения (2.2) в множестве ОО-определений $\mathcal{A}$, то мы говорим, что концепт C *непосредственно зависит от* ролей $R_1, \ldots, R_k$ и атрибутов $P_1, \ldots, P_l$. C *зависит* от роли R (атрибута P), если он непосредственно зависит от роли (атрибута), или существует C' в $\mathcal{A}$, который непосредственно зависит от R (P), и $C \prec C'$.

Теперь обратимся к определению ОО–проекции. Пусть $\mathcal{L}$ – дескриптивная логика, содержащая $\mathcal{SHOIN}(D)$ как подлогику, и $\mathcal{K} = TBox_{\mathcal{K}} \cup ABox_{\mathcal{K}}$ – $\mathcal{L}$–описание предметной области в словаре $\mathcal{W} = \langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$. Пусть $\mathcal{W}_o = \langle \mathcal{C}_o, \mathcal{R}_o, \mathcal{P}_o, \mathcal{O}_o \rangle$ – подсловарь $\mathcal{W}$, $\mathcal{O}_o = \{O_1, \ldots, O_l\}$, а θ – оператор типизации на $\mathcal{W}_o$.

Определение 4 (ОО–проекция)**.**

$$\mathcal{K}_o = TBox_o \cup ABox_o$$

– ОО-проекция словаря $\mathcal{W}_o$ и оператора θ, если

1. *$TBox_o$ – ацикличное множество ОО–определений, любая интерпретация которого согласована с θ;*
2. *для каждого ОО–определения $A \in TBox_o$ выполняется $\mathcal{K} \vdash A$;*
3. *любая роль и атрибут из словаря $\mathcal{W}_o$ входит в ОО–определения $TBox_o$ не более одного раза;*
4. *$ABox_o \subseteq ABox_{\mathcal{K}}$ и*
   - *для любого $C(O) \in ABox_o$: $O \in \mathcal{O}_o$ и $C \in \mathcal{C}_o$, причем для каждого O такой C является единственным;*
   - *для любого $R(O, O') \in ABox_o$: $O, O' \in \mathcal{O}_o$, $R \in \mathcal{R}_o$ такие, что $C(O), C'(O') \in ABox_o$, причем $\mathcal{K}_o \models C \sqsubseteq C_d$, $\mathcal{K}_o \models C' \sqsubseteq C_r$ и $\theta(R) = [C_d, C_r]$;*
   - *для любого $P(O, v) \in ABox_o$: $O \in \mathcal{O}_o$, $P \in \mathcal{P}_o$, $C(O) \in ABox_o$ и $v \in \mathcal{D}_r$, где $\theta(P) = [C_d, D_r]$, и $\mathcal{K}_o \models C \sqsubseteq C_d$.*

Определение ОО–проекций настроено на использование в рамках логических архитектур. Являясь очень легким формализмом, ОО–проекции исполняют роль наименьшего элемента в иерархии логик логической архитектуры. Поэтому определение ОО–проекции содержит "большую" логику $\mathcal{L}$ – наибольший элемент иерархии, подлогикой которого ОО–проекция должна быть. В этой терминологии $\mathcal{K}_0$ – маленькая но эффективная объектная модель, являющаяся подмоделью большой, но, возможно, тяжелой модели $\mathcal{K}$.

В [2] было показано, что иерархии концептов ОО–проекций изоморфно отображаются в иерархии типов данных некоторого объектно-ориентированного языка программирования с множественным наследованием, что обосновывает гипотезу о том, что ОО–проекции ведут себя как объектные модели.

3. Критика ОО-проекций

ОО–проекции оказались весьма полезным инструментом, позволившим построить логически корректное понятие объектных моделей, а также логическую семантику ядра языка Libretto. Однако дальнейший опыт работы с ОО–проекциями показал, что эти логики обладают рядом недостатков, которые усложняют их использование. По большому счету ОО–проекции обладают тремя крупными проблемами, которые существенно влияют на точность логического представления объектных моделей:

1. В определении ОО–проекций имеются концептуальные изъяны, снижающие качество представления объектных моделей.
2. В ОО–проекциях слабо представлена возможность управления значениями свойств у объектов.
3. В ОО–проекциях отсутствует возможность определения инверсных свойств.

Эти проблемы носят не только концептуальный, но и практический характер, поскольку управление значениями свойств и инверсные отношения реализованы в системе Libretto. Поэтому отсутствие этих средств в ОО–проекциях не позволяет строить на их основе полноценные семантики языка.

Рассмотрим проблемы более подробно.

Проблема 1.1. Начнем с вопросов, связанных с декларированием доменов и рангов свойств. Оператор θ используется в ОО–проекциях для задания домена и ранга как ролей, так и атрибутов. Но ОО–определения тоже позволяют задавать домены и ранги. Например, в определении (2.2) наличие подформулы $\forall R_i.B_i$ содержательно интерпретируется как "для объектов класса C рангом свойства R_i является класс B_i". Таким образом, получается, что для объектов класса C мы имеем двойное определение ранга, и результирующее ограничение должно выглядеть как $C_r \cap B_i$, где $\theta(R_i) = [C_d, C_r]$. То же самое происходит и с областью определения R_i: для объектов класса C имеем область определения $C \cap C_d$. Возможность задавать ранг свойств была введена в ОО–

определения с целью повышения гибкости и выразительности формализма, однако дальнейшее использование ОО–проекций показало, что эти средства мало что дают, принося при этом ряд проблем. Среди других неприятностей оказалось, что эти средства очень легко приводят к нарушению одного из базовых принципов объектного подхода – принципа подстановки Б. Лисков [6].

Проблема 1.2. Сам оператор θ составляет еще одну концептуальную проблему. Поскольку θ не является частью логики ОО–проекций и представляет собой набор внешних ограничений на роли и атрибуты, это усложняет весь формализм в целом и затрудняет использование ОО–проекций, в частности, для описания семантики языка Libretto. Для решения этой проблемы данные ограничения должны быть выражены формульно внутри самой логики (например, в виде набора дополнительных аксиом). Однако слабая выразительная сила ОО–проекций не позволяет это сделать. Очевидным образом этот недостаток снижает эффективность ОО–проекций при решении задач, связанных с обработкой знаний в рамках объектных моделей. Например, используя только язык ОО–проекций невозможно определить, что есть объектная модель.

Проблема 2. Существенной проблемой ОО–проекций является то, что они не обладают средствами для управления количеством значений свойств объекта. Пусть объект с именем o имеет свойство T (неважно, роль или атрибут), значениями которого являются некоторые $t_1, \ldots, t_k$. Это означает, что в *ABox* ОО–проекции должны содержаться аксиомы

$$T(o, t_1), \ldots, T(o, t_k)$$

Язык ОО–проекций позволяет

- моделировать выделенное значение `null` (`NULL` в базах данных и т.д.) – это случай, когда $k = 0$;
- моделировать множественные значения (абстрактный аналог множеств, списков, массивов) – случай $k > 1$.

Однако выразительности ОО–проекций не хватает для того, чтобы отличить поля с единственным значением и поля, содержащие коллекцию элементов. Например, в Java легко отличить переменные, содержащие единственное значение, от переменных, содержащих массив значений:

```
Object single;
Object[] array = new Object[115];
```

В ОО–проекциях эти случаи неотличимы – и в том, и в другом соответствующее ОО–определение будет содержать подформулу:

$$\begin{array}{l}\forall \mathtt{single.Object} \\ \forall \mathtt{array.Object}\end{array} \tag{3.1}$$

С содержательной точки зрения невозможность управлять количеством значений также серьезно обедняет выразительные возможности ОО–проекций при построении баз знаний. Например, юридически у человека должна быть ровно одна фамилия, но это ограничение в ОО–проекциях не выразимо.

Проблема 3. В ОО–проекциях отсутствуют инверсные роли и атрибуты. Инверсные свойства не имеют прямых аналогов в объектных моделях, однако они позволяют формульно выразить ряд важных характеристик. Кроме того, они являются выразительным средством, реализованным в системе Libretto и весьма полезным на практике.

Проблема 1.1 может быть решена с помощью консервативной корректировки определения ОО–проекций, в то время, как решение проблем 1.2, 2 и 3 требует расширения их языка.

4. Корректировка определения ОО–проекций

Прежде чем переходить к расширению формализма, уточним поведение ОО–проекций при определении домена и ранга свойств, решив тем самым первую часть проблемы 1. Как уже говорилось выше, основная неприятность заключается во вхождении в ОО–определения вида (2.2) подформул $\forall R.B_i$ и $\forall P.D_i$, которые вовлекают B_i и D_i в определение ранга, а C в определение домена ролей R_i и атрибутов P_i. Решение этой проблемы состоит в исключении возможности дублирования при определении доменов и рангов. Для этого заменим определение 2 на следующее:

Определение 5. *Назовем ОО–определением аксиому вида*

$$C \sqsubseteq C_1 \sqcap \ldots \sqcap C_n, \tag{4.1}$$

где $C, C_1, \ldots, C_n \in \mathcal{C}$.

После этой замены только оператор θ будет отвечать в ОО–проекции за декларирование домена и ранга свойств. Отметим, что (4.1) является частным случаем (2.2), поэтому перестройка ОО–определения лишь ограничивает множество допустимых ОО–проекций. С другой стороны, данная корректировка позволяет нам разделить инструменты, ответственные за определение иерархии наследования (ОО–определения) и структуры, ответственные за декларирование доменов и рангов свойств (оператор θ). С этого момента определение 2 упраздняется, и вместо него применяется определение 5.

5. Определение логики $\mathcal{OODL}$

Выразительные средства, необходимые для решения проблем 1–3, содержатся в логике $\mathcal{SHOIN}(D)$. Поэтому все, что нам нужно, это аккуратно расширить откорректированное в предыдущем пункте понятие ОО–проекции в рамках $\mathcal{SHOIN}(D)$. Очевидно, что добавлений должно быть минимальное количество – для того, чтобы в наименьшей степени задеть алгоритмическую легкость ОО–проекций как минимального элемента иерархии логик в логической архитектуре.

Определим дескриптивную логику $\mathcal{OODL}$ как модернизацию ОО–проекций. Язык $\mathcal{OODL}$ включает язык ОО–проекций за исключением конъюнкции $\sqcap$: имена концептов C_i, имена ролей R_i, имена доменов D_i, имена объектов O_i, логические константы $\top$ и $\bot$ и квантор всеобщности $\forall T.F$, где T – роль или атрибут, а F – атомарный концепт, либо имя типа данных из $\mathcal{D}$. Расширим этот набор

- инверсными ролями при кванторах всеобщности: $\forall T^{-}.F$, где T – либо роль, либо атрибут, а F – либо атомарный концепт, либо имя типа данных;
- кардинальностями $\geq n.T$ и $\leq n.T$, где $n \in \mathbb{N}_0$, а T – либо роль, либо атрибут, либо инверсия роли/атрибута.

Семантика инверсных свойств и кардинальностей определена на рис. 1.

Добавление данных конструкций существенно усиливает выразительные возможности. Например, с помощью кардинальности можно выразить неограниченный квантор существования $\exists$:

$$\exists R \stackrel{\text{def}}{=} \geq 1.R$$

Данный концепт выделяет все объекты, у которых существуют значения свойства R. Он эквивалентен выражению $\exists R.\top$ в логике $\mathcal{SHOIN}(D)$.

6. Решение проблем ОО–проекций

Обогатив язык ОО–проекций, возвратимся к решению проблем 1–3, начав с проблемы 2.

Решение проблемы 2. Эта проблема решается с помощью минимальной и максимальной кардинальности. Например, теперь вместо (3.1) мы можем записать

$$\begin{array}{lll} \forall\mathtt{single.Object} & \leq 1.\mathtt{single} & \geq 1.\mathtt{single} \\ \forall\mathtt{array.Object} & \leq 115.\mathtt{array} & \geq 115.\mathtt{array} \end{array} \tag{6.1}$$

Введем сокращение пары аксиом

$$= n.T \quad \stackrel{\text{def}}{=} \quad \{\leq n.T, \quad \geq n.T\}$$

которое позволяет выразить мысль о том, что свойство T должно иметь ровно n значений. Кардинальности позволяют определять и допустимые отрезки значений, например, то, что свойство имеет не меньше 10 и не больше 100 значений:

$$\geq 10.T \qquad \leq 100.T.$$

Решение проблемы 3. Добавленные инверсные роли в сочетании с квантором всеобщности и кардинальностями позволяют характеризовать не только область значений, но и область определений свойств. Например, теперь мы можем выделить элементы, принадлежащие области значений свойства R:

$$\geq 1.\ R^{-}$$

что с использованием сокращения записывается как $\exists R^{-}$. Появившиеся выразительные возможности помогут нам решить следующую проблему.

Проблема 1.2. В пункте 4 мы уже решили первую часть проблемы 1, откорректировав понятие ОО–определения, поэтому нам осталось избавиться от оператора θ. Верно следующее утверждение:

Лемма 1. *Условия оператора θ ОО–проекций формульно выразимы на языке логики $\mathcal{OODL}$.*

Доказательство. Пусть $\mathcal{K}$ – ОО–проекция в словаре $\mathcal{W} = \langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$ с оператором θ, и для некоторого $R \in \mathcal{R}$ выполняется

$$\theta(R) = [C_d, C_r]$$

где $C_d, C_r \in \mathcal{C}$. Тогда в соответствии с (2.1) на интерпретациях, согласованных с θ, должна выполняться

$$\forall R \in \mathcal{R}.(\theta(R) = [C_d, C_r] \ \rightarrow \ R^I \subseteq C_d^I \times C_r^I)$$

Используя выразительные средства $\mathcal{OODL}$, мы можем определить эти требования формульно. Множество всех объектов, обладающих свойством R, выделяется с помощью концепта $\exists R$. Множество всех значений свойства R выделяется с помощью концепта $\exists R^{-}$ (напомним, что $\exists R$ есть сокращение для формулы $\geq 1.R$). Тогда оператор θ для свойства R определяется с помощью двух аксиом:

$$\begin{array}{l} \exists R \sqsubseteq C_d \\ \exists R^{-} \sqsubseteq C_r \end{array} \tag{6.2}$$

Для атрибутов вторая аксиома будет выглядеть $\exists P^{-} \sqsubseteq D_i$, где D_i – имя типа данных. □

Назовем первую и вторую аксиомы из (6.2) *аксиомой домена* и *аксиомой ранга*, соответственно. Пусть $\mathbb{R}_\theta$ – множество аксиом домена и ранга, описывающих оператор θ для каждого $T \in (\mathcal{R} \cup \mathcal{P})$.

Лемма 2. *Интерпретация I согласована с θ тогда и только тогда, когда $I \models \mathbb{R}_\theta$.*

Доказательство. Лемма доказывается непосредственной проверкой на основе определений оператора θ и аксиом домена и ранга с учетом леммы 1. □

Последние две леммы говорят о том, что нам для типизации свойств уже не нужен оператор θ, поскольку соответствующие ограничения на домен и ранг свойств выразимы на языке $\mathcal{OODL}$. Проблема 1.2 решена. Таким образом переход на логику $\mathcal{OODL}$ позволяет решить все три проблемы ОО–проекций.

7. Выразимость аксиом домена и ранга

Следует отметить, что аксиома ранга выразима на языке ОО–проекций, о чем говорит следующая лемма.

Лемма 3. *Аксиомы $\exists T^- \sqsubseteq C_r$ и $\top \sqsubseteq \forall T.C_r$ эквивалентны.*

Доказательство. Рассмотрим утверждение для ролей. Случай атрибутов рассматривается аналогично.

Необходимость. Предположим, что на некоторой интерпретации I вторая аксиома не верна. Тогда по определению квантора всеобщности существуют объекты o и o' такие, что

$$I \models T(o, o') \ \wedge \ \neg C_r(o')$$

По определению инверсного свойства $I \models T^-(o', o)$. Отсюда следует, что o' принадлежит концепту $\exists T^-$, и поскольку $I \models \neg C_r(o')$, то первая аксиома также не верна.

Достаточность доказывается в том же стиле. Пусть первая аксиома неверна. Тогда существует объект o' такой, что для некоторого o выполняется $T^-(o', o)$ и не выполняется $C_r(o')$, то есть $I \models \neg C_r(o')$. Теперь ложность второй аксиомы получаем по определению инверсного свойства и квантора всеобщности. □

Выражение $\top \sqsubseteq \forall T.C_r$ из формулировки леммы – это аксиома языка ОО–проекций, эквивалентная аксиоме ранга. К сожалению, в отличие от аксиомы ранга, аксиома домена на языке ОО–проекций не выразима. Минимальным расширением языка ОО–проекций, достаточным для того, чтобы избавиться от оператора θ является добавление min–кардинальности:

Лемма 4. *Условия оператора θ формульно выразимы на языке ОО–проекций, расширенном оператором min–кардинальности.*

Доказательство. Аксиома домена формулируется с помощью min–кардинальности, в то время, как аксиома ранга формульно выразима в ОО–проекции на основании леммы 3 $\square$

Конечно, аналогичный эффект можно получить с помощью других расширений. Для иллюстрации приведем пример альтернативного расширения, позволяющего по–иному выразить аксиому домена: вместо min–кардинальности добавим дизъюнкцию. Тогда аксиома примет вид:

$$\top \sqsubseteq \forall T.\bot \sqcup C_d$$

Для нас же важно, что мы смогли выразить аксиому домена в качестве побочного эффекта за счет средств, уже добавленных в ОО–проекцию при решении других проблем, избежав таким образом излишних усложнений формализма.

8. Погружение ОО–проекций в $\mathcal{OODL}$

В предыдущих пунктах нами выделено пять схем аксиом, описывающих объектные модели в рамках логики $\mathcal{OODL}$. Это

$$\begin{array}{ll} C_1 \sqsubseteq C_2 & \text{(аксиома наследования)} \\ \exists T \sqsubseteq C_d & \text{(аксиома домена)} \\ \exists T^- \sqsubseteq C_r & \text{(аксиома ранга)} \\ C_d \sqsubseteq\ \geq n.T & \text{(аксиома min-кардинальности)} \\ C_d \sqsubseteq\ \leq n.T & \text{(аксиома max-кардинальности)} \end{array}$$

Определение 6. *Объектной теорией $\mathbb{O}$ логики $\mathcal{OODL}$ в словаре $\mathcal{W} = \langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$ назовем непротиворечивую теорию, $TBox$ которой состоит из конечного множества аксиом наследования, домена, ранга, max– и min–кардинальности, удовлетворяющую следующим условиям:*

1. *множество аксиом наследования в $\mathbb{O}$ ациклично;*
2. *каждое свойство $T \in (\mathcal{R} \cup \mathcal{P})$ обладает в $\mathbb{O}$ ровно одной аксиомой домена и ровно одной аксиомой ранга.*

Заметим, что $TBox$ любой объектной теории без аксиом кардинальности непротиворечив, то есть имеет хотя бы одну интерпретацию. В качестве такой интерпретации можно выбрать одноэлементное множество $\Delta = \{o\}$, определив для всех $C \in \mathcal{C}$: $C^I = \Delta$, и всех $T \in (\mathcal{R} \cup \mathcal{P})$: $T^I = \emptyset$. Поэтому требование непротиворечивости накладывает условия только на аксиомы кардинальности и $ABox$ объектной теории, согласовывая его с ограничениями, содержащимися в $TBox$.

Прежде чем перейти к основному результату, сформулируем очевидную лемму:

Лемма 5. *В любой ОО–проекции ОО–определение вида (4.1) эквивалентно некоторому набору аксиом наследования.*

Доказательство. По свойствам логических связок аксиома вида (4.1) эквивалентным образом может быть представлена в виде набора аксиом наследования $C \sqsubseteq C_i, i \in \overline{1,n}$. □

Сформулируем основной результат данной работы.

Теорема 1. *Для любой непротиворечивой ОО–проекции $\mathcal{K}$ с оператором θ в словаре $\mathcal{W} = \langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$ существует эквивалентная ей объектная теория $\mathbb{O}$ логики $\mathcal{OODL}$.*

Доказательство. Пусть $\mathcal{K}$ – ОО–проекция в словаре $\mathcal{W}$. Построим соответствующую ей объектную теорию $\mathbb{O}$ следующим образом:

1. Для каждого ОО–определения вида (4.1) добавляем в $\mathbb{O}$ n аксиом
$$C \sqsubseteq C_1, \ldots, C \sqsubseteq C_n$$
определяющих наследование класса C от классов $C_1, \ldots, C_n$.
2. Для каждой роли $R \in \mathcal{R}$, $\theta(R) = [C_d, C_r]$, добавляем в $\mathbb{O}$ по одной аксиоме домена и ранга:
$$\exists R \sqsubseteq C_d \quad \exists R^- \sqsubseteq C_r$$
3. Для каждого атрибута $P \in \mathcal{P}$, $\theta(P) = [C_d, D_r]$, добавляем в $\mathbb{O}$ по одной аксиоме домена и ранга:
$$\exists P \sqsubseteq C_d \quad \exists P^- \sqsubseteq D_r$$
4. В качестве *ABox* теории $\mathbb{O}$ берем *ABox* ОО–проекции $\mathcal{K}$.

Прежде всего отметим, что $\mathbb{O}$ не содержит аксиом кардинальности. В первую очередь нам необходимо доказать, что получившееся множество $\mathbb{O}$ действительно является объектной теорией. Для этого, по определению, нужна ацикличность аксиом наследования – но она напрямую следует из ацикличности $\mathcal{K}$. Второе условие определения также выполняется, поскольку по пп. 2 и 3 построения $\mathbb{O}$ каждое свойство обладает ровно одной аксиомой домена и одной аксиомой ранга. Осталось проверить непротиворечивость. Для этого докажем, что произвольная интерпретация I, $I \models \mathcal{K}$, является интерпретацией и для $\mathbb{O}$ (по условию теоремы $\mathcal{K}$ – непротиворечивая теория, поэтому как минимум одна такая интерпретация существует). Поскольку А–боксы в $\mathcal{K}$ и $\mathbb{O}$ совпадают (по п. 4 построения), нам достаточно проверить, что каждая аксиома наследования, домена и ранга из $\mathbb{O}$ выполняется на I. Аксиомы наследования выполняются по лемме 5 и по п. 1 построения $\mathbb{O}$. Истинность аксиом домена и ранга следует из семантики оператора θ и леммы

2. По определению интерпретации ОО–проекций, любая интерпретация должна быть согласована с этим оператором. Таким образом, мы показали

$$I \models \mathcal{K}_0 \Rightarrow I \models \mathbb{O}$$

для любой I, что не только доказывает, что $\mathbb{O}$ – объектная теория, но и является необходимым условием теоремы.

Достаточность. Предположим, что $I \models \mathbb{O}$. Покажем, что I согласована с оператором θ. Пусть для некоторого $R \in \mathcal{R}$ $\theta(R) = [C_d, C_r]$. Тогда по п.2 построения $\exists R \sqsubseteq C_d \in \mathbb{O}$ и $\exists R^- \sqsubseteq C_r \in \mathbb{O}$, которые означают, что для любой пары $\langle o, o' \rangle \in R^I$ следует, что $o \in C_d^I$ по первой аксиоме и $o' \in C_r^I$ по второй и, таким образом, выполняется условие согласованности (2.1). Аналогично рассматривается случай с атрибутами. Истинность на I ОО–определений $\mathcal{K}$ напрямую вытекает из леммы 5. Теперь утверждение теоремы следует из п. 4 построения $\mathbb{O}$, который говорит о том, что $ABox_{\mathbb{O}} = ABox_{\mathcal{K}}$. $\square$

9. Объектные теории и язык Libretto

Продемонстрируем то, как соотносятся механизмы описания моделей в виде объектных теорий логики $\mathcal{OODL}$ с соответствующими конструкторами объектно–итерационного языка Libretto [7][3]. Например, определение на Libretto

```
class Человек extends Примат {
    супруг Человек[0,1];
    фамилия v:string[1,1]
}
```

которое задает класс `Человек`, обладающий ролью (о–свойством) `супруг` и атрибутом (т–свойством) `фамилия`, эквивалентно следующему набору аксиом объектной модели

$\exists\,\texttt{супруг} \sqsubseteq \texttt{Человек}$	$\texttt{Человек} \sqsubseteq \texttt{Примат}$
$\exists\,\texttt{супруг}^- \sqsubseteq \texttt{Человек}$	$\texttt{Человек} \sqsubseteq\ \leq 1\,.\,\texttt{супруг}$
$\exists\texttt{фамилия} \sqsubseteq \texttt{Человек}$	$\texttt{Человек} \sqsubseteq\ = 1\,.\,\texttt{фамилия}$
$\exists\texttt{фамилия}^- \sqsubseteq \texttt{string}$	

Заметим, что Libretto позволяет определять все аксиомы по отдельности, за исключением аксиом кардинальности – кардинальность должна быть привязана к классам или типам данных. Например, предыдущее определение на Libretto эквивалентно следующей Libretto–программе:

```
class Человек extends Примат;
```

```
class Человек {
    супруг Человек[0,1]
}
class Человек {
    фамилия v:string[1,1]
}
```

Таким образом Libretto позволяет создавать базы знаний постепенно, через последовательное уточнение информации о классах и других сущностях описываемой предметной области. На практике такое качество оказывается весьма полезным.

10. Заключение

Определение логики $\mathcal{OODL}$ и понятия объектной теории, представленное в данное работе, решает две основные задачи:

1. С концептуальной точки зрения – построение логического формализма, наиболее точно отражающего содержательное понятие объектных моделей.
2. С практической точки зрения – приближение логического описания абстрактного хранилища данных к реализации системы OntoBox – хранилища онтологий, с которыми работает язык Libretto, что дает инструменты для описания точной денотационной семантики более широкого сегмента Libretto, чем это было представлено в работе [7].

Существует еще одна проблема, которая не затронута в данной работе – проблема упорядочения значений свойств. На практике в наибольшей степени используются структуры данных, элементы которых упорядочены – достаточно упомянуть массивы, списки и итераторы. В дескриптивных логиках в качестве коллекций значений выступают множества, поэтому введение в декскриптивные логики порядка принципиально усложняет как сам формализм, так и его алгоритмические качества. С нашей точки зрения, данную проблему лучше всего решать в рамках теории GES [1][4] с использованием наследственно–конечных списочных надстроек над моделями. Это решение является весьма кардинальным, поскольку означает переход от дескриптивных логик к специальным логикам первого порядка. Однако такой шаг представляется весьма перспективным, поскольку теория GES обеспечивает необходимые нам качества при сохранении гибкости в регулировании соотношения выразительность/эффективность.

Список литературы

1. Гончаров С. С. Σ–программирование / С. С. Гончаров, Д. И. Свириденко // Вычисл. системы. – Новосибирск, 1985. – Вып. 107. – С. 3–29.
2. 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.
3. Объектно-итерационный язык Libretto [Электронный ресурс]. – URL: http://ontobox.org.
4. 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.
5. 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.
6. Liskov B. A behavioral notion of subtyping / Barbara Liskov, Jeannette Wing // ACM Transactions on Programming Languages and Systems (TOPLAS). - 1994. – Vol. 16, Iss. 6. – P. 1811–1841.
7. Malykh A. A Query Language for Logic Architectures / A. Malykh, A. Mantsivoda // Perspectives of System Informatics : Proceedings of 7th International Conference. – Springer-Verlag Berlin Heidelberg, 2010. – Lecture Notes in Computer Science 5947. – P. 294-305.
8. The Description Logic Handbook: Theory, Implementation, Applications / F. Baader, D. Calvanese , D. L. McGuinness, D. Nardi, P.F. Patel-Schneider. - Cambridge, 2003. – P.574.
9. Web Ontology Language (OWL) [Electronic resource]. – URL: www.w3.org/2004/OWL.

A. V. Mantsivoda, A. A. Malykh

Object Oriented Description Logic

Abstract. In this paper logical formalisms connected to object–oriented modeling are considered. In particular, the experience of the application of description logics intended for logical object model development (so–called OO–projections) is analyzed, and the issues of the further development of the approach are presented and discussed. The notion of an object theory is introduced as an extension of OO–projections.

Малых Антон Александрович, Институт математики,экономики и информатики, Иркутский государственный университет, 664003, Иркутск, ул. К. Маркса, 1 тел.: (3952)242210 (`malykh@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`)

Anton Malykh, Irkutsk State University, 1, K. Marks St., Irkutsk, 664003 Phone: (3952)242210 (`malykh@baikal.ru`)

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