



УДК 004.8

ОДИН ЯЗЫК ДЛЯ СПЕЦИФИКАЦИИ ОНТОЛОГИЙ

Плесневич Г.С.

*Московский энергетический институт (Национальный исследовательский университет)
г. Москва, Россия
salve777@mail.ru*

Описывается некоторый язык для спецификации онтологий, входящий в систему «Бинарная Модель Знаний», которая предназначена для спецификации и интерпретации онтологий. Определен синтаксис, денотативная и операционная семантика этого языка. Рассматривается записанная в этом языке онтология, представляющая известную задачу Steamroller. Показано, как можно решить эту задачу, используя продукционный вывод.

Ключевые слова: искусственный интеллект, представление знаний, онтологии.

Введение

В работе [Плесневич, 2013] мы кратко описали систему «Бинарная Модель Знаний» (БМЗ) языков, предназначенных для спецификации онтологий. Представленные в БМЗ онтологии записываются с помощью языков структурной и логической спецификации понятий. Эти языки ориентированы на представление понятий.

Формальное понятие имеет имя C , универсум U^C и экстенционал Ext^C . Универсум U^C – это множество всех имен, которые могут обозначать примеры (экземпляры) понятия C . Экстенционал Ext^C есть пара (E^C, \sim^C) , где E^C – множество всех имен из U^C , которые обозначают примеры понятия (в данном контексте), а \sim^C – отношение эквивалентности на множестве E^C , определяющее *корреферентность* имен. (Два имени a и b корреферентны, если они обозначают один и тот же объект предметной области.)

Мы предполагаем, что универсум U^C любого понятия C включает множество Sur так называемых *суррогатов* #1, #2, #3, ... – системных идентификаторов для объектов моделируемых предметных областей. Мы также предполагаем, что для любого $x \in E^C$ существует единственный суррогат, корреферентный E^C .

В системе БМЗ имеется язык ЯСС структурной спецификации объектов, при помощи которого определяются универсумы понятий. Эти понятия в БМЗ являются либо *классами*, либо *бинарными связями*. Последние устанавливают отношения между объектами двух классов. компонентами которых служат выражения видов суррогаты.

Для спецификации экстенционалов понятий моделируемой предметной области применяются логические предложения. Эти предложения используют термины тех структурных предложений языка ЯСС, которые определяют универсумы этих понятий.

Онтология – это (конечное) множество O структурных и логических предложений. Таким образом, онтология O делится на две части $O = O_{стр} \cup O_{лог}$, где $O_{стр}$ состоит из структурных предложений, а $O_{лог}$ – из логических предложений.

В системе БМЗ языки логической спецификации составляют иерархию по своей выразительности. В настоящей работе мы опишем некоторый язык логической спецификации, названный ЯЛС-0, который является в некотором смысле минимальным в этой иерархии.

1. Задача Steamroller

Сначала запишем в языке ЯЛС-0 и в языке ЯСС известную задачу Steamroller (паровой каток). Эта задача была сформулирована в 1978 году Л. Шубертом (L. Schubert) и была им предложена в качестве тестовой задачи для программ логического вывода, основанных на методе резолюции (см., например, [Walther 2003]).

Задача Steamroller состоит в доказательстве, что из следующих утверждений (а) – (л) логически следует утверждение (м).

- (а) Волки, лисы, птицы, гусеницы и улитки – животные.
- (б) Существуют волки, лисы, птицы, гусеницы и улитки.
- (в) Злаки – растения.

- (г) Злаки существуют.
 (д) Каждое животное ест некоторые растения или ест всех животных, которые его меньше и которые едят некоторые растения.
 (е) Гусеницы меньше птиц.
 (ж) Птицы меньше лис.
 (з) Лисы меньше волков.
 (и) Волки не едят гусениц и не едят растений.
 (к) Птицы едят гусениц, но не едят улиток.
 (л) Гусеницы и улитки едят некоторые растения.
 (м) Существует животное, которое ест некоторых животных, питающихся злаками.

Знание, содержащееся в утверждениях (а) – (л), представим в следующей онтологии **O-St**, записанной в языке ЯСС U ЯЛС-0.

- O-St** = {
 1. (Животное Больше Животное)
 2. (Животное Ест Животное)
 3. (Животное Ест Растение)
 4. Волк, Лиса, Птица, Гусеница, Улитка ISA Животное
 5. EXIST Волк, Лиса, Птица, Гусеница, Улитка
 6. Злак ISA Растение
 7. EXIST Злак
 8. EACH Животное: X (ЕСТ SOME Растение OR Ест EACH Животное THAT (Меньше X) AND Ест SOME Растение)
 9. EACH Гусеница Больше EACH Птица
 10. EACH Птица Больше EACH Лиса
 11. EACH Лиса Больше EACH Волк
 12. EACH Волк -Ест EACH Гусеница, Растение
 13. EACH Птица (Ест EACH Гусеница AND - Ест EACH Улитка)
 14. EACH Гусеница, Улитка Ест SOME Растение}

Мы видим, что предложения 4 – 14 в точности отвечают утверждениям (а) – (л). Утверждение (м) в языке ЯЛС-0 записывается так:

15. EXIST Животное THAT Ест SOME Животное THAT Ест EACH Злак.

В онтологии **O-St** предложения 1, 2 и 3 принадлежат языку ЯСС. Они задают структуру бинарных связей Больше и Ест. Остальные понятия Животное, Растение, Волк, Лиса и т.д. не имеют структуры.

Решение задачи Steamroller состоит в установлении, что из онтологии **O-St** логически следует предложение 15.

2. Синтаксис и семантика языка ЯЛС-0

Предложения языка ЯЛС-0 строятся из термов трех видов: С-термы, L₂-термы и Р-термы.

С-термы обозначают классы объектов. Они интерпретируются как подмножества множества Surr суррогатов. Примеры С-термов, содержащихся в онтологии **O-St**: Животное, Животное THAT Ест SOME Растение.

L-термы обозначают бинарные связи. Они интерпретируются как подмножества множества (Surr, Surr) всех пар суррогатов. Примеры L-термов:

Меньше, Ест, -Ест, Ест SOME Растение, Меньше X AND Ест SOME Растение

Р-термы обозначают одноместные предикаты, заданные на множестве Surr суррогатов. Примеры Р-термов: Меньше X, Меньше EACH Волк.

При описании синтаксиса и семантики выражений языка ЯЛС-0 мы будем использовать переменные, имеющие следующие значения:

- *c, d* – индивидные константы (имена объектов);
- *C, D, E* – имена классов;
- *L, M, N* – имена бинарных связей;
- *P, Q, R* – имена предикатов;
- *Name* – простые (атомарные) имена классов, бинарных связей или предикатов;
- *S* – предложение языка ЯЛС-0;
- *V* – имена *X, X₁, X₂, ...*;
- *T* – имена **true** и **false** (или **1** и **0**), обозначающие истину и ложь.

Через «*Exp*», где *Exp* – какое-либо выражение, обозначим семантику этого выражения, т.е. его значение в данной интерпретации в универсуме Surr.

СИНТАКСИС С-ТЕРМОВ:

$C, D ::= -C \mid C \text{ AND } D \mid C \text{ OR } D \mid C ::= D \text{ THAT } P \mid L \text{ SOME } D \mid L \text{ ONLY } D \mid D : V \mid \text{Name}$

СЕМАНТИКА С-ТЕРМОВ:

« $-C$ » = $\text{Surr} \setminus \langle C \rangle$;
 « $C \text{ AND } D$ » = « C » \cap « D »;
 « $C \text{ OR } D$ » = « C » \cup « D »;
 « $L \text{ SOME } C$ » = $\{x \in \text{Surr} \mid (\exists y \in \langle C \rangle) (x, y) \in \langle L \rangle\}$;
 « $L \text{ ONLY } C$ » = $\{x \in \text{Surr} \mid (\forall y \in \text{Surr}) [(x, y) \in \langle L \rangle \rightarrow x \in \langle C \rangle]\}$;
 « $C \text{ THAT } P$ » = $\{x \in \langle C \rangle \mid \langle P \rangle(x)\}$;
 « $D : V$ » = « D »;

«*Name*» – множество всех строчек символов начинающихся с заглавных букв.

СИНТАКСИС L-ТЕРМОВ:

$L, M ::= -L \mid L \text{ AND } M \mid L \text{ OR } M \mid \text{INV}(L) \mid \text{Name}$

СЕМАНТИКА L-ТЕРМОВ:

« $-L$ » = $(\text{Surr}, \text{Surr}) \setminus \langle L \rangle$;
 « $C \text{ AND } D$ » = « C » \cap « D »;
 « $C \text{ OR } D$ » = « C » \cup « D »;
 « $\text{INV}(L)$ » = $\{(y, x) \mid (x, y) \in \langle L \rangle\}$.

СИНТАКСИС Р-ТЕРМОВ:

$P, Q ::= -P \mid P \text{ AND } Q \mid P \text{ OR } Q \mid L \text{ SOME } C \mid L \text{ EACH } C \mid (L V) \mid (L c) \mid \text{Name}$

СЕМАНТИКА Р-ТЕРМОВ:

$x \langle -P \rangle \Leftrightarrow \neg (x \langle P \rangle)$;
 $x \langle P \text{ AND } Q \rangle \Leftrightarrow x \langle P \rangle \wedge x \langle Q \rangle$;
 $x \langle P \text{ OR } Q \rangle \Leftrightarrow x \langle P \rangle \vee x \langle Q \rangle$;
 $x \langle L \text{ SOME } C \rangle \Leftrightarrow (\exists y \in \langle C \rangle) (x, y) \in \langle L \rangle$;
 $x \langle L \text{ EACH } C \rangle \Leftrightarrow (\forall y \in \text{Surr}) [(x, y) \in \langle L \rangle \rightarrow y \in \langle C \rangle]$;
 $x \langle (L V) \rangle \Leftrightarrow (x, V) \in \langle L \rangle$;
 $x \langle (L c) \rangle \Leftrightarrow (x, c) \in \langle L \rangle$.

Замечания. 1) Выражение $L \text{ SOME } D$ является С-термом или Р-термом в зависимости от контекста (положения в предложении). 2) Вместо обычной префиксной записи « P »(*x*) для результата

применения предиката « P » к объектной переменной x мы используем постфиксную запись $x \llbracket P \rrbracket$. 3) Символ операции AND (операции OR) используется как для обозначения конъюнкции (дизъюнкции) предикатов, так и для обозначения пересечения (объединения) классов.

СИНТАКСИС ПРЕДЛОЖЕНИЙ:

$S ::= -S \mid T \mid \text{EXIST } C \mid \text{EXIST } L \mid$
 $\text{NULL } C \mid \text{NULL } L \mid C \text{ ISA } D \mid C = D \mid \text{EACH } C P \mid$
 $\text{SOME } C P; \text{--} S = S.$

СЕМАНТИКА ПРЕДЛОЖЕНИЙ:

$\llbracket -S \rrbracket \Leftrightarrow \neg \llbracket S \rrbracket$; $\llbracket \text{EXIST } C \rrbracket \Leftrightarrow C \neq \emptyset$;
 $\llbracket \text{NULL } C \rrbracket \Leftrightarrow C = \emptyset$; $\llbracket C \text{ ISA } D \rrbracket \Leftrightarrow \llbracket C \rrbracket \subseteq \llbracket D \rrbracket$; $\llbracket C = D \rrbracket \Leftrightarrow \llbracket C \rrbracket = \llbracket D \rrbracket$; $\llbracket \text{EACH } C P \rrbracket \Leftrightarrow$
 $(\forall x \in \llbracket C \rrbracket) \llbracket P \rrbracket(x)$; $\llbracket \text{SOME } C P \rrbracket \Leftrightarrow$
 $(\exists x \in \llbracket C \rrbracket) \llbracket P \rrbracket(x).$

Следующие выражения мы рассматриваем как паттерны для получения примитивных предложений:

$C = -D \mid L = -M \mid \text{EXIST } C \mid \text{EXIST } L \mid$
 $\text{NULL } C \mid \text{NULL } L \mid C \text{ ISA } D \mid L \text{ ISA } M \mid C = D \mid$
 $L = M \mid E = C \text{ AND } D \mid E = C \text{ OR } D \mid N = L \text{ AND } M \mid N =$
 $L \text{ OR } M \mid C = D \text{ THAT } P \mid C = L \text{ SOME } D \mid$
 $C = L \text{ ONLY } D \mid M = \text{INV}(L) \mid Q = -P \mid R = P \text{ AND } Q \mid R =$
 $P \text{ OR } Q \mid P = L \text{ SOME } C \mid P = L \text{ EACH } C \mid$
 $P = (L \vee).$

Конкретное *примитивное предложение* получается путем подстановки в паттерн имен из *Name*.

Любое предложение языка ЯЛС-0 можно преобразовать в множество примитивных предложений. Покажем это на примере.

Пример 2.1. Возьмем предложение 15, т.е.

EXIST Животное THAT Ест SOME
 Животное THAT Ест EACH Злак,

и положим:

$P1 = \text{Ест EACH Злак},$
 $C1 = \text{Животное THAT Ест EACH Злак},$
 $P2 = \text{Ест SOME Животное THAT Ест}$
 $\text{EACH Злак},$
 $C2 = \text{Животное THAT Ест SOME}$
 $\text{Животное THAT Ест EACH Злак}.$

Тогда предложение 15 можно заменить на множество следующих примитивных предложений:

- $O_1 = \{$
1. $P1 = \text{Ест EACH Злак}$
 2. $C1 = \text{Животное THAT } P1$
 3. $P2 = \text{Ест SOME } C1$
 4. $C2 = \text{Животное THAT } P2$
 5. $\text{EXIST } C2\}.$

Множество O_1 примитивных предложений можно рассматривать как эквивалентное расширение предложения 15 в следующем смысле: любая модель для предложения 15 (т.е. интерпретация, при которой это предложение истинно) продолжается до модели множества O_1 и наоборот, сужение модели для O_1 на понятия, входящие в 15, является моделью этого предложения.

3. Продукционный вывод

Операционную семантику языка ЯЛС-0 мы определим в терминах продукций, действующих на базах фактах данной онтологии.

Мы дадим абстрактный синтаксис для языка продукций. Факты, относящиеся к классу C , будем записывать в виде $+a(C)$ и $-a(C)$, а факты, относящиеся к бинарной связи L , записывать в виде $+(c,d)(L)$ и $-(c,d)(L)$. Компоненты продукций также будем записывать в виде $+X(C)$, $-X(C)$, $+(X,Y)(L)$ и $-(X,Y)(L)$, где X и Y – переменные, используемые для сопоставления этих компонент с фактами.

Продукции действуют только на базах фактов для онтологий, составленных из примитивных предложений. Поэтому сначала произвольную онтологию нужно преобразовать в состоящую из примитивных предложений.

Обозначим через $O\text{-St}^*$ онтологию, полученную преобразованием всех предложений из $O\text{-St}$ в примитивные предложения. (Онтология $O\text{-St}^*$ состоит 36 примитивных предложений.)

С каждым из 24 паттернов для примитивных предложений мы связываем набор продукций. Укажем некоторые из них.

1. Паттерн $C = -D$.

Продукции:
 $+X(C) \Rightarrow -X(D)$
 $-X(C) \Rightarrow +X(D)$
 $+X(D) \Rightarrow -X(C)$
 $-X(D) \Rightarrow +X(C)$

Например, первая продукция действует так: в базе фактов для класса C разыскиваются факты вида $+a(C)$ и, если такой факт есть, то к базе фактов для D присоединяем факт $-a(C)$.

2. Паттерн $C \text{ ISA } D$.

Продукции:
 $+X(C) \Rightarrow +X(D)$
 $-X(D) \Rightarrow -X(C)$

3. Паттерн $\text{EXIST } C$.

Продукции:
 $\Rightarrow +c_j(C)$
 $+X(C) \Rightarrow \text{EXIST } C$

Здесь c_j – новая сколемовская константа. Таким образом, результатом применения первой продукции является присоединение факта $+c_j(C)$ к базе фактов для класса C .

4. Паттерн $E = C \text{ AND } D$.

Продукции:
 $+X(E) \Rightarrow +X(C); +X(D)$
 $-X(E); +X(C) \Rightarrow -X(D)$
 $-X(E); +X(D) \Rightarrow -X(C)$
 $-X(C) \Rightarrow -X(E)$
 $-X(D) \Rightarrow -X(E)$

Например, первая продукция действует так: в базе фактов для класса E разыскиваются факты вида $+a(E)$ и, если такой факт есть, то к базе фактов для C присоединяем факт $+a(C)$ и к базе фактов для D присоединяется факт $+a(D)$. Вторая продукция действует так: в базе фактов для E и в базе фактов для C разыскиваются соответственно факты вида

$-a(E)$ и $+a(C)$; если такие два факта найдены, то к базе фактов для D присоединяется факт $-a(D)$.

5. Паттерн $C = D$ THAT P .

Продукции:

$+X(C) \Rightarrow +X(D) ; +X(P)$
 $-X(C) ; +X(D) \Rightarrow -X(P)$
 $-X(C) ; +X(P) \Rightarrow -X(D)$
 $-X(D) \Rightarrow -X(C)$
 $-X(P) \Rightarrow -X(C)$

7. Паттерн $C = L$ SOME D .

Продукции:

$+(X, Y)(L) \Rightarrow +X(C) ; +Y(D)$
 $+X(C) ; +(X, Y)(L) \Rightarrow +Y(D)$
 $+X(C) \Rightarrow +(X, cj)(L) ; +cj(D)$
 $-X(C) \Rightarrow -(X, xj)(L)$
 $-X(D) \Rightarrow -(xj, X)(L)$

В указанных продукциях C, D, E, P, L, M являются параметрами со значениями в множестве *Name*. Поэтому это продукции общего вида. Конкретные продукции получают тогда, когда параметры получают значения

В БМЗ интерпретатор продукции реализует тактику параллельного их исполнения «снизу вверх».

Пример 3.1. Возьмем онтологию, состоящую из предложений онтологии $O\text{-St}^*$ и онтологии O_1 , но без предложения EXIST C2 Пусть Prod – множество конкретных продукций, ассоциированных с этой онтологией. Процесс применения Prod начинается с пустой базы фактов $B_0 = \emptyset$. Тогда к B_0 применимы только продукции из Prod с паттерном EXIST C. В результате их применения получается новая база фактов

$B_1 = \{+c_1(\text{Волк}), +c_2(\text{Лиса}), +c_3(\text{Птица}), +c_1(\text{Гусеница}), +c_1(\text{Улитка}), +c_1(\text{Злак})\}$. К базе фактов B_1 применимы продукции из Prod с паттерном $C \text{ ISA } D$.

Продолжая применять продукции из Prod, мы на некотором шаге получим базу фактов, содержащую факт $+c_3(C2)$. Применяв к этому факту продукцию $+X(C2) \Rightarrow \text{EXIST } C2$, получим EXIST C2, т.е. истинно предложение 15.

Заключение

В настоящей статье мы определили некоторый язык ЯЛС-0 для логической спецификации онтологий. Этот язык является расширением базового языка ALC дескриптивной логики [Baader, 2003]. На самом деле C-термы языка ЯЛС-0 (обозначаемые классы объектов) записываются в так называемом манчестерском синтаксисе, который был предложен в [Horridge et al. 2006] для основанного на дескриптивной логике языка OWL. Для логического вывода в онтологиях, записанных в языке ЯЛС-0, мы предложили метод, основанный на применении продукций.

Работа выполнена при финансовой поддержке Российского Фонда Фундаментальных Исследований по проекту №14-07-00387.

Библиографический список

[Baader et al. 2003] Baader F., Calvanese D., McGuinness D., Nardi D., Patel-Schneider P. (eds.) Description Logic Handbook. – Cambridge University Press, 2003.

[Horridge et al. 2006] Horridge M., Drummond D., Goodwin J., Rector A., Stevens R., Wang H. The Manchester OWL Syntax // OWL Experience and Directions Workshop, 2006.

[Walther 2003] Walther C. A mechanical solution of Schubert's Steamroller by many-sorted resolution // Artificial Intelligence, .26 (2), 1985.

[Плесневич 2013] Плесневич Г.С. . Формальные онтологии //20-я Международная научно-техническая конференция "Открытые семантические технологии проектирования интеллектуальных систем" OSTIS-2012 (16 - 18 февраля 2012, Минск), Минск, 2012.

A LANGUAGE FOR SPECIFYING ONTOLOGIES

Plesniewicz G.S.

Moscow Power Engineering University, Russia

salve777@mail.ru

Some language for logical specification of ontologies is presented. This language is an extension of the description logic language ALC.

Introduction

We present some language ЯЛС-0 for specifying ontologies. The language belongs to the system “Binary Model of Knowledge” of concept-based languages. (The system is under development at the Applied Mathematics Department of the Moscow Power Engineering Institute.) The syntax, denotative and operational semantics of the language ЯЛС-0 are defined. As example, we have written in ЯЛС-0 the ontology for the well-known problem “Steamroller”. It is shown how to solve the problem by applying productions associated with the ontology.

Main part

The syntax of the language ЯЛС-0 comprises the rules for defining the so-called C-terms, L-terms and P-terms. C-terms denote classes of individual objects of a given problem domain, L-terms denote binary relations between individual objects, and P-terms denote unary predicates. C-terms have the same semantics as the semantics of concept descriptions in the language ALC. Therefore, ЯЛС-0 is an extension of ALC. Syntax of ЯЛС-0 is closed to the so-called Manchester syntax. Statements of ЯЛС-0 have the form almost exactly corresponded to natural language statements.

With any ontology O it is associated some set Prod(O) of productions. We show how Prod(O) can be applied for obtaining logical consequences from O .

Conclusion

In the paper we present some language ЯЛС-0 for logical specifying ontologies. The language is an extension of the basic description logic language ALC.