Использование системы интерактивного доказательства для отображения онтологий ♣

© Н. А. Скворцов

Институт проблем информатики РАН
nskv@synth.ipi.ac.ru

Аннотация

Сложность онтологических моделей варьируется от простых таксономий до логики первого порядка. Оптимальная сложность выбирается, исходя из необходимой выразительной мощности с одной стороны и возможности решения задач логического вывода с другой. Для проверки правильности отображения двух онтологий необходимо доказательство отношения поглощения между онтологическими понятиями. Эта задача автоматически разрешима в дескриптивных логиках. На одной из таких логик основана онтологическая модель языка OWL DL. Однако с задачей отображения между онтологическими понятиями приходится сталкиваться и для более сложных моделей. В статье показано, как система интерактивного доказательства уточнения спецификаций – B-Toolkit, основанная на логике первого порядка, может использоваться для проверки отображения онтологических понятий.

1 Введение

Для понятийного описания предметных областей и выражения семантики объектов реального мира используются онтологии. Онтологические модели за время исследований в этой области претерпели значительное развитие. Для представления онтологий используются как простые неформальные модели, основанные на лингвистических определениях и связях, так и выразительные логические модели. Оптимальная сложность модели выбирается, исходя из необходимой выразительной мощности с одной стороны и возможности решения задач логического вывода с другой.

Среди первых широко распространённых онтологических моделей известна модель языка Ontolingua [2]. Он создавался как язык для однородного представления онтологических знаний, созданных в различных системах представления знаний, и для создания общих онтологий для групп агентов. Предикативные выражения в Ontolingua представляются на языке KIF [1], который основан на логике первого порядка. В связи с выразительностью языка KIF спецификации на языке Ontolingua в общем случае не поддаются автоматическому логическому выводу для многих задач. В основном Ontolingua используется для обмена неоднородными онтологиями, так как существует множество отображений онтологических языков в язык Ontolingua.

Сегодня наиболее распространённой онтологической моделью, рекомендуемой консорциумом W3C, является язык OWL [7]. В нём определены три диалекта: OWL Lite, OWL DL и OWL Full. Эти стили представления онтологий отличает друг от друга сложность моделей. Тот или иной диалект может использоваться в зависимости от требований к простоте вывода и формальности описаний. Диалект OWL DL составлен в соответствии с дескриптивной логикой [3], основной принцип которой состоит в возможности автоматического доказательства непротиворечивости внутри одного определения или отношения поглощения между разными определениями. Такие возможности формального вывода необходимы во многих задачах: проверка корректности онтологии, обработка запросов в терминах онтологии, отображение и интеграция онтологий. Диалект OWL Full позволяет использовать конструкции языка таким образом, при котором обозначенные задачи вывода становятся неразрешимыми.

Возникают ситуации, когда использование моделей, основанных на простых логиках, подобных OWL DL, оказывается недостаточным. Например, когда их выразительная мощность не позволяет удовлетворительно описать семантику понятий предметной области, либо когда используемая онтология изначально описана в более сложной модели, и приведение её в соответствие модели OWL DL приведёт к недопустимой потере семантики.

В случае, когда автоматический вывод в онтологической модели невозможен ввиду её сложности, доказательство приходится проводить эксперту. Для этого можно привлекать системы интерактивного доказательства теорем, помогающие

Труды 8ой Всероссийской научной конференции "Электронные библиотеки: перспективные методы и технологии, электронные коллекции" - RCDL'2006, Суздаль, Россия, 17-19 октября, 2006

доказывать логические утверждения. И если обработку запросов в терминах онтологии вряд ли можно решать таким способом, то в задачах проверки корректности онтологии, а также отображения и интеграция онтологий такой подход может оказываться вполне оправданным. Одним из средств интерактивного доказательства является система B-Toolkit [6], предназначенная для доказательства уточнения спецификаций программ и основанная на логике первого порядка.

Примечательно, что отношение уточнения между спецификациями, проверяемое в системе B-Toolkit, удачно применимо для формальной проверки отображения между онтологиями, что будет показано далее в разделе 2. Раздел 3 кратко описывает систему интерактивного доказательства B-Toolkit. Для исследовательских целей язык OWL отображается во внутренний язык системы B-Toolkit. В разделе 4 приводится пример отображения конкретной онтологии, проводятся рассуждения о более сложных моделях.

2 Применение уточнения типов для отображения онтологий

Задача отображения онтологий возникает часто, когда необходимо согласовать разные представления о предметной области и семантику объектов, описанных в терминах разных онтологий. При отображении онтологий производится поиск отношений между онтологическими понятиями разных онтологических контекстов, выясняется, какие понятия другой онтологии могут быть эквивалентны, какие являются суперпонятиями или подпонятиями данной онтологии. Так, для проверки правильности отображения понятий двух онтологий, выраженных в формальных моделях, необходимо доказательство восстановленных отношений между понятиями. В моделях, приводимых к дескриптивным логикам, эта задача сводится к автоматическому доказательству отношения поглощения между онтологическими понятиями. Поглощение одного класса другим в дескриптивных логиках означает, что для любых интерпретаций классов элементы поглощаемого класса являются также элементами и поглощающего класса.

Онтологические понятия, выраженные формальными спецификациями, можно трактовать как абстрактные типы данных. И в терминах абстрактных типов данных проверка отображения онтологических понятий заключается в проверке уточнения между спецификациями типов понятий. Данная идея уже высказывалась ранее в [4]. Отношение уточнения абстрактных типов данных отличается от отношения тип/подтип тем, что спецификации типа наследуются подтипом, в то время как при отношении уточнения элементы спецификаций могут переименовываться, иметь отличную структуру и функциональность, но с известными отношениями между элементами двух моделей. В этом отношении в задаче отображения онтологий термин уточнения онтологических понятий предпочтительнее использовать, нежели понятие/подпонятие или поглощение.

3 Принципы работы системы B-Toolkit

Как говорилось выше, система B-Toolkit является системой интерактивного доказательства уточнения спецификаций. Спецификация A уточняет спецификацию B, если A можно использовать вместо B, не замечая факта подмены. В качестве входного языка системы используется нотация абстрактные машины (AMN). В абстрактной машине можно описывать константы, их свойства, переменные, инварианты переменных, инициализацию их значений и операции с пред- и постусловиями. Одна абстрактная машина может декларироваться как уточнение другой. Основой языка является логика предикатов первого порядка. Элементы нотации абстрактных машин и их интерпретацию в логике первого порядка смотрите в документации к системе [6].

Из спецификаций в нотации абстрактных машин система генерирует теоремы и применением определённого набора правил доказывает их. Если для некоторых теорем автоматического доказательства провести не удалось, к процессу доказательства привлекается эксперт, работающий с системой. Принципы интерактивного доказательства уточнения экспертом включают:

- изменение глубины применения правил, после которой система прекращает попытки доказательства;
- изменение последовательности применения правил;
- сокращение перебора указанием частных решений и другое.

Для представления онтологий в виде абстрактных машин необходимо произвести отображение в неё онтологической модели данной онтологии. В частности, было проведено отображение в AMN элементов языка OWL. В таблице приведены элементы языка OWL, соответствующие им конструкции дескриптивной логики SHIQ, семантика элементов языка, выраженная на языке теории множеств и логики предикатов первого порядка, и соответствующие спецификации в нотации абстрактным машин AMN системы B-Toolkit. Ключевые слова VARIABLES, CONSTANTS, INVARIANT и PROPERTIES означают разделы, в которых должны быть описаны следующие за ними утверждения. Это не значит, что для каждого элемента генерируется отдельный раздел. Отображение сделано с целью демонстрации отношения между моделями, и исследования, уточнение какого подмножества языка OWL может доказываться автоматически системой B-Toolkit. В следующем разделе на простом примере показывается отображение онтологии в нотацию абстрактных машин.

Элемент OWL	Конструкция SHIF	Семаника	Представление в AMN
Nothing	$\bot$	$\varnothing$	{}
Thing	$\top$	Δ	SETS Obj
Class	C	$C \subseteq \Delta$	VARIABLES C INVARIANT C: POW(Obj) OPERATION set_C(p)= PRE p:Obj & … THEN C := C ∨ {p} || … END
intersectionOf	$C_3 := C_1 \sqcap C_2$	$C_3 = C_1 \cap C_2$	C3 = C1 ∧ C2
unionOf	$C_3 := C_1 \sqcup C_2$	$C_3 = C_1 \cup C_2$	C3 = C1 ∨ C2
complementOf	$C_2 := \neg C_1$	$C_2 = \Delta \setminus C_1$	C2 = Obj - C1
oneOf	$C := \{I_1\} \sqcup \{I_2\} \sqcup \ldots$	$C = \{I_1, I_2, \ldots\}$	CONSTANTS I1,I2 INVARIANT C={I1,I2,…}
allValuesFrom	$C_2 := \forall R.C_1$	$C_2 = \{p \mid \forall q <p,q> \in R \rightarrow q \in C_1\}$	R: C2 <-> C1
someValuesFrom	$C_2 := \exists R.C_1$	$C_{pch2} = \{p \mid \exists q <p,q> \in R \wedge q \in C_1\}$	C2={p:dom(R) | #q.((p|->q):R & q:C1)}
hasValue	$C := \exists R.\{I\}$	$C = \{p \mid <p,I> \in R\}$	C={p:dom(R) | (p|->I):R}
minCardinality	$C := \geq_n R$	$C = \{p \mid \#\{q \mid <p,q> \in R\} \geq n\}$	C={p:dom(R) | card(R[{p}])>=n}
maxCardinality	$C := \leq_n R$	$C = \{p \mid \#\{q \mid <p,q> \in R\} \leq n\}$	C={p:dom(R) | card(R[{p}])<=n}
cardinality	$C := =_n R$	$C = \{p \mid \#\{q \mid <p,q> \in R\} = n\}$	C={p:dom(R) | card(R[{p}])=n}
subClassOf	$C_1 \sqsubseteq C_2$	$C_1 \subseteq C_2$	C1 <: C2
equivalentClass	$C_1 \equiv C_2 \equiv \ldots$	$C_1 = C_2 = \ldots$	C1 = C2
equivalentProperty	$R_1 \equiv R_2 \equiv \ldots$	$R_1 = R_2 = \ldots$	R1 = R2
disjointWith	$C_1 \sqsubseteq \neg C_2$ $C_1 \sqcap C_2 \equiv \bot$	$C_1 \cap C_2 = \varnothing$ $C_1 \subseteq \Delta \setminus C_2$	C1 ∧ C2 = {} C1 <: Obj-C2
ObjectProperty	R	$R \subseteq \Delta \times \Delta$	VARIABLES R INVARIANT R:C <-> … OPERATION set_R(p1,p2)= PRE p1:C & p2:… & … THEN R := R ∨ {p2 |-> p1} || … END
DatatypeProperty	A	$A \subseteq \Delta \times D$	VARIABLES A INVARIANT A:C --> D OPERATION set_C_A(p) = PRE p:D THEN A := p END
subPropertyOf	$R_1 \sqsubseteq R_2$	$R_1 \subseteq R_2$	R1 <: R2
domain	$\leq_1 R \sqsubseteq C$	$R \subseteq C \times \Delta$	R: C <-> … C <| R
range	$\top \sqsubseteq \forall R.C$	$R \subseteq \Delta \times C$	R: … <-> C R |> C
inverseOf	$R_1 \equiv R_2^-$	$\Delta \times \Delta$	R1 = R2~
TransitiveProperty	$R^+ \sqsubseteq R$	$<R>^+$	!p,q,t.((p|->q):R & (q|->t):R -> (p|->t):R)
FunctionalProperty	$\top \sqsubseteq \leq_1 R$	$\{<p,q> \mid \#\{q \mid <p,q> \in R\} \leq 1\}$	!p.(p:dom(R) -> card(R[{p}])<=1)
InverseFunctional Property	$\top \sqsubseteq \leq_1 R^-$	$\{<p,q> \mid \#\{q \mid <p,q> \in R^-\} \leq 1\}$	!p.(p:dom(R~) -> card(R~[{p}])<=1)
SymmetricProperty	$R \equiv R^-$	$R = R^-$	R = R~
individual	I	$I \in \Delta$	CONSTANTS I PROPERTIES I:Obj
type	I:C	$I \in C$	CONSTANTS I INVARIANT I:C
value	$<I_1,I_2>:R$	$<I_1,I_2> \in R$	(I1|->I2):R
sameAs	$I_1 = I_2 = \ldots$	$I_1 = I_2 = \ldots$	PROPERTIES I1 = I2 = …
differentFrom, AllDifferent	$I_1 \neq I_2 \neq \ldots$	$I_1 \neq I_2 \neq \ldots$	PROPERTIES I1 /= I2 & …

Таблица: Отображение элементов языка OWL в нотацию абстрактных машин системы B-ToolKit

4 Пример отображения онтологии в систему B-Toolkit

Представим простую онтологию, описывающую понятие Wine (сорт вина), которое является подпонятием понятия Drink (напиток). Понятие Wine имеет отношение madeOfGrape (сделано из винограда) с понятием Grape (сорт винограда). Вводится ограничение множественности отношения в понятии Wine не меньше 1. Данная онтология является небольшой частью онтологии винного ресторана, приводимой в качестве примера спецификации на языке OWL в документах рекомендации W3C [7].

Данная онтология на языке OWL выглядит так:

```
<owl:Class rdf:ID="Grape"/>

<owl:Class rdf:ID="Wine">
<rdfs:subClassOf rdf:resource="&food;Drink"/>
<owl:Restriction>
```

```
  <owl:onProperty rdf:resource="#madeOfGrape"/>
  <owl:minCardinality>1</owl:minCardinality>
</owl:Restriction>
</owl:Class>

<owl:ObjectProperty rdf:ID="madeOfGrape">
  <rdfs:domain rdf:resource="#Wine"/>
  <rdfs:range rdf:resource="#Grape"/>
</owl:ObjectProperty>
```

Соответствующее определение понятия Wine в дескриптивной логике будет выглядеть следующим образом:

$$\begin{aligned} \text{Wine} := \quad & \text{Drink} \sqcap \\ & \forall \text{madeOfGrape.Grape} \sqcap \\ & \geq_1 \text{madeOfGrape} \end{aligned}$$

В нотации абстрактных машин системы B-Toolkit определяется множество `Obj` всех объектов. Если бы в онтологии упоминались конкретные объекты, они декларировались бы в машине как константы. Классы понятий `Drink`, `Wine`, `Grape` и отношение `madeOfGrape` между классами становятся переменными. В инварианте абстрактной машины декларируется, что классы понятий могут быть любым подмножеством множества `Obj`, устанавливается отношение подмножества между `Drink` и `Wine`, определяется область определения и значений отношения `madeOfGrape`, и накладывается ограничение на его множественность. Для переменных должны быть определены функции `get` и `set`, однако функции get опускаются как не влияющие на состояние абстрактной машины.

```
MACHINE WineOntology
SETS Obj
VARIABLES Drink, Wine, Grape, madeOfGrape
INVARIANT
  Drink: POW(Obj) &
  Wine: POW(Obj) &
  Grape: POW(Obj) &
  Drink <: Wine &
  madeOfGrape: Wine <-> Grape &
  Wine = { p: dom(madeOfGrape) |
    card(madeOfGrape[{p}]) >= 1}
OPERATIONS
  set_Wine(p1) =
  PRE p1: Wine
  THEN
    Wine := Wine \/ {p1} || Drink := Drink \/ {p1}
  END;
  set_madeOfGrape(p1, p2) =
  PRE p1: Wine & p2: Grape &
    Wine = { p: dom(madeOfGrape) |
      card((madeOfGrape \/ {p1 |-> p2})[{p}])>=1}
  THEN
    madeOfGrape := madeOfGrape \/ {p1 |-> p2}
  END
END
```

Предположим, что описанная онтология уточняет другую онтологию `RefinedWineOntology` с более широким определением понятия вино, RefinedWine, в котором может встретиться сорт вина и не из винограда. С понятием RefinedWine связано отношение ingredients (состав), экземпляры которого могут быть любыми фруктами.

В этом случае для проверки корректности уточнения понятий строится по приведённым выше принципам спецификация абстрактной машины, а в уточняющей онтологии в нотации абстрактных машин дополнительно необходимо указать имя уточняемой спецификации машины.

```
REFINING RefinedWineOntology
```

В инварианте уточняющей абстрактной машины необходимо доопределить, как именно уточняются константы и переменные уточняемой спецификации.

```
RefinedWine -> Wine &
ingredients -> madeOfGrape & …
```

Описанный подход проверки отображения онтологий используется при решении задач над множественными неоднородными источниками информации. Перед любыми манипуляциями онтологическими спецификациями необходимо отобразить вовлекаемые в решение задачи онтологии в выбранную каноническую модель онтологий, являющуюся подмножеством модели языка СИНТЕЗ. Это может быть достаточно сложная модель. Для использования интерактивной системы доказательства спецификации в канонической модели отображаются в нотацию абстрактных машин. Такое отображение для языка СИНТЕЗ описано в [5].

5 Заключение

Рассмотрено применение системы интерактивного доказательства уточнения спецификаций B-Toolkit для формальной проверки корректности отображения онтологий в сложных онтологических моделях. Приведено отображение языка OWL в нотацию абстрактных машин, являющуюся входным языком спецификаций системы B-Toolkit.

Литература

[1] M. Genesereth, R. Fikes. Knowledge interchange format, version 3.0, reference manual. Technical report, Logic-92-1, Computer Science Dept., Stanford University, 1992

[2] T. Gruber. Ontolingua: a Mechanism to Support Portable Ontologies. Knowledge Systems Laboratory of Stanford University, Jun 1992

[3] I. Horrocks, U. Sattler, S. Tobies. Practical reasoning for very expressive description logics. Logic Journal of IGPL, 8(3), 2000

[4] L. Kalinichenko, N. Skvortsov. Ontology reconciliation in terms of type refinement In Proceedings of the 6th Russian Conference on Digital Libraries RCDL2004, Pushchino, Russia, Sep 2004
[5] L. Kalinichenko, S. Stupnikov, D. Briukhov, N. Zemtsov, D. Martynov, N. Skvortsov, M. Bernadskiy, F. Fomenko, V. Zadorozhny et al. The Systems and Means of Informatics: Special Issue. Formal Methods and Models for Compositional Infrastuctures of Distributed Information Systems / Ed. by I.A. Sokolov. M: IPI RAN, 2005. - 304 p. (In Russian)
[6] B-Toolkit User's Manual - Release 3.4. Copyright B-Core (UK) Ltd., 1997
[7] OWL Web Ontology Language Guide. W3C. http://www.w3.org/TR/owl-guide/

Using of an interactive proving system for ontology mapping

N. A. Skvortsov

Complexity of ontological models varies from simple taxonomies to first order logic. Optimal complexity is chosen with respect to necessary expressive power on one hand and ability to solve tasks of logical inference on the other hand. To verify mapping of two ontologies, proof of subsumption relation between ontological concepts required. This task is solvable automatically in description logics. OWL DL language is based on one of such logics. However this task emerges in more complex models too. The paper shows how the system of interactive proving of specification refinement B-Toolkit based on the first order logic may be used for verification of ontological concept mapping.

♣ Исследование поддержано грантами РФФИ 06-07-89188-а, 05-07-90413-в, 06-07-08072-офи.