

ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ ДЕСКРИПТОРНЫХ ЛОГИК В СИСТЕМАХ УПРАВЛЕНИЯ ЗНАНИЯМИ

Рассматривается популярный в последнее время подход к манипулированию данными учетом семантической составляющей. Обсуждаются различные варианты применения формализма в сравнении с традиционными. Предлагаются возможные пути развития рассматриваемого подхода.

В последние годы большой интерес вызывает возможность использования программно-аппаратных комплексов при решении проблем автоматизации исследований для предметных областей (ПО), которые принято называть «слабо-формализованными» – последнее, на содержательном уровне, означает отсутствие строгой теории, обеспечивающей алгоритмическое описание поведения исследуемых объектов. Как правило, слабо формализованными являются модели, соответствующие сверхсложным многофакторным и многосвязным проблемам. К последним можно смело отнести большинство задач, относящихся к физике атмосферы.

Целью данной статьи является изложение базовых принципов весьма перспективного современного формализма, нашедшего применение в системах управления базами знаний (СУБЗ) – формализма дескрипторных (терминологических) логик [1]. Обсуждаемый формализм базируется на идеях, сформулированных Р. Брачманом в [2] и воплотившихся в 1978 г. в системе KLONE. Так же как семантические сети и объектно-ориентированные СУБД, дескрипторные логики (ДЛ) основываются на распространенном интуитивно понятном представлении об окружающем мире, как о состоящем из множества объектов, связанных между собой отношениями и объединенных в классы. Дескрипторные языки (ДЯ) представляют собой удобный инструмент для описания и управления такими классами.

1. ДЕСКРИПТОРНЫЕ ЯЗЫКИ

Дескрипторные логики являются подмножеством модальных логик первого порядка, при этом они дают исследователю более мощные и удобные средства для представления знаний об анализируемой предметной области. Речь идет о «классических» дескрипторных логиках. Позднее появились различные их модификации (многозначные, вероятностные, временные и т.д.).

Введем несколько понятий, которые понадобятся нам в дальнейшем. Как правило, дескрипторные логики рассматриваются как специализированные языки, поддерживающие композицию термов [3]. Все ДЛ обладают, как минимум, двумя видами термов: концептами (классами – элементами домена) и ролями (бинарными отношениями на домене). Концепт может также быть комбинацией ранее определенных концептов и ролей, которая строится с использованием одного из конструкторов дескрипторного языка, что позволяет моделировать более сложные объекты анализируемой ПО. На рис. 1 приведен пример типичного композиционного описания в синтаксисе логики CLASSIC (CLASSification of Individuals and Concepts) [4].

(and
COURSE,
(at-most 6 takers),
(all takers GRADS))

Рис. 1. Пример композиционного описания

Смысл его можно сформулировать так: «Курсы с не более чем 6-ю слушателями, все из которых принадлежат классу GRADS». Дескрипторные логики позволя-

ют расширять набор конструкторов сверх стандартных (конъюнкции, отрицания, квантора всеобщности и т.д.), обеспечивая таким образом лаконичное описание предметной области. На рис. 2 представлен практически исчерпывающий набор не зависящих от конкретного приложения конструкторов, предложенный в обзоре [5]. Здесь C и D обозначают концепты, p и q – роли, b_i – объекты. Семантика конструкторов определяется через понятие интерпретации. Интерпретация I состоит из домена Δ_I и интерпретирующей функции (отображения) V_I . Отображение V сопоставляет каждому концепту подмножество Δ_I и каждой роли – бинарное отношение на Δ_I . Объекты отображаются в единственный элемент Δ_I .

TOP-CONCEPT	Δ_I
NOTHING	$\emptyset$
TOP-ROLE	$\Delta_I \times \Delta_I$
IDENTITY	$\{(d,d) \mid d \in \Delta_I\}$
and $[C,D]$	$C_I \cap D_I$
or $[C,D]$	$C_I \cup D_I$
not $[C]$	$\Delta_I \setminus C_I$
all $[p,C]$	$\{d \in \Delta_I \mid p_I(d) \subseteq C_I\}$
some $[p,C]$	$\{d \in \Delta_I \mid p_I(d) \cap C_I \neq \emptyset\}$
at-least $[n,p]$	$\{d \in \Delta_I \mid p_I(d) \geq n\}$
at-most $[n,p]$	$\{d \in \Delta_I \mid p_I(d) \leq n\}$
at-least-c $[n,p,C]$	$\{d \in \Delta_I \mid p_I(d) \cap C_I \geq n\}$
at-most-c $[n,p,C]$	$\{d \in \Delta_I \mid p_I(d) \cap C_I \leq n\}$
same-as $[p,q]$	$\{d \in \Delta_I \mid p_I(d) = q_I(d)\}$
subset $[p,q]$	$\{d \in \Delta_I \mid p_I(d) \subseteq q_I(d)\}$
not-same-as $[p,q]$	$\{d \in \Delta_I \mid p_I(d) \neq q_I(d)\}$
fills $[p,b]$	$\{d \in \text{dom}_I \mid b_I \in p_I(d)\}$
not-fills $[p,b]$	$\{d \in \text{dom}_I \mid b_I \notin p_I(d)\}$
one-of $[b_1, \dots, b_m]$	$\{b_{i_1}, \dots, b_{i_m}\}$
role-and $[p,q]$	$p_I \cap q_I$
role-or $[p,q]$	$p_I \cup q_I$
role-not $[p]$	$\Delta_I \times \Delta_I \setminus R_I$
inverse $[p]$	$\{(d,d') \mid (d',d) \in R_I\}$
restrict $[p,C]$	$\{(d,d') \in p_I \mid d' \in C_I\}$
compose $[p,q]$	$p_I \circ q_I$
product $[C,D]$	$C_I \times D_I$
trans $[p]$	$\bigcup_{n \geq 1} (p_I)^n$

Рис. 2. Набор конструкторов описаний

Важнейшей особенностью дескрипторных логик, их визитной карточкой, является способность строить иерархии на концептах. Говорят, что концепт C является подклассом концепта D ($C \Rightarrow D$), если при любой интерпретации I , $C_I \subseteq D_I$ (например, концепт с рис. 1 является подклассом концепта: **and** COURSE, (**at-most** 10 takers)). Аналогично, концепты C и D несвязны, если при любой интерпретации I , $C_I \cap D_I = \emptyset$ (например, концепт с рис. 1 и (**and** COURSE, (**at-least** 7 takers))). И, наконец, C согласован, если существует хотя бы одна интерпретация I , такая, что $C_I \neq \emptyset$.

3. ПРИМЕНЕНИЕ ДЕСКРИПТОРНЫХ ЛОГИК В СУБЗ

Проиллюстрируем использование ДЛ в описании данных и запросов на примере, приведенном в [4]. Предположим, что мы определили схему небольшой базы знаний, включая некоторые примитивные классы PERSONS, STUDENTS, SUBJECTS, SCIENCES и роли has-subject, teaches, taughtBy, age и takers. Опишем некоторое состояние системы. Для начала необходимо ввести новый объект, например Crs431:

```
Crs431:=CREATE-IND().
```

Информация о подобных объектах хранится в базе знаний двумя способами: указанием класса, к которому они принадлежат («Crs431 есть экземпляр COURSE») и указанием их связей через роли («Объекту Crs431 преподает Смит и Анна посещает объект Crs431»). Для этих целей используются операторы INSERT-IN и FILL-WITH, применяемые следующим образом:

```
INSERT-IN(Crs431,COURSE)
FILL-WITH(Crs431,taughtBy,Смит)
FILL-WITH(Crs431,takers,Анна)
```

3.1. Использование ДЛ в запросах

Очевидно, что после некоторого заполнения БД информацию потребуется извлечь. Запрос к СУБЗ, основанный на ДЛ, представляет собой описание, характеризующее искомое множество объектов. Например, если мы захотим получить данные о «всех курсах с не менее чем 10 студентами, на которых преподают преподаватели с факультета информатики», запрос должен выглядеть следующим образом:

```
and (COURSE, at-least (10,takers),
all (taughtBy, all (in-dept,Информатика)))
```

Дескрипторные логики хорошо зарекомендовали себя в ситуациях, когда пользователи слабо знакомы с содержимым или структурой данных или когда они не совсем уверены в том, какой запрос сформировать. Помимо этого они очень хорошо подходят к применению в «data-mining» – поиске интересных последовательностей и зависимостей в больших объемах данных, накопленных в других целях [6]. В последнем случае используется возможность построения иерархии запросов, с помощью которой можно:

1. Определять недопустимые запросы, для которых из схемы базы знаний вытекает невозможность вернуть хотя бы одно значение и предупреждать о неправильной формулировке запроса [7].

2. Плавню обобщать запрос, поскольку довольно часто пустой ответ неудовлетворителен даже для правильно сформулированного запроса. В таких ситуациях запрос обобщается до тех пор, пока в ответе не будет получено непустое значение.

3. Оптимизировать запросы на основе семантики данных [8]. Большинство работ по оптимизации запросов к БД, как правило, концентрируются на оптимизации алгебраического представления запроса и физическом доступе к данным. Дескрипторные логики позволяют строить отношение классификации между запросами и так называемыми проекциями (простыми запросами с хранимым ответом), значительно сокращая область поиска и, следовательно, время исполнения за-

просов. Кроме того, такие области, как исследование данных, предполагают исполнение огромного количества запросов, возможно группой людей, и иногда в течение довольно продолжительного времени. Запросы и ответы на них могут при этом сопровождаться комментариями авторов. Построение иерархии из таких запросов помогает не только сократить общую нагрузку на сервер данных, но и позволяет согласовать работу коллектива из нескольких человек [6]. Для всех современных систем такая операция классификации нового запроса по отношению к ранее исполненным является стандартной.

В нашем языке (рис.2) есть конструктор **fills**, который используется для определений подобно **fills(age,40)**, чтобы описать класс объектов, у которых среди связей с ролью age имеется значение 40. Используя его, можно представить операцию

```
FILL-WITH (Crs431,takers,Иван)
```

как

```
INSERT-IN(Crs431, fills(takers,Иван)).
```

Таким образом, мы переформулируем операторы языка СУБЗ в ограничения в описании объекта:

```
INSERT-IN(New-crs,
and (COURSE,
at-least (25,takers),
fills(subject,"ИИ"),
all (taughtBy, fills (department, Информатика) ) )
```

Рис. 3. Пример определения

Такие конструкции языка, несмотря на свою кажущуюся незначительность, открывают очень широкие возможности: они позволяют СУБЗ обрабатывать неполную информацию об объектах. Например, в приведенном на рис. 3 описании мы не знаем точно человека, который преподает на курсе, но кое-какую информацию все же имеем (в частности, что значение атрибута department у него равно «Информатика»). Эта информация может использоваться при построении ответа на запросы: на запрос типа «Найти все курсы, на которых преподают преподаватели с факультета информатики», может быть возвращено значение New-crs, если описание объекта наследуется от запроса. Стоит обратить внимание на то, что ни одна современная СУБД с такой неполной информацией о New-crs работать не сможет. В противовес этому, очень большую долю в приложениях ДЛ-систем составляет как раз работа с неполной информацией. Современные системы, построенные на основе дескрипторных логик, могут представлять данные с неограниченным количеством «null»-значений (например, «нечто, имеющее не менее 15 связей по роли») и корректно отвечать на запросы, касающиеся их. Помимо этого, большинство систем позволяют давать ответ за полиномиальное время, правда, при некоторых ограничениях на вид запроса.

3.2. Использование ДЛ в ответах

Традиционно, на запрос вида «Кто преподает на New-crs?» или «Сколько лет Ивану?» ответ представляет собой список значений. Используя то, что в ДЛ можно ассоциировать описание с объектом, мы можем давать описательные ответы в терминах, которыми

объект был описан в БД или которые явились результатом дедуктивного вывода. Таким образом, для вышеприведенных вопросов мы можем получить ответ вида **and** (**FACULTY**, **fills** (department, Ин-форма-тика)) или **range** (19, 25). Эта возможность полезна не только, когда информация неполна, но и когда мы не хотим получить ответ в виде списка из-за его большого размера. В [9] было предложено в таких ситуациях давать абстрактные ответы: в случае с ДЛ таким ответом является генерирование наименьшего общего надкласса, описывающего общие свойства найденных объектов. На рис. 4 для двух описаний (*a*) представлены их наибольший общий подкласс (*б*), наименьший общий надкласс (*в*) и связанное дополнение между (*в*) и первым описанием из (*a*). Последнее нуждается в дополнительном комментарии: для двух описаний В и С, таких, что $B \Rightarrow C$, связанным дополнением называется такое максимальное описание D, что **and** (C, D) $\equiv B$.

```

and (COURSE,
      at-most(25,takers),
      all(taughtBy, one-off(Gauss, Euclid)))
and (COURSE, prim(FUNNY-EVENT),
      at-most(20,takers),
      all(taughtBy, one-off(Gauss,Marx)))
      а) Два описания

and (COURSE, FUNNY-EVENT,
      at-most(20,takers),
      all(taughtBy, one-off(Gauss)))
      б) Пересечение двух описаний из а)

and (COURSE,
      at-most(25,takers),
      all(taughtBy, one-off(Gauss,Euclid,Marx)))
      в) Объединение двух описаний из а)

all(taughtBy, one-of(Gauss,Euclid,Marx)))
      г) Связанное дополнение в)
      и первого описания а)??????

```

Рис. 4. Пример дедуктивных операций над двумя описаниями

Наконец, когда схема БД очень велика либо когда пользователь не знаком с семантикой БД, находят свое применение интенциональные ответы на запросы: они показывают, какими свойствами на самом деле должен был бы обладать объект, чтобы удовлетворять запросу. Например, такая возможность бывает полезна при работе группы разработчиков над одним очень большим проектом, содержащим множество ограничений на данные в схеме БД [10]. При добавлении в команду нового разработчика, интенциональные ответы помогают ему быстро разобраться в этой скрытой структуре.

Существуют довольно развитые логики и приложения [6, 7], ориентированные именно на удобство и богатые возможности работы с запросами и полученными ответами.

4. ИСПОЛЬЗОВАНИЕ ДЛ В ОПИСАНИИ И КЛАССИФИКАЦИИ ГРАФИЧЕСКИХ ДАННЫХ

Для многих приложений данные могут представлять- ся не только текстовыми и числовыми выражениями, но

и графическими объектами (такими как видео или фотоснимки). Типичным примером могут являться геоинформационные системы (ГИС), включающие в себя большое количество аэро- и фотоснимков поверхности Земли. Для того, чтобы эффективно использовать графические данные ([13]), необходимо хранить не только изображения сами по себе, но и некоторые сведения о них, которые можно разделить на две группы: регистрационные данные – связанные с форматом изображения (размер в пикселях, способ сжатия, дата создания и т.д.) и метаданные – содержащие семантическую и синтаксическую информацию об изображении. Регистрационные данные состоят обычно из элементарных типов, поддерживаемых классическими СУБД, и не доставляют особых трудностей в своем хранении и обработке. Существенно более проблемными в этом отношении оказываются метаданные, которые используются для представления семантического содержания и структуры (синтаксиса) графических данных.

Синтаксические данные, как правило, могут быть извлечены из изображения автоматически (как, например, спектральные составляющие и пространственные отношения на свойствах изображения, специфичных для носителя и не связанных с реальным миром). Запросы к таким данным основываются на:

- вычислимых характеристиках (например, распределении цвета, текстурных сигнатурах);
- функциях, определенных пользователем (например, определение изменений распределения цвета в видеоряде);
- нетекстовых описаниях (например, цветовых палитрах, поиске по примеру);
- структурных пространственно-временных отношениях (например, «желтый объект в правом верхнем углу снимка»).

Очевидно, что в системах, ориентированных на синтаксические данные, слабо учитывается семантика изображения, его связь с реальным миром.

Напротив, семантические данные отражают смысл изображения. Для этих целей используются аннотации, обычно выраженные в виде ключевых слов, привязанных к графическому объекту, организованных в индексы и, как правило, использующих SQL в качестве языка запросов. Ключевые слова не заменяют собой синтаксических описаний – довольно сложно описать словами текстуру, форму или цвет. Однако использование их наталкивается на естественную преграду: отсутствие стандартного словаря влечет зависимость результата запроса от переименования одних и тех же объектов, а фиксированный словарь делает систему нерасширяемой для использования графических данных при другой интерпретации их содержимого.

Гораздо более эффективным представляется [11, 12] моделирование отношений между сущностями реального мира и графическими данными. Именно для этих целей очень хорошо подходят дескрипторные логики, которые позволяют соединить неинтерпретируемые «голые» графические данные и семантическую модель окружающего мира, автоматически классифицировать описания (и, следовательно, изображения) и использовать понятие «концептуальной схожести для неточных запросов, а так же позволяют рассматривать изображения с различных точек зрения в различных ситуациях.

5. ЗАКЛЮЧЕНИЕ

Дескрипторные языки представляют удобный инструмент для манипулирования данными, нашедший применение практически во всех областях, включая базы данных [8], медицину [11], лингвистику и языки программирования. Область их применения постоянно растет. В настоящее время,

помимо ставших уже «классическими» дескрипторных логик, разработаны расширения для вероятностных ДЛ [13], дескрипторных логик, позволяющие строить n -арные отношения на домене [14], и другие, увеличивающие выразительность языка и стройность создаваемой с их помощью модели предметной области.

ЛИТЕРАТУРА

1. Baader F., Burckert H.-J., Heinsohn J., Hollunder B., Muller J., Nebel B., Nutt W. and Profitlich H. Terminological knowledge representation: a proposal for a terminological logic, DFKI Report. Saarbrücken: DFKI, 1992.
2. Brachman R.J. A Structural paradigm for representing knowledge, PhD Thesis, Harvard University, Division of engineering and applied physics, 1977. Revised as BBN Report № 3605, Bolt Beranek and Newman, Inc., Cambridge, MA, May, 1978.
3. Ait-Kaci H. A lattice theoretic approach to computation based on a calculus of partially ordered type structures., PhD Thesis, University of Pennsylvania, 1984.
4. Borgida Alex. Description logics in data management // IEEE transactions on knowledge and data engineering, 7(5), October, 1995.
5. Borgida A. On the relative expressiveness of description logics and predicate logics // JAIR, 1996. Vol. 82. P. 353–367.
6. Brachman R., Selfridge P., Terveen L., Altman B., Borgida A., Halper F., Kirk T., Lazar A., McGuinness S., Resnick L. Knowledge representation support for data archeology // Int. J. of intelligent and cooperative information systems. 1993, Vol. 2. № 2. P. 159–186.
7. Bechhofer S., Globe C. Using a description logic to drive query interfaces. <http://www.cs.man.ac.uk/mig/people/seanb>
8. Buchheit M., Jeusfeld Manfred A., Staudt M., Nutt W. Subsumption between queries to object-oriented databases., JAIR.
9. Shum C.-D. and Muntz R. Implicit representation of extensional answers // Proc. second int. conf. on expert database systems, in Kerscheberg L., editor Benjamin Cummings, 1989, P. 497–522.
10. Davanbu P., Brachman R., Selfridge P., Ballard B. LaSSIE: A knowledge-based software information system, communications of the ACM, 34(5), May 1991.
11. Rector A.L., Bechhofer S.K., Globe C.A., Horrocks I., Nowlan W.A. and Solomon W.D. The GRAIL concept modelling language for medical terminology // Artificial intelligence in medicine, 1997. № 9. P. 139–171.
12. Smoliar S.W., Zhang H. Content-based video indexing and retrieval. IEEE multimedia. 1994. Vol. 1. № 2. P. 62–72.
13. Koller D., Levy A., Pfeffer A. P-CLASSIC: A tractable probabilistic description logic // Proceed. of the fourteenth national conference on artificial intelligence (AAAI-97). P. 390–397.
14. Luntz C., Sattler U., Tobies S. A suggestion for n -ary description logic.

Статья представлена кафедрой программирования факультета прикладной математики и кибернетики Томского государственного университета, поступила в научную редакцию «Кибернетика» 19 июля 2004 г.