

УДК 004.89

ПРЕИМУЩЕСТВА ДЕСКРИПТИВНОЙ ЛОГИКИ ПРИ ОБРАБОТКЕ ЗНАНИЙ

А.Р. Вахитов, В.Б. Новосельцев

Томский политехнический университет

E-mail: var-sasha@tpu.ru

Обсуждаются области применения дескриптивных логик, базовые термины, а также основные их преимущества по сравнению с классическими исчислениями. Особое внимание уделяется возможности применения дескриптивной логики при обработке знаний, показаны примеры использования дескриптивной логики в этой области. Рассматриваются принципы построения логического вывода для обсуждаемой теории.

Ключевые слова:

Дескриптивная логика, обработка знаний, искусственный интеллект.

В последнее время в области инженерии знаний особой популярностью стал пользоваться формализм представления знаний, называемый дескриптивной логикой (ДЛ). Этот формализм широко применяется при построении интеллектуальных систем, основанных на знаниях, — для Интернет-приложений, в медицине, при решении задач конфигурирования оборудования, для построения электронных библиотек и т. д. Большинство таких систем относятся к классу экспертных, имитируя работу специалиста высокого уровня компетентности.

Как известно, одной из важнейших составляющих экспертных систем является механизм логического вывода, поэтому разработка эффективных алгоритмов и процедур вывода, а также связанных с ними методов представления знаний, привлекает особое внимание исследователей в области когнитивных технологий. Суть состоит в том, что размерности баз знаний практически полезных систем достаточно велики, и, если для вывода на таких объемах информации использовать неэффективные (переборные) алгоритмы, время получения требуемого от системы ответа может оказаться неприемлемо большим.

Исследования в области представления знаний обычно сосредотачиваются на методах обеспечения описания высокого уровня, который может эффективно использоваться для создания интеллектуальных приложений. Под термином «интеллектуальный» подразумевается способность системы найти неявные следствия из явно представленных знаний, имитируя стиль рассуждений человека. Базовые подходы к представлению знаний были разработаны во второй половине прошлого столетия, когда эта область была на пике популярности, и разделились на два направления: логические формализмы и представления, основанные не на логике. Первые позиционировались как универсальный инструмент для различных типов задач. В подобных системах знания представлялись специальными структурами данных (семантические сети, сети фреймов, нейроструктуры и т. д.) и дополнялись специальными, как правило, эвристическими процедурами для работы с ними. В последние годы в этой области стали весьма популярными дескриптивные логики.

Дескриптивные логики позволяют описывать конфигурацию сложных (многокомпонентных) взаимодействующих. В последнее время эта область, вследствие тесной связи с прикладными задачами, становится чрезвычайно популярной в исследованиях по представлению и обработке знаний. Так, для задач робототехники важно уметь по изображению некоторой сцены восстановить ее вербальное (формальное) описание, чтобы использовать его, например, для планирования действий робота. Обсуждаемые формализмы описывают знания о предметной области, предварительно фиксируя некоторые базовые ее понятия (терминологию), а затем используя эти понятия для точного описания свойств и связей соответствующих объектов [1].

Ключевой особенностью ДЛ, отличающей их от предшественников, таких как семантические сети и фреймы, является то, что это — именно логики, т. е. строгие теории с хорошо определенной семантикой. Другой их отличительной особенностью является упор на формальный вывод, как основной способ использования: машина вывода позволяет извлекать имплицитные знания из информации, которая явно содержится в базе системы. ДЛ поддерживают стратегии логического вывода, которые встречаются во многих вариантах интеллектуальных систем обработки информации, и давно используются людьми, для структурирования и понимания мира — классификация понятий и их представителей (экземпляров), типизация, иерархия, а также наследование свойств.

Базовыми терминами ДЛ являются концепты и роли. Концепт — это выражение, соединяющее в себе набор ранее определенных концептов и их свойств (описываемых ролями), которое строится с использованием одного из конструкторов дескриптивного языка, что позволяет моделировать более сложные объекты анализируемой предметной области. ДЛ предоставляют методы работы с такими сложными концептами, технику рассуждений об их свойствах и выводимости на них. ДЛ могут быть довольно успешно использована для построения объяснительной компоненты соответствующей экспертной системы.

Смысл описания можно сформулировать так: «Консультация с не менее, чем семью слушателями, все из которых принадлежат классу *Студент*». ДЛ позволяют расширять набор конструкторов сверх стандартных (конъюнкции, отрицания, квантора всеобщности и т. д.), обеспечивая тем самым лаконичное описание предметной области.

Проиллюстрируем использование ДЛ в описании данных и запросов к ним на примере. Для этого опишем некоторое состояние системы в определенной ими базе знаний, связанной с НИРС. Необходимо ввести новый объект, например, *НИРС1*:

НИРС1:=create-ind()

Информация о подобных объектах хранится в базе знаний двумя способами: указанием класса, к которому они принадлежат («НИРС1 есть экземпляр класса *НИРС*») и указанием их связей через роли («Объектом *НИРС1* руководит *Петров*, и *Иванов* занимается объектом *НИРС1*»). Для этих целей используются операторы **insert-in** и **fill-with**, применяемые следующим образом

insert-in (*НИРС1*, *НИРС*)

fill-with (*НИРС1*, *руководит*, *Петров*)

fill-with (*НИРС1*, *занимается*, *Иванов*)

Очевидно, что после некоторого заполнения базы данных информацию потребуется извлечь. Запрос к системе управления базой знаний, основанной на ДЛ, представляет собой описание, характеризующее искомое множество объектов. Например, если мы захотим получить данные обо всех НИРС, которыми руководит *Петров*, запрос должен выглядеть следующим образом:

and (*НИРС*,

all (*руководит*, *Петров*))

ДЛ 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 при отслеживании последовательностей и зависимостей в больших объемах данных [5]. В последнем случае используется возможность построения иерархии запросов, с помощью которой можно:

- Определять недопустимые запросы, для которых из схемы базы знаний вытекает невозможность вернуть хотя бы одно значение, и предупреждать о неправильной формулировке запроса.
- Плавно 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.
- Оптимизировать запросы на основе семантики данных.

Большинство работ по оптимизации запросов к базе данных (БД), как правило, концентрируются на оптимизации алгебраического представления запроса и на физическом доступе к данным. В свою очередь ДЛ позволяют строить отношение классификации между запросами и так называемыми проекциями (простыми запросами с хранимым ответом), значительно сокращая область поиска и, следовательно, время исполнения запросов. Кроме того, такие области как извлечение информации из данных (*data mining*) 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. Для всех современных систем такая операция классификации нового запроса по отношению к ранее исполненным является стандартной.

В синтаксисе логики CLASSIC 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 системы управления базы знаний (СУБЗ) в ограничения в описании объекта:

insert-in (*New_НИРС*,

and (*НИРС*,

at-least (7, *takers*),

fills (*тема, Нейронные сети*),

all (*руководит, fills (ученая степень, д.т.н.)*))

Такие конструкции языка, несмотря на свою кажущуюся незначительность, открывают очень широкие возможности: они позволяют СУБЗ обрабатывать неполную информацию об объектах. Например, в приведенном выше описании мы не знаем точно человека, который руководит данной темой НИРС, но кое-какую информацию все же имеем (в частности, что его *ученая степень* – д.т.н.). Эта информация может использоваться при построении ответа на запросы: на запрос типа «Найти все темы НИРС, которыми руководят преподаватели, являющиеся докторами технических наук», может быть возвращено значение *New_НИРС*, если описание объекта наследуется от запроса. Стоит обратить внимание на то, что ни одна современная система управления базами данных с такой неполной информацией о *New_НИРС* работать не сможет. В противовес этому, очень большую долю в приложениях систем ДЛ является как раз работа с неполной информацией. Современные системы, построенные на основе ДЛ, могут представлять данные с неограниченным количеством *null*-значений (например, «нечто, имеющее не менее 15 связей по ро-

ли») и корректно отвечать на запросы, касающиеся выражений с подобными недоопределенностями.

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 полезной при работе группы специалистов над одним очень большим проектом, содержащим множество ограничений на данные в схеме БД. При добавлении в команду нового разработчика интенциональные ответы помогают ему быстро разобраться в этой скрытой структуре.

Дескриптивные языки представляют удобный инструмент для манипулирования данными, нашедший применение практически во всех областях, включая базы данных, медицину, лингвистику и языки программирования. Область их применения постоянно растет. В настоящее время, помимо ставших уже классическими ДЛ, разработаны расширения для вероятностных ДЛ, позволяющие строить n -отношения на домене, и другие, увеличивающие выразительность языка и стройность создаваемой с их помощью модели предметной области.

ДЛ предоставляет широкие возможности построения вывода. Для этого сначала необходимо строго сформулировать исчисление (построить аксиоматическую теорию).

Исчисление высказываний L – наиболее простой пример формальной аксиоматической теории в логике, использующий функционально полную систему связей, состоящую из отрицания « \neg » и импликации « \rightarrow ». Исчисление L описывает множество правильно построенных формул, среди которых выделяются тождественно истинные формулы (тавтологии). Это соответствует положению дел в математике, где теоремы, как доказуемые на базе аксиом высказывания, являются тождественно истинными утверждениями.

Язык исчисления L включает набор пропозициональных и специальных символов, а также строгие правила построения допустимых выражений (формул). Пропозициональные символы выбираются из не более чем счетного множества (алфавита), возможно, с индексами: $\{x_i, y_j, \dots\}$; $i, j, \dots = 1, 2, \dots$

К специальным относятся символы логических операций отрицания « \neg » и импликации « \rightarrow », а также вспомогательные символы скобок «(,)».

Понятие формулы исчисления L определяется индуктивно: формулами полагаются все индивидуальные переменные x_i, y_j, \dots , а затем и любые выражения вида $(\neg A)$ и $(A \rightarrow B)$, где в качестве A и B могут выступать произвольные правильно построенные формулы.

Систему (предметных) аксиом исчисления L составляет бесконечное (счетное) множество формул исчисления, построенных с использованием следующих трех схем аксиом (конструкторов):

$$(A_1) \quad (Z_1 \rightarrow (Z_2 \rightarrow Z_1));$$

$$(A_2) \quad ((Z_1 \rightarrow (Z_2 \rightarrow Z_3)) \rightarrow ((Z_1 \rightarrow Z_2) \rightarrow (Z_1 \rightarrow Z_3)));$$

$$(A_3) \quad ((\neg Z_1 \rightarrow \neg Z_2) \rightarrow ((\neg Z_1 \rightarrow Z_2) \rightarrow Z_1)).$$

Аксиомой считается любая формула, полученная согласованной подстановкой в схему аксиом допустимых формул исчисления L вместо переменных Z_1, Z_2, Z_3 .

Единственным правилом вывода в исчислении L является правило *modus ponens*, согласно которому из заданных формул A и $(A \rightarrow B)$ с необходимостью выводится B . Для исчислений описанного класса могут быть построены эффективные и полные стратегии автоматического вывода (см., например, [6]).

В заключение отметим что, дескриптивные языки представляют удобный инструмент для манипулирования данными, нашедший применение практически во всех областях, включая базы данных, медицину, лингвистику и языки программирования. Область их применения постоянно растет. ДЛ 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 при отслеживании последовательностей и зависимостей в больших объемах данных. Таким образом, с момента своего появления в конце XX в. как средство решения семантических и логических задач в семантических сетях ДЛ стала уникальным и важным механизмом в важном современном разделе информационных технологий – представлении знаний.

СПИСОК ЛИТЕРАТУРЫ

1. Vaader F., Nutt. W. Basic description logics [Электронный ресурс] = Основы дескриптивной логики – Электронный документ – Режим доступа: <http://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-02.pdf>. – Загл. с экрана.
2. Nardi D., Brachman J. An Introduction to Description Logics [Электронный ресурс] = Введение в дескриптивную логику. – Электронный документ – Режим доступа: <http://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-01.pdf>. – Загл. с экрана.
3. Тейз А., Грибомон П., Луи Ж. и др. Логический подход к искусственному интеллекту. – М.: Мир, 1990. – Т. 1. От классической логики к логическому программированию. – С. 125–134.
4. Voronkov A. Theorem proving in non-standard logics based on the inverse method // 11th Intern. Conf. on Automated Deduction, D. Kapur, Ed. Lecture Notes in Artificial Intelligence. – 1992. – P. 648–662.
5. Brachman R., Sefridge P. Knowledge representation support for data archeology // Intelligent and Cooperative Information Systems. – 1993. – № 2. – P. 159–186.
6. Новосельцев В.Б. Теория функциональных моделей // Сибирский математический журнал. – 2006. – Т. 47. – № 6. – С. 1242–1354.

Поступила 12.09.2008 г.