

В настоящее время разрабатывается коммерческая версия Флэнг -системы.

Литература

1. HENTENRYCK P., van. Constraint satisfaction in logic Programming. - Cambridge: The MIT Press, 1989.

ПРИМЕНЕНИЕ ЛОГИКИ ДЛЯ РЕШЕНИЯ ЗАДАЧ РЕАЛЬНОГО ВРЕМЕНИ

Мартьянов В.И., Иркутск

Программные комплексы решения задач:

- распознавания ситуаций в условиях неполной и нечеткой информации,

- организации взаимодействия менеджеров;

- поддержки принятия решений

предъявляют высокие требования к быстрдействию систем обработки знаний. Современные реализации "Пролога" в значительной мере удовлетворяют этим требованиям, но только на небольших базах знаний (это верно по крайней мере для реализации "Пролога" на ПЭВМ). Решение задач диагностики (распознавания ситуаций) на "Прологе" может производиться только последовательным применением правил (моделирование прямого вывода [1]). Для некоторых предметных областей, где количество правил исчисляется тысячами и десятками тысяч, подобный подход неприемлем в принципе.

Наш опыт решения задач реального времени показывает неплохие перспективы расширений "Пролога" следующими средствами:

- ограниченным прямым выводом;

- правилом разбора случаев.

Не вдаваясь в аспекты создания единой методологической основы такого расширения, рассмотрим задачи, где данные дополнительные средства необходимы.

Прямой вывод необходим в задаче распознавания ситуаций, которую приведем в следующей облегченной от деталей постановке.

Ситуации задаются n -кой параметров x_1, x_2, \dots, x_n . Каждый параметр x_i имеет фиксированное конечное множество значений A_i . Пространство ситуаций $A_1 \times A_2 \times \dots \times A_n$ содержит некоторое множество классов ситуаций K_1, K_2, \dots, K_s . Задача распознавания ситуаций состоит в определении, какому классу принадлежит некоторая конкретная ситуация $\{a_1, a_2, \dots, a_n\}$, где $a_i \in A_i$.

Очевидно, что каждый класс ситуаций может быть задан бескванторной формулой, использующей константы из множеств A_1, \dots, A_n и отношение равенства. В реальных задачах для описания классов ситуаций используется более мощный язык. Если отбросить процедурные средства в этих описаниях, а также моменты, связанные с индуктивным заданием классов (например, на основе обучения по положительным и отрицательным примерам), то в языке описания классов ситуаций используются отношения и операции, определенные на многоосновной модели $\mathcal{M} = \{A_1, A_2, \dots, A_n\}$. В дальнейшем будем считать, что эти операции и отношения заданы своими графиками. Пусть формулы

$$\Phi_1(x_1, x_2, \dots, x_n), \dots, \Phi_s(x_1, x_2, \dots, x_n)$$

задают классы ситуаций K_1, \dots, K_s соответственно. Тогда для распознавания конкретной ситуации $\{a_1, a_2, \dots, a_n\}$ необходимо вычислить истинностное значение формул

$$\Phi_1(a_1, a_2, \dots, a_n), \dots, \Phi_s(a_1, a_2, \dots, a_n).$$

Для ограниченного прямого логического вывода данная задача имеет верхнюю границу сложности $n^3 * \log_2(n)$.

Данный подход позволил производить распознавание ситуаций на ЕС-1046 за 0,1-0,01 с при 54 параметрах и примерно 5000 классах ситуаций.

При логической спецификации некоторых классов задач (планирование в сетях, теория расписаний) формулы вида

$$A \& (B_1 \vee \dots \vee B_n) \& \dots \& (C_1 \vee \dots \vee C_m) \quad (1)$$

задают варианты возможных решений. Все множество вариантов решений - это совокупность конъюнкций

$$A \& B_1 \& \dots \& C_1; \\ \dots \dots \dots \\ A \& B_n \& \dots \& C_m. \quad (2)$$

Переход от формулы (1) к формулам (2) можно считать применением правила разбора случаев. Формулы вида (1) могут получаться при работе систем АДТ [2]. Конечно, последовательный перебор формул из совокупности (2) практически невозможен, но применение методов ветвей и границ при просмотре путей в дереве решений позволяет работать весьма эффективно. В качестве примера приведем работу [2], а также:

- планирование информационных потоков в иерархической системе - поиск относительно оптимального плана за 2-3 с из примерно нескольких триллионов вариантов;

- планирование занятий в вузе: формула (1) имеет более 4000 дизъюнкций, выбор непротиворечивого варианта за 10-15 с.

Отметим, что расширение "Пролога" средствами удовлетворения ограничением (CSP) можно трактовать как расширение "Про-лога" [3] правилом разбора случаев, если выбор значений параметров из фиксированных конечных множеств рассматривать как выбор члена дизъюнкции. В рамках данного подхода [3] получены блестящие приложения:

- составление графика полетов для компании "Люфтганза";
- составление расписания движения электричек в Бельгии.

Литература

1. КОВАЛЬСКИЙ Р. Логическое программирование // Логическое программирование. - М.: Мир, 1988. - С. 134-166.
2. АЛЕКСАНДРОВ С.Г., МАРТЬЯНОВ В.И. Применение системы АДТ для решения задач сетевого планирования // Интеллектуализация программных средств. - Новосибирск: Наука, 1990. - С. 160-168.

3. HENTENRYCK P., van.Constraint satisfaction in logic programming. - Cambridge: The MIT Press, 1989.

СТРУКТУРНЫЙ ПОДХОД К ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНЫХ МОДЕЛЕЙ

Новосельцев В.Б., Габриель В.Д., Томск

Ниже предложены структура данных и дисциплина их использования, предназначенные для организации описаний предметных областей. Подобные описания могут служить формульными спецификациями при автоматическом синтезе программ. В качестве базового используется понятие схемы отношения. Этот объект определяется выражением вида: $T = (A_1, \dots, A_n | Q)$, где A_i - атрибуты, Q - предикат, отражающий функциональные связи между ними. Атрибут A_i может иметь вид: $N_i : D_i$, где N_i - имя атрибута, а D_i - имя домена. В этом случае атрибут называется простым, а совокупность всех таких объектов образует общую часть отношения. Кроме этого, атрибут может быть представлен выражением:

$$\begin{aligned} & \text{if } P_1(x) \text{ then } A_1, \dots, A_{n_1}, \dots \\ & \dots P_2(x) \text{ then } B_1, \dots, B_{n_2} \text{ fi,} \end{aligned}$$

где A_i, B_i - как и прежде, атрибуты, причем по крайней мере одна из ветвей $\text{if} \dots \text{fi}$ не содержит рекурсивно определенных описаний; $P_j(x)$ - имена предикатов, аргументами которых выступают имена атрибутов общей части схемы отношения. Для отличия имен атрибутов на различных шагах рекурсии используется точечная дисциплина именования.

При отсутствии дополнительных условий вполне реальной представляется ситуация, когда наборы атрибутов, поставленные в соответствие одному и тому же набору объектов моделируемого "мира", появляются в разных отношениях (являясь, таким образом, различными в модели) вместе с семантически эквивалентными, но различными в предметной теории наборами связей. Последнее зачастую приводит к потере эффективности синтеза, по меньшей мере, в силу тавтологичности спецификации. Для устранения этого недостатка предложен формализм, в котором набор схем отношений дополнительно структурируется.

Базисным в предлагаемой модели является понятие метаотношения. Метаотношение определяется тройкой: $M\tau = (H | T | R)$, где $M\tau$ - имя метаотношения; H - заголовок (список имен переменных метаотношения); T - список ссылок на другие метаотношения (каждая ссылка имеет вид: $NT = (HT)$, где HT - имя метаотношения, на которое производится ссылка, HT - список имен переменных); R -