

УДК 681.32

Е. П. Бакулин

*(Новосибирск)***МЕТОДЫ ЛОГИЧЕСКОГО ВЫВОДА
В ЭКСПЕРТНЫХ СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ**

Рассматриваются методы логического вывода в экспертных системах реального времени. Описываются подходы к проблеме ускорения логического вывода при различных временных ограничениях, накладываемых на систему. Показана перспективность параллельной реализации логического вывода.

В последнее время значительное внимание уделяется технологиям искусственного интеллекта (ИИ), в особенности практическим приложениям методов ИИ, таким как экспертные системы (ЭС). Предметом теории ЭС служат методы и приемы конструирования человеко-машинных систем, предназначенных для использования в некоторой узкоспециальной области. Известно большое количество примеров построения ЭС в различных сферах деятельности: в медицине, управлении финансовыми ресурсами, на производстве. В то же время применение ЭС для управления промышленными объектами во многих случаях связано с определенными трудностями. Одна из основных проблем — необходимость работы в реальном масштабе времени. Это требование можно определить как наличие строго ограниченного интервала времени, за который система должна выдать реакцию на внешнее воздействие.

Основой любой ЭС является совокупность знаний, структурированная в целях упрощения процесса принятия решений [1—3]. Знания в ЭС организованы таким образом, чтобы отделить знания о предметной области от других типов знаний, таких как общие знания о том, как решать задачи или взаимодействовать с пользователем. Выделенные знания о предметной области называются базой знаний, тогда как общие знания о нахождении решений задач называются механизмом логического вывода. В ЭС чаще всего используются три основных способа представления знаний: правила (продукции), семантические сети и фреймы. При этом время срабатывания всей системы определяется в основном временем осуществления логического вывода.

В данном обзоре рассмотрены методы ускорения логического вывода в экспертных системах реального времени (ЭС РВ). Как отмечено в [1], работы по ЭС РВ находятся на стадии развития, и пока отсутствует единая методология построения таких систем. Вместе с тем количество примеров использования ЭС РВ достаточно велико. В этих условиях особенно актуальна задача обобщения накопленного опыта и определения основных направлений дальнейшего развития, попытке решения которой посвящена данная работа.

Методы ускорения логического вывода можно условно разделить на программные и аппаратные, причем выбор метода определяется допустимым временем ответной реакции для конкретного приложения.

Программные методы. Следует отметить, что существуют приложения, в которых допустимое время необходимой реакции достаточно велико (порядка десятков минут — нескольких часов). При этом не требуется применения методов ускорения логического вывода. Так, в [4] описана советующая ЭС оператора доменной печи, которая является, по существу, ЭС РВ. Однако процессы в доменной печи протекают достаточно медленно, а количество продукционных правил относительно невелико, так что не возникает необходимости в каких-либо специальных методах ускорения логического вывода.

Близкий к приведенному случай рассмотрен в [5], где описывается программирование ЭС РВ на языке LISP. При этом авторы находят время реакции ЭС достаточным для рассматриваемых приложений и акцентируют внимание только на операциях динамического управления памятью.

Для ускорения логического вывода в системах продукций, содержащих большое количество правил, применяется разбиение множества правил на группы и/или уровни. Данный подход используется во многих работах. Так, например, в [6] предложена иерархическая система продукционных правил, позволяющая получать вывод с различной степенью достоверности в зависимости от времени, отводимого на вывод. Для этого база знаний системы разбита на несколько уровней. На более низких уровнях могут быть получены более детальные выводы, требующие, однако, больших затрат времени. Поиск по дереву (детализация ответа) производится до тех пор, пока не исчерпаны доступные ресурсы времени. В [7] описан интеллектуальный регулятор, в котором на нижнем уровне используются традиционные алгоритмы управления, а на верхнем — система продукционных правил.

Для сокращения пространства поиска в процессе логического вывода может использоваться имитационная модель объекта управления. При этом с помощью модели качественно оценивается ситуация на объекте управления, которая впоследствии подвергается более точному исследованию с использованием системы правил [8]. Теоретическое исследование возможности применения моделей в ЭС РВ приведено в [9].

Интересный подход к ускорению логического вывода предложен в [10]. Авторами разработан планировщик управляющих вычислений, функцией которого является анализ ситуации на объекте управления, определение реакции на сложившуюся ситуацию, нахождение алгоритма реализации выбранной реакции, проектирование программного комплекса, реализующего данный алгоритм, выполнение созданного программного комплекса. Таким образом, система осуществляет самопрограммирование в зависимости от ситуации на объекте управления. Авторами утверждается, что данный подход приводит к сокращению времени реакции системы из-за эффекта, аналогичного динамическому распределению памяти.

Аппаратные методы. Аппаратные методы ускоренного логического вывода сводятся к средствам физического ускорения процесса перебора правил с использованием цифровых [11] или аналоговых [12] СБИС, а также оптических процессоров [13] и к средствам параллельного осуществления логического вывода.

Необходимость применения параллельных вычислений для ускорения логического вывода отмечается во многих работах [14, 15]. Причем подчеркивается, что при решении типичных для искусственного интеллекта задач с экспоненциальной сложностью (NP-полных) мультиобработка — единственный путь повышения компьютерной производительности.

В некоторых работах описывается распределенная ЭС, причем сама ЭС не является параллельной (т. е. реализована на одном процессоре), однако параллельно с ядром ЭС работают периферийные процессоры, обеспечивающие решение низкоуровневых задач, коммуникацию с внешней средой и т. д. Так, в [16] описывается двухпроцессорная система, состоящая из LISP — процессора, используемого для ЭС, и процессора 68010 для сбора данных в реальном времени и других низкоуровневых задач.

Рассмотрим проблемы осуществления параллельного логического вывода более подробно. В общем случае продукционная система состоит из трех основных компонент: рабочей памяти (данные), базы правил и механизма вывода.

Рабочая память представляет собой некоторую информационную структуру для хранения данных. Она доступна для всех без исключения правил, причем весь обмен информацией осуществляется только через рабочую память. База правил состоит из совокупности правил — продукций. Каждая продукция имеет условную часть, задающую состояние рабочей памяти, в которой применима данная продукция, и исполнительную часть, содержащую инструкции по изменению состояния данных. Механизм вывода является цик-

лическим процессом, состоящим из трех тактов: сопоставление правил, выбор правила (из множества готовых к выполнению) и выполнение правила.

Для осуществления параллельного логического вывода необходимо параллельное выполнение сопоставления правил и параллельная активация продукций. В простейшем случае параллельного сопоставления продукций для каждого правила может быть использован свой процессор.

Пусть в результате сопоставления получено множество готовых к выполнению продукций $\{P_1, P_2, \dots, P_n\}$. Параллельное выполнение продукций допустимо только при их взаимной независимости. В то же время одновременное выполнение зависимых правил может привести к нежелательному явлению интерференции правил, заключающемуся в получении такого состояния базы данных S (рабочей памяти), которое не может наступить ни при каком последовательном выполнении правил, т. е. $S(P_i \parallel P_j) \not\subseteq S(P_i \circ P_j) \cup S(P_j \circ P_i)$, где $P_i \parallel P_j$ — параллельное выполнение правил P_i и P_j , $P_i \circ P_j$ — последовательное.

Исключить интерференцию правил можно путем предварительного (на стадии компиляции) разбиения всего множества правил на независимые подмножества (что не всегда осуществимо из-за наличия переменных в левых и правых частях продукций) или при контроле непосредственно в продукционном цикле.

В [17] предложено задавать продукционную систему с помощью таблицы решений (матрицы отображения условия — действия). При этом используются только двоичные значения, что упрощает обработку и делает возможным анализ системы на полноту и непротиворечивость, т. е. устраняет возможность интерференции на этапе разработки.

Многопроцессорная архитектура с общей памятью предложена в [18]. В данной системе применяется вывод с использованием «доски объявлений», причем в системе используется несколько досок объявлений, соответствующих различным (независимым) уровням организации правил в базе знаний.

В [19] проведено исследование возможности параллельного выполнения системы продукционных правил с контролем в цикле интерпретатора.

Так, пусть имеется набор продукционных правил $P = \{P_i\}$, $i = \overline{1, n}$, реализующих отображение $S \rightarrow S$. Каждое правило имеет вид $P_i: L_i \rightarrow R_i$, $L_i, R_i \subseteq S$. Правила P_i и P_j независимы (обозначается $P_i \text{ IND } P_j$), если

$$(L_i \cap L_j = \emptyset) \& (R_i \cap R_j = \emptyset) \& (L_i \cap R_j = \emptyset) \& (L_j \cap R_i = \emptyset).$$

Правила P_i и P_j зависимы:

по входу ($P_i \text{ IN } P_j$), если $(L_i \cap L_j \neq \emptyset)$;

по выходу ($P_i \text{ OUT } P_j$), если $(R_i \cap R_j \neq \emptyset)$;

по входу/выходу ($P_i \text{ IO } P_j$), если $(L_i \cap R_j \neq \emptyset) \vee (L_j \cap R_i \neq \emptyset)$.

Правила P_i и P_j допускают параллельное выполнение:

$$P_i \parallel P_j \Leftrightarrow (((P_i \text{ IND } P_j) \vee (P_i \text{ OUT } P_j)) \& \lceil ((P_i \text{ IN } P_j) \vee (P_i \text{ IO } P_j))) \vee (R_i = R_j).$$

Аналогично для n правил можно построить матрицу взаимозависимости $\parallel Q \parallel$:

$$Q_{ij} = \begin{cases} 1, & \text{если } ((P_i \text{ IN } P_j) \vee (P_i \text{ IO } P_j)) \& (R_i \neq R_j); \\ 0 & \text{в противном случае.} \end{cases}$$

P_i и P_j могут выполняться параллельно, если $Q_{ij} = 0$.

На основе анализа матрицы взаимозависимости правил (выполняемого в цикле интерпретатора) возможно определение множества независимых правил, допускающих параллельное исполнение.

Обсуждение. Упомянутые в обзоре работы в большинстве своем соответствуют развивающемуся направлению так называемых экспертных систем второго поколения, для которых характерны такие особенности, как разбиение базы знаний на знания о проблемной области и управляющие знания, разбиение

ние знаний по уровням абстракции, применение систем продукций совместно с моделями объекта анализа, параллелизм [20].

Следует отметить, что необходимость ускорения логического вывода особенно актуальна при применении экспертных систем, основанных на нечеткой логике, так как в этом случае в процесс вывода обычно вовлекается большее количество правил (в общем случае все правила) и возникает необходимость оперировать массивами данных (нечеткими множествами).

Теория ЭС РВ затрагивает такой фундаментальный вопрос ИИ, как рассуждения в условиях ограниченных ресурсов. В сложной динамической обстановке объем поступающей информации, необходимых выводов и действий обычно превосходит вычислительные ресурсы ЭС, таким образом, все потенциально возможные операции не могут быть проделаны за приемлемое время. Более того, развитие методов аппаратной и программной оптимизации вывода может решить задачу только для конкретного приложения и не решает проблемы осуществления логического вывода в условиях ограниченных ресурсов в общем случае. Для ЭС с любой скоростью вывода может быть найдена задача, для решения которой требуемая вычислительная мощность превосходит ресурсы данной ЭС. Поэтому особого внимания заслуживают параллельные методы осуществления логического вывода как наиболее гибкие, позволяющие построить системы с различным временем ответной реакции.

СПИСОК ЛИТЕРАТУРЫ

1. Чиркова Р. Ю. Методы представления, приобретения и использования знаний в экспертных системах реального времени // Вестник МГУ. Сер. 15. Вычислительная математика и кибернетика. 1994. № 1.
2. Hayes-Roth В. Architectural foundation for real-time performance in intelligent agents // Second Generation Expert Systems. N. Y.: Springer-Verlag, 1993. P. 643.
3. Еремин А. А., Новожилов И. М. Аппаратное, алгоритмическое и программное обеспечение цифровых систем управления с элементами искусственного интеллекта // Изв. Санкт-Петербургского электротехн. ун-та. 1993. № 462.
4. Гринберг С. Я., Яхно Т. М. Решение задач технической диагностики с использованием оболочки ДИ*ГЕН // Изв. АН СССР. Сер. Техн. кибернетика. 1990. № 5.
5. Allard J., Hawkinson L. Real-time programming in common lisp // Commun. ACM. 1991. 34, N 9. P. 103.
6. Bharadwaj K., Jain N. Hierarchical censored production rules (HCPRs) system // Data and Knowledge Eng. 1992. 8, N 1.
7. Wright M. HEXSCON: A hybrid microcomputer-based expert system for real-time control applications // Proc. WESTEX-86: IEEE Western Conf. on Knowledge-Based Engineering and Expert Systems. USA, 1986. P. 49.
8. Редкозубов С. А., Кальбова В. С. Проектирование экспертных систем, работающих в реальном масштабе времени // Механизация и автоматизация производства. 1991. № 5.
9. Suen J. Knowledge architectures for real-time decision support // Second Generation Expert Systems. N. Y.: Springer-Verlag, 1993. P. 699.
10. Косенко А. Б. Технология самопрограммирования экспертных систем реального времени на языке ПРОЛОГ // Сб. тр. Ин-та проблем управления РАН. 1993. № 2.
11. Yamakawa T. High-speed fuzzy controller hardware system: The mega-FIPS machine // Inform. Sci. 1988. 45, N 2. P. 113.
12. Yamakawa T. Intrinsic fuzzy electronic circuit // Fuzzy Computing. North-Holland, 1988. P. 115.
13. Zhang S. Optical fuzzy vector-matrix composition operation using shadow-casting // Opt. Commun. 1992. 94, N 6. P. 497.
14. Переверзев-Орлов В. С. Советчик специалиста. М.: Радио и связь, 1990.
15. Алексеев Г. Г., Петрухина Н. А. К вопросу о выборе архитектуры специализированной мультипроцессорной системы для решения задач искусственного интеллекта // Вопросы радиоэлектроники. Сер. ЭВТ. 1993. № 2.
16. Hawkinson L., Levin M. A paradigm for real-time inference // Proc. of AI and Advanced Computer Technology. 1985. P. 51.
17. Еремеев А. П. Параллельная модель для производственной системы табличного типа // Изв. АН СССР. Сер. Техн. кибернетика. 1990. № 5.
18. Dai H., Anderson T. Issues of real-time expert systems // AI and Cognitive Science'89. N. Y.: Springer-Verlag, 1990. P. 179.
19. Вагин В. Н., Еремеев А. П. Параллелизм в производственных моделях представления знаний // Изв. РАН. Сер. Техн. кибернетика. 1994. № 2.
20. David J., Krivine J. Second generation expert systems: A step forward in knowledge engineering // Second Generation Expert Systems. N. Y.: Springer-Verlag, 1993. P. 3.

Поступила в редакцию 30 января 1996 г.