

## プロダクションシステムを構築するための 推論関係型データベース管理システムについて

高木利久<sup>\*</sup>, 松尾文碩<sup>\*\*</sup>, 二村祥一<sup>\*\*</sup>, 牛島和夫<sup>\*</sup>  
<sup>\*</sup>九州大学工学部情報工学科, <sup>\*\*</sup>九州大学大型計算機センター

筆者らは, Adbisと呼ぶ, 推論関係型データベース管理システムを開発し, 九州大学大型計算機センターにおいて実用に供してきた. AdbisはHorn集合の反証に基づく推論機構を備えている. 本研究は, Adbisにプロダクションシステムの構築を支援する機能を追加するとともに, これを用いて実用規模のエキスパートシステムをAdbisの上に構築することを目指すものである.

本稿では, プロダクションシステム構築支援ツールの仕様と, このツールをHorn集合推論系を用いて実現する方式の概要とについて報告する.

### An Inferential Relational Database System for Constructing Production Systems

Toshihisa Takagi<sup>\*</sup>, Fumihiro Matsuo<sup>\*\*</sup>, Shooichi Futamura<sup>\*\*</sup>, and Kazuo Ushijima<sup>\*</sup>

<sup>\*</sup>Department of Computer Science and Communication Engineering, Kyushu University 36,  
Hakozaki, Higashi-ku, Fukuoka 812, Japan

<sup>\*\*</sup>Computer Center, Kyushu University 91, Hakozaki, Higashi-ku, Fukuoka 812, Japan

We developed an inferential relational database management system called Adbis at the Computer Center, Kyushu University. Adbis has inference functions based on Horn set refutation procedures. We are planning to develop a tool for constructing production systems and build practical expert systems on Adbis using this tool.

In this paper, the specification of the tool and the implementation method of the functions by using the inference system of Horn set are described.

## 1. まえがき

知識ベースシステムやエキスパートシステムの開発、および、これらのシステムの構築を支援するための、いわゆる知識工学用言語やツールの研究開発が近年活発に行われている<sup>1)</sup>。これらのシステムや言語、ツールで使われている知識表現のモデルとしては、述語論理、フレーム、意味ネットワーク、プロダクションルールなどがよく知られている<sup>2)</sup>。

これらのモデルのなかでプロダクションルールに基づくモデルは、

- (1) すべての知識をプロダクションルールと呼ばれる“もし～ならば～せよ”形式の規則で簡単に表現でき、計算機の専門家以外にも理解しやすいこと、
- (2) しかも、それらの規則はコンテキストや作業記憶などと呼ばれるデータ領域を介してしか通信しない、つまり、プロダクションルール同士が直接呼び合うことがないこと、
- (3) そのため、個々の知識のモジュール性が高く、知識の追加、削除、更新などが容易に行えること、
- (4) データ領域、それに作用するプロダクションルール、および、どのプロダクションルールを適用するかを決定する制御機構とがはっきり分離されていること

などの理由から、知識の数が多く、しかも広範囲に渡るような領域において、大規模なエキスパートシステムを改良発展させながら開発するのに適している<sup>3), 4)</sup>。そのため、プロダクションルールに基づく知識表現モデルはOPS5<sup>5)</sup>やEMYCIN<sup>6)</sup>などの代表的な知識工学用ツールで多く採用されている。また、これらのツールを用いて実際にいくつかのエキスパートシステムが開発されている。

しかし、問題の構造に応じた推論の柔軟な制御方式や確度係数(certainty factor)<sup>7)</sup>の処理の問題は未解決である。また、実用化のための技術も十分に確立しているとは言い難い。また、プロダクションシステム以外のモデル、例えば述語論理などの知識表現能力や推論の機構と、プロダクションシステムのそれらとの関係も必ずしも明らかではない。

一方、筆者らは、Adbis<sup>8)</sup>と呼ぶ、推論関係型データベース管理システムを開発し、1982年以来、九州大学大型計算機センターにおいて実用に供してきた。AdbisはHorn集合の反証に基づく推論機構を備えている<sup>9)</sup>。最近、Adbisを使用している機関あるいは使用予定の機関から、エキスパートシステムをAdbisの上に構築したいという要望が出てきた。

本研究は、Adbisにプロダクションシステムの構築を支援する機能を追加するとともに、これを用いた実用規模のエキスパートシステムの構築を目指すものである。また、このシステムの評価および改良を行い、上記の問題点の解決法を探らうとするものである。

現在、上記の計画のうち、プロダクションシステム構築支援ツールの機能の基本設計と、この機能をHorn集合推論系を用いて実現する方式の設計とを終了したので以下に報告する。

2節ではプロダクションシステムを簡単に概説し用語の定義を行う。3節では、今回設計したプロダクションシステム構築支援ツールの構文とその意味について述べる。4節では、Horn集合への翻訳機構について述べる。5節では、Adbisの上にこのツールを実現する方式の概要について述べる。6節では、代表的なプロダクションシステム構築用ツールOPS5との比較を交えながら、提案したツールとその処理方式の特長を明らかにする。

## 2. プロダクションシステム

一般にプロダクションシステムは次の三つの要素から構成される<sup>2), 3), 10), 11)</sup>。

- ・コンテキスト、データベース、作業記憶などと呼ばれるデータの集合。本稿ではコンテキストと呼ぶことにする。
- ・条件部→実行部の形式で表現されるプロダクションルールの集合。以下ではこの集合をルールベースと呼ぶ。
- ・コンテキストによって条件部が満足されるプロダクションルールが複数個ある場合に、どのルールを適用するかを決定するインタプリタと呼ばれる制御機構。

利用者が、ルールベース(プロダクションルールの集合)およびコンテキストの初期状態と目標状態を与えると、プロダクションシステムは以下のように動作する。

- (1) コンテキストを利用者が定義した初期状態に設定する。
- (2) プロダクションルールの条件部とコンテキストを照合する。
- (3) コンテキストが条件部を満足するとそのプロダクションルールを適用可能とする。
- (4) 適用可能なルールが複数個あるとき、制御機構がその中から一つのルールを選択する。
- (5) (4)で選ばれたルールの実行部を適用して、

コンテキストを変更する。

- (6) コンテキストが目標状態に達するとプロダクションシステムの動作を終了する。そうでなければ(2)に戻り、以下の動作を繰り返す。

### 3. プロダクションシステム構築支援ツール

#### 3.1 構文

今回設計したツールの形式的な構文は本稿の付録に載せたので、ここでは構文の概略を説明するにとどめる。

プロダクションルール：個々のプロダクションルールは次の形式で与える。

$$\text{rule名: } A_1, \dots, A_n \rightarrow B_1, \dots, B_m \quad (1 \leq n, 0 \leq m) \quad (1)$$

ここで、 $A_i$ 、 $B_i$ は次のような述語論理における原子論理式(atomic formula)と同じ形式を持つ。

$$P(t_1, \dots, t_k) \quad (2)$$

ここで、 $P$ は述語名、 $t_i$ は変数または定数を表す。以下では、 $A_i$ をプロダクションルールの条件要素、 $B_i$ を実行要素と呼ぶ。

なお、(1)式のrule名は、プロダクションルールの入力や編集時におけるプロダクションルールの管理のための単なる識別子であり、省略可能である。そのためプロダクションシステムの動作上は、まったく意味を持たない。

コンテキストの要素：コンテキストの要素は次の形式とする。

$$Q(x_1, \dots, x_n) \quad (3)$$

これはプロダクションルールの条件要素や実行要素と同じ形式である。

プログラム：2節で述べたように、プロダクションシステムを起動するには、利用者は、ルールベースと、コンテキストの初期状態と目標状態との三つの集合を定義しなければならない。これらの三つの集合の組を本稿ではプログラムと呼ぶことにする。これは次の形式で与える。

$$\begin{aligned} \{ \text{rule1: } & A_{11}, \dots, A_{1n_1} \rightarrow B_{11}, \dots, B_{1m_1}; \\ & \text{rule2: } A_{21}, \dots, A_{2n_2} \rightarrow B_{21}, \dots, B_{2m_2}; \\ & \dots \end{aligned} \quad (4)$$

$$\begin{aligned} \text{rulek: } & A_{k1}, \dots, A_{kn_k} \rightarrow B_{k1}, \dots, B_{km_k} \} \\ \{ & C_1; \dots; C_r \} \\ \{ & D_1; \dots; D_s \} \end{aligned}$$

ここで、rule1からrulekは、(1)の形式のプロダクションルール、 $C_1$ から $C_r$ はコンテキストの初期状態を表す(3)の形式をしたコンテキストの要素である。また、 $D_1$ から $D_s$ も目標状態を表すコンテキストの要素である。

なお、目標状態を書く代わりに、あるプロダクションルールの右辺に、プロダクションシステムの実行を停止するための特殊な組み込み述語を書いてもよい。

#### 3.2 プロダクションルールの適用

プロダクションルールをコンテキストに適用すると、コンテキストがどのように変更されるかということ定義する前に、まずコンテキストがプロダクションルールの条件部を満足するということを定義する必要がある。

いま、コンテキストの現在の状態を、

$$C = \{ \dots, Q_1, \dots, Q_i(x_1, \dots, x_n), \dots, Q_n, \dots \} \quad (5)$$

とし、あるプロダクションルールを

$$A_1, \dots, A_i(t_1, \dots, t_k), \dots, A_n \rightarrow B_1, \dots, B_m \quad (6)$$

とする。 $A_i(t_1, \dots, t_k)$ がコンテキストに含まれているとき、つまり $A_i$ と $Q_i$ 、 $h$ と $k$ 、 $x_i$ と $t_i$ がそれぞれ一致するならば条件要素 $A_i(t_1, \dots, t_k)$ はコンテキストによって満足されたという。このとき、 $t_i$ が変数であれば、適当に定数で置き換えてもよい。

$A_1$ から $A_n$ までのすべての条件要素がコンテキストによって満足されたとき、このプロダクションルールは満足されたといい、このルールを適用可能であると定義する。

プロダクションルールを満足するために必要な変数の置き換えを $\sigma$ とし、条件要素 $A$ 、帰結要素 $B$ に対して置き換え $\sigma$ を行った結果をそれぞれ $A\sigma$ 、 $B\sigma$ とする。

このとき、上の(6)のプロダクションルールをコンテキスト(5)に適用すると、コンテキスト $C$ は次のように変更される。

$$C \leftarrow C - \{A_1\sigma, \dots, A_n\sigma\} + \{B_1\sigma, \dots, B_m\sigma\} \quad (7)$$

つまり、本ツールにおけるプロダクションルールの適用とは、プロダクションルールの条件部の要素をコンテキストから取り除き、実行部の要素を追加することである。

### 3. 3 否定条件の記述

付録で示しているように、以下のようにプログラクシヨナルールの条件要素に否定条件が記述できる。

$$A_1, \dots, \neg A_i, \dots, A_n \rightarrow B_1, \dots, B_m \quad (8)$$

この場合、否定条件要素  $\neg A_i$  がコンテキストによって満足されるか否かは、否定を取り除いた条件要素  $A_i$  により、次のように決定される。

$$\neg A_i \begin{cases} \text{満足される;} & \text{if } A_i \text{ が満足されない} \\ \text{満足されない;} & \text{if } A_i \text{ が満足される} \end{cases}$$

なお、否定の条件要素はプログラクシヨナルールの選択には関係するが適用には関係しない。そのため、否定条件要素自身はコンテキストにはまったく影響を及ぼさない。

### 3. 4 手続き型プログラムの呼び出し

プログラクシヨナルールの条件要素や帰結要素として、Fortran77などの手続き型言語で記述したサブルーチン形式のプログラム名とその引数を書くことにより、それらのプログラムを呼び出し実行することができる。3. 1節で定義したプログラクシヨナルシステムにおけるプログラムと区別するために、手続き型プログラムをプロシジャと呼ぶことにする。

addを3引数のプロシジャとすると、たとえば次のようにして呼び出すことができる。

$$\dots, \text{add}(x, y, z), \dots \rightarrow B_1, \dots, B_m \quad (9)$$

この場合も否定条件の場合と同じくコンテキストにはまったく影響を及ぼさない。

### 3. 5 推論の制御機構

推論の制御は基本的に利用者が与えるものとし、設計の現段階では、次の二つの制御戦略を用意した。

- (1) 適用可能なプログラクシヨナルールが複数あるとき、本ツールでは、定義時に、より上にあるプログラクシヨナルールを選択する。これは、たとえば3. 1節の(4)式のプログラムが与えられた場合、 $i > j$ なら  $\text{rule}_i$  より  $\text{rule}_j$  を選ぶことを意味する。これはPr o

logにおける節選択と同じ戦略である。また、条件部がコンテキストの二箇所以上と一致するときは、より新しいコンテキストの要素を選ぶ。

- (2) プログラムの実行が行き詰まったとき、すなわち、どのプログラクシヨナルールも適用可能でなくなったとき、適用可能なプログラクシヨナルールが複数あった時点または条件部がコンテキストの二箇所以上と一致した時点で自動的にバックトラックし、前に選んだルールあるいはコンテキスト要素とは別のルールあるいはコンテキスト要素を上(1)の基準で選び、実行を続ける。

### 3. 6 記述例

・10の階乗の計算

$$\begin{aligned} &\{\text{factorial}(x, n), \text{add}(x, 1, y), \text{mult}(y, n, m) \\ &\hspace{10em} \rightarrow \text{factorial}(y, m)\} \\ &\{\text{factorial}(0, 1)\} \hspace{10em} (10) \\ &\{\text{factorial}(10, x)\} \end{aligned}$$

ここで、addおよびmultiは足し算およびかけ算を行うプロシジャであるとする。また、 $x, n, y, m$ は変数を表し、それ以外は定数である。 $\{\text{factorial}(0, 1)\}$ は初期状態、 $\{\text{factorial}(10, x)\}$ は終了状態を記述したものである。

### 4. Horn集合推論系への翻訳

#### 4. 1 Horn集合推論系

3節で述べたプログラクシヨナルシステムの動作はHorn集合推論系の機能を拡張することにより容易に実現できる。

Horn節は次の四つの型に分類できる。

$$A_1, \dots, A_n \rightarrow B \quad (n \geq 1) \quad (11)$$

$$\rightarrow B \quad (12)$$

$$A_1, \dots, A_n \rightarrow \quad (n \geq 1) \quad (13)$$

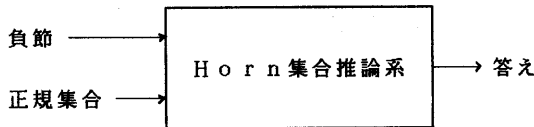
$$\rightarrow \quad (14)$$

(11),(12),(13),(14)をそれぞれ、混合節、正節、負節、空節と呼ぶ。

本稿でいうHorn集合推論系<sup>9)</sup>とは、一つの負節と正規集合(正節と混合節との集合)を入力とし、それらに対して導出原理<sup>12)</sup>に基づく推論を行い、空節を導くことができればyes、そうでなければnoを答えとして返すシステムである。

- (step1) コンテキストの初期状態 $\{C_1; \dots; C_r\}$ を正節の集合 $\{\rightarrow C_1; \dots; \rightarrow C_r\}$ に変換する。  
(step2) プロダクションルール $(A_1, \dots, A_n \rightarrow B_1, \dots, B_m)$ の条件部と実行部を分けて、負節 $(A_1, \dots, A_n \rightarrow)$ と正節の集合 $\{\rightarrow B_1; \dots; \rightarrow B_m\}$ とを作る。これをすべてのプロダクションルールについて行う。  
(step3) コンテキストの終了状態 $\{D_1; \dots; D_s\}$ を負節 $\{D_1, \dots, D_s \rightarrow\}$ に変換する。  
(step4) コンテキストCを正節の集合 $\{\rightarrow C_1; \dots; \rightarrow C_r\}$ で初期化する。  
(step5) Cと負節 $\{D_1, \dots, D_s \rightarrow\}$ とを人力として推論を行う。その答えがyesならば、これはCが終了状態に到達したことを表しているので実行を終了する。そうでなければ、(step6)に行く。  
(step6) iを1に初期化する。  
(step7) iの値がプロダクションルールの個数を越えると(step9)へ行く。  
(step8) プロダクションルールの条件部から作られた負節 $(A_{i1}, \dots, A_{in1} \rightarrow)$ とCとを人力として推論を行う。その答えがyesで、そのときの単一化子が $\sigma$ ならば、Cを $C - \{\rightarrow A_{i1} \sigma, \dots, \rightarrow A_{in1} \sigma\} + \{\rightarrow B_{i1} \sigma, \dots, \rightarrow B_{im1} \sigma\}$ で置き換える。後のバックトラックのためにコンテキストの変化分の情報などを保存して(step5)に戻る。答えがnoならiの値を1増やして(step7)に行く。  
(step9) バックトラックの情報を復元して(step6)へ行く。

図1. 実行の内部処理手順



#### 4. 2 翻訳

コンテキストを正節の集合、プロダクションルールの条件部を負節とみなすことにより、プロダクションルールの条件部とコンテキストの照合は、Horn集合上の推論に置き換えることができる。

推論の結果、答えがyesならば、そのプロダクションルールの条件部はコンテキストによって満足されたことになり、ルールは適用可能になる。もし、答えがnoならば満足されなかったことを意味する。答えがyesの場合、そのときの単一化子(unifier)<sup>12)</sup>を3.2節の置き換え $\sigma$ とみなして(7)式の集合操作を行えばよい。

いま、3.1節(4)のようなプログラムが与えられたとすると図1の手順でプロダクションシステムの動作を遂行することができる。

### 5. 実現

#### 5.1 Adbisの拡張

今回設計したプロダクションシステム構築支援ツールは、先に述べたように推論関係型データベース管理システムAdbis上に実現する。

Adbisは、筆者らが九州大学大型計算機センターのFACOM O SIV/F4上で開発した

関係型データベース管理システムであり、Horn集合の反証に基づく推論を利用したデータベース問い合わせ機構を備えている。プロダクションシステム構築支援ツールを付加したAdbisの構成図を図2に示す。図の破線部が今回追加拡張する箇所を示している。

図2のDMPは関係型データベースの構築・管理・検索などを行うためのモジュールである。

HSIはHorn集合の推論を行うモジュールである。HSIは、4.1節で述べたHorn集合推論系の機能だけでなく、DMPを使って関係型データベースを検索し、その結果をHorn集合に変換する機能やFortran77などの手続き型言語で書かれたプログラムを呼び出す機能を持っている。また、HSIには、失敗による否定(negation as failure)<sup>13)</sup>と呼ばれる推論規則をもっている。さらに、推論の高速化と記憶領域の節約のために、Horn節を内部形式に変換して推論を行う機能をもっている。

Tsuno<sup>14)</sup>, Tsuno-Kakushi<sup>14)</sup>, Waku<sup>15)</sup>は端末利用者のための会話型インタフェースである。

前節で述べたプロダクションシステム構築支援ツールを実現するには、以下の追加および拡張を行えばよい。

会話型インタフェースの追加: プロダクションシステムのプログラムの入力編集を会話型で行うインタフェースを追加する。これは、入力されたプログラムをその管理下のファイルに蓄えると同時に、Horn節へ変換しHSIに送る。この処理は図1の(step1)から(step3)に相当する。なお、この部分の作成には既にあるTsunoのプログラムを再利用することができる。

HSIの拡張: 図1の(step4)以降を行うための

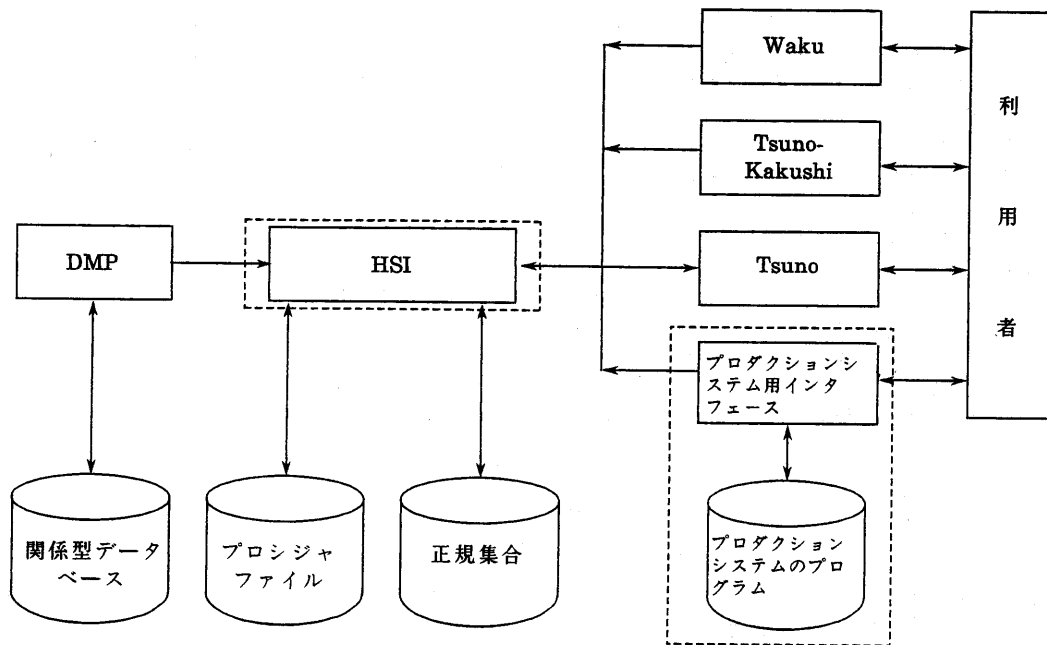


図2. Adbisの構成図

拡張, つまり集合演算とバックトラックのための情報を保持する機能をHSIに追加すればよい。なお, プロシジャ呼び出し機能と否定処理機能はHSIに既に備わっているので新たに作成する必要はない。

## 5. 2 関係型データベースの利用

HSIの関係型データベース検索機能を利用すれば, 関係型データベースとして蓄えられた大量のデータをコンテキストの一部とみなすことができ, プロダクションシステムと関係型データベースとを結合することが可能となる。この場合, コンテキストの初期状態は, 利用者が3. 1節(4)式の形で陽に指定した初期状態と関係型データベースに格納されている関係の組(tuple)集合との和集合と考えることができる。また, プロダクションルールの適用は関係型データベースへの追加とみなせる。

しかし, コンテキストとプロダクションルールの条件部との照合の度に関係データベースをアクセスするのは実用的ではない。そこで, プログラムの初期状態の指定時に, 関係型データベースの検索処理も記述できるようにするとともに, それにより検索された結果集合だけをコンテキストに取り入れることとする。また, プロダクションルールの適用しても関係型データベースへの追加は行わないものとする。つまり, 関係型データベー

スへのアクセスはプロダクションシステムの起動時にだけ行うものとし, それ以降は行わないこととする。

関係型データベースの検索処理は, たとえば, 次のように記述する。

$$\{C_1; \dots; C_r; \text{REL}(x, 'ABC', y) \rightarrow \text{NEWREL}(y, x)\} \quad (15)$$

これは2番目の属性の値が'ABC'であるという条件でRELという関係を検索し, その結果をNEWRELという新しい関係として保存し, それをコンテキストの初期状態とすることを表している。C<sub>1</sub>, ..., C<sub>r</sub>は利用者が陽に指定した初期状態である。

## 6. 考察

本稿で提案したツールの特長および検討課題を以下にまとめる。

- (1) 本稿で提案したプロダクションシステム構築支援ツールの処理方式は, プロダクションルールとコンテキストとの照合をHorn集合の推論として, プロダクションルールの適用をコンテキスト上の集合演算としてとらえる方式であるため, プロダクションシステムを容易に実現することができるであろう。

(2) 本ツールにおけるプロダクションルールのコンテキストへの適用は、ルールの条件部をコンテキストから取り除き、実行部をコンテキストに追加するというただ一つの簡単な動作である。これによってOPS5におけるコンテキスト更新命令であるMAKE(要素の追加)MODIFY(要素の変更), REMOVE(要素の削除)はすべて行うことができる。

(3) OPS5の上記以外の命令もBUILD命令を除いて、プロシジャ呼び出し機能を使って実現できる。BUILD命令は、実行中に新しいプロダクションルールをルールベースに追加するものである。これはルールが直接ルールを呼ぶことになり、プロダクションシステムの精神に反する。また、あらかじめBUILDで作るプロダクションルールをルールベースに定義しておき、その実行条件をプログラム中に陽に指定することにより、効率を考慮しなければ代替できるので、本稿で提案したツールの能力が本質的に低いわけでない。

(4) OPS5では、条件要素の引数に関数を書くことができる。本ツールでは、現在その機能はない。これも、それに相当する組み込み述語をあらかじめシステム側で準備すれば代替できる。しかし、利用者から見るとこれではプロダクションルールの読解性が一般に落ちるので、本ツールでもOPS5のような関数表現記述を許し、それを述語に変換する前処理部をAdbisに組み込む予定である。

(5) 3.5節の(1)で述べたプロダクションルールの選択方法は非常に単純な戦略であるが、バックトラック機構を併用しているので、最初どのルールを選ぶかは効率の面を除けば重要ではないと言える。しかし、効率を上げるためには、問題の性質に応じた制御戦略が必要であり、今後利用者が柔軟に指定できる方式を検討し、本ツールに組み込む予定である。

(6) 本稿で提案したプロダクションルールの構文は条件部と実行部が否定条件を除いて同じ形をしているため、矢印の向きを変えるだけで逆方向の推論が原理的にできる。ただし、矢印の向きを変える前の条件要素にプロシジャが書かれている場合は、そのプロシジャの逆関数が存在しなければならない。

## 7. むすび

プロダクションシステム構築支援ツールを設計するとともに、それをHorn集合推論系を用いて処理する方式を提案した。今後は計画に従い、

これを実現し、その評価を行う予定である。なお、本稿では触れなかったが、プロダクションシステムの実行をトレースし、それを分かりやすく説明する機能や推論の確からしさを扱う機構についても検討を加える予定である。

## 参考文献

- 1) Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., eds.: Building Expert Systems, Addison-Wesley (1983) (A I U E O 訳: エキスパート・システム, 産業図書)。
- 2) Barr, A. and Feigenbaum, E.A., eds.: The Handbook of Artificial Intelligence, Vol.1, Pitman, Los Altos (1982) (田中, 淵監訳: 人工知能ハンドブック第I巻, 共立出版)。
- 3) Davis, R. and King, J.: An Overview of Production Systems, in Machine Intelligence 8, (Elcock and Michie eds.), pp.300-331, Ellis Horwood (1977)。
- 4) 小林重信: プロダクションシステム, 情報処理, Vol.26, No.12, pp.1487-1496 (1985)。
- 5) Brownston, L., Farrell, R., Kant, E., and Martin, N.: Programming Expert Systems in OPS5, Addison-Wesley (1985)。
- 6) Melle, W., Shortliffe, E.H., and Buchanan, B.G.: EMYCIN: A Knowledge Engineer's Tool for Constructing Rule-Based Expert Systems, in Rule-Based Expert Systems, (Buchanan and Shortliffe eds.), pp.302-313, Addison-Wesley (1985)。
- 7) Buchanan, B.G. and Shortliffe, E.H. eds.: Rule-Based Expert Systems, Addison-Wesley (1985)。
- 8) 松尾文碩, 二村祥一, 高木利久: 推論関係型データベース管理システムAdbis, 情報処理学会論文誌, Vol.24, No.2, pp.249-255 (1983)。
- 9) 松尾文碩, 高木利久: データベース操作のためのHorn集合反証器, 情報処理学会論文誌, Vol.25, No.3, pp.458-464 (1984)。
- 10) Nilsson, N.J.: Principles of Artificial Intelligence, Tioga Press (1980)。
- 11) Waterman, D.A. and Hayes-Roth, F.: An Overview of Pattern-Directed Inference Systems, in Pattern-Directed Inference Systems, (Watermann and Hayes-Roth eds.), pp.3-22, Academic Press (1978)。
- 12) Loveland, D.W.: Automated Theorem Proving: A Logical Basis, North-Holland,

