

知識ベースマシン構築のための一考察

横田 治夫 角田 健男 宮崎 収兄 栗山 茂樹 村上 国男
 ((財) 新世代コンピュータ技術開発機構)

1. はじめに

第5世代コンピュータ・システム(FGCS)研究開発プロジェクトでは、知識情報処理に向けたコンピュータ開発のため、推論・知識ベース機構の実現を目標としている。

本報告では、推論・知識ベース機構の基礎となる知識ベースマシンを構築するために、現在開発中の関係データベースマシン(Delta)と逐次型推論マシン(ψ)を中期に向けていかに発展させて行くかの考察を行い、有望と思われる一つのアプローチを提案する。

2. 推論・知識ベース機構

高度な知識情報処理を行うためには、適当な知識の表現方法を定めると共に、その知識を高速に処理する推論機能と、膨大な知識を効率良く管理する知識ベース機能とが必要となる。どのような知識表現が最適であるかは今後研究を進めて行かなければならないが、FGCSではその知識表現を実現するための言語として、Prologを基にした論理型言語 KL (Kernel Language) を開発している。

KL あるいは Prolog のプログラムはホーン節 (Horn Clause) の集合で、節 (Clause) はその形態により以下の3つのクラスに分類することが可能である。

- a) UCG ... Unit Clauses Grounded (without variables)
- b) UCV ... Unit Clauses with Variables
- c) NUC ... Non-Unit Clauses

UCG は引数に変数を含まない1つの正の句 (literal) のみからなる節で、factとも呼ばれ、関係データベースの1つのタプルとみなすこともできる。UCV も1つの正の句のみからなる節であるが、引数の中に変数を含んでいる。NUC は1つの正の句と1つ以上の負の句とからなる節で、UCV とNUC とをあわせてruleと呼ぶ。

現在までに行われているこの分野の研究では、factの集合を関係データベースに格納し、ruleの集合を推論機構部分に持たせて、関係データベースと推論機構の間のインタフェースを取ることによって推論・知識ベース機構を実現しようとしている (評価方式と呼ぶ) [1], [2], [3]。これは、fact型知識の体系化に比べrule型知識の体系化が遅れているため、大容量の知識ベースでは、ruleよりfactの数の方が巨大になるという予測 [2] を基にしている。

しかし、もともとPrologではプログラムとデータを識別せずにプログラムが書けるのに、知識ベースの対象をfactのみとしてデータ部分を意識させるのはPrologの記述力を

弱めることになる。一方、知識ベース中にruleも格納することにより、知識ベースに格納されている知識の柔軟性が非常に増加されることになる。現在知識ベースのアプリケーションの有力な候補とされている自然言語処理用の辞書やコンサルテーション用の知識にしても、factのみではなくruleも大量に含んだものになると予想される。

確かに評価方式は、現在の関係データベースマシンと推論マシンとの開発体制には適合している。しかし、評価方式では、問合せを一括して行うコンパイルド・アプローチにしても、問合せを逐次行うインタープリティブ・アプローチにしても、知識ベース側で検索等の対象となるものはUCG という一系スタティックなものだけである [1], [3]。UCV やNUC のようなダイナミックなものは、知識ベース側に格納するとしても、単なるファイルサーバーのように内容については何の処理も行わないものとなるであろう。このため、評価方式では大層になると予想されるruleに対しては検索能力が十分でない。

そこで我々は、ruleとfactを1つの集合として一緒に持つ非評価方式の1つのアプローチとして、知識ベース機構側に推論機能の一部を取込む方法を提案する。

3. 推論機能の分解

Prologプログラムは、ホーン節の導出 (Resolution) 処理を行うことによって実行される。この導出の処理は、

- ① 与えられたゴール列の中の (一番左の) ゴールを1つ取り出し、
- ② そのゴールと統一可能 (Unifiable) な節の候補をプログラムの中から1つ探してきて、
- ③ その節の帰結部の述語とゴールの統一 (Unification) を行い、
- ④ その結果の代入 (Substitution) を候補の節の条件部の述語とゴール列の残りとに適用してそれぞれの代入例 (Instance) を作って、
- ⑤ その2つの代入例を結合して導出式 (Resolvent) を形成し、
- ⑥ その導出式が空節 (Empty Clause) になるまで、その導出式を新しいゴール列として、以上の導出の処理を再帰的に繰返す、

ものである。ただし、選んだ節の候補が適当でない場合には、バックトラックをかけて他の候補を選ぶことができる。

ここで我々は、知識ベースの検索機能に、統一可能な節

の集合を求めて統一を行い、それぞれの要素の代入を作る機能を盛込むことを考える。このように、知識ベース機構側で統一の機能を持つことにより、知識ベース内に大量の rule を格納した時に、格納された rule に対して効率的な検索を行うことが可能となる。

この時に推論機構側に要求される機能としては、ゴール列の中から1つのゴールを選んで知識ベース側に受け渡す機能と、知識ベースの検索の結果得られた節と代入を組にした集合のうちの1つの要素とゴール列の残りから1つの導出式を作り出して、空節のチェックをしながら（バックトラックを許して）再帰的に繰返す機能である。

このように、推論機能を2つに分解したのは、知識ベース機構側では、主に大規模知識ベースの検索機能を受け持ち、推論機構側ではその検索結果を有効利用することを目的とするためである。

この方法によると知識ベース機構と推論機構の間のやりとりが増える可能性があるため、2つの機構の間はかなり密に結合する必要がある。

4. 知識の格納形態

知識ベースは、基本的には関係データベースの延長上にあると考える。

知識ベース中に格納する対象となる節のうち、UCG がそのまま関係データベースのテーブルに対応することは前にも述べた。UCV については、変数の表現が可能であれば、やはりそのまま関係データベースに格納できる。変数の表現は（Delta では）タグ等で実現することが可能である。問題となりそうなのは NUC であるが、統一の対象となるのは NUC の帰結部の述語のみである。条件部については単なる文字ストリングとしておいても推論機構側で代入例を作ることは可能であるから、帰結部の述語を UCG や UCV と同様に扱えばよい。

以上の考察の結果から、知識ベース中の節は、前述の3つのクラスのいずれに属するとしても、関係データベースの関係と同様に、正の句の述語名ごとにまとめて表として格納することができる。表の各行は、述語の各引数に対応した属性格納エリアを持つと同時に、統一の結果生じる代入を格納するエリアと、NUC の場合に条件部の文字ストリングを格納するエリアからなる。UCG や UCV の場合には、条件部を格納するエリアは空となる。

5. 知識ベース演算

ここで提案した方法によると、知識ベース機構側の機能としては、一般の関係演算の機能や更新機能等の他に、統一のための機能を盛込めばよい。これは、例えば関係演算の Selection コマンドと同様なレベルで Unification というコマンドを用意することによって実現できる。ここで言

う Unification コマンドとは、条件の述語と同一の述語名の表に対して、統一可能な述語を選び、それぞれの述語に対して統一を行い、代入を作成してそれぞれの代入格納エリアに格納し、それらの述語と代入の組を全て取出してくるものである。

Delta を発展させた形で知識ベースマシンの構成を考えると、この Unification コマンドは、他の関係演算系のコマンドと同様に、関係データベースエンジンで階層構造メモリから送られるストリームを処理することにより実行される。

6. おわりに

rule を知識ベースに格納しようとした場合の知識ベース機構の実現方法、及びその時に必要とされる推論機構の機能について提案を行った。

Prolog の並列性は、ゴール列中のゴール間の and 並列と、同一述語名による統一可能な節の候補間の or 並列とに分類される。知識ベース機構側で統一可能な節の集合を探すのは、or 並列に当る。推論機構でストリームを介した and 並列を行うことにより、KL に今後取入れられる予定の並列処理機能の知識ベース機構と推論機構による機能分担が可能となる。

今後検討が必要な問題としては、

- 1) 述語の条件部の実現方法として文字ストリングより適切な方法がないか、
- 2) 変数の表現が（バックトラック発生時などに）本当にタグによるもので適当かどうか、
- 3) 引数に現れる構造体を（マッチングを取り易くするためには）どのように格納するのがよいか、
- 4) 検索の結果得られた述語の集合をうまく取扱う方法がないか、
- 5) プログラム実行時に必要な述語の知識ベース中のローカリティを上げるのにはどうしたらよいか、

などの問題がある。

〔参考文献〕

- [1] S. Kunifuji and H. Yokota: PROLOG and Relational Data Bases for Fifth Generation Computer Systems, Proc. of CERT Workshop on "Logical Bases for Databases", CERT, Dec. 1982.
- [2] 田中 隆、他：推論システムとデータベースシステムとの部分評価機構による結合、計測自動制御学会、第一回知識工学シンポジウム、東京、昭和58年 3月
- [3] U.S. Chakravarthy, et al.: Interfacing Predicate Logic Languages and Relational Databases, Proc. of the First Int. Logic Prog. Conf., Faculte des Sciences de Luminy Marseille, France, Sept. 1982.