知識処理向き並列推論メカニズム

北上　始，横田治夫，服部　彰

富士通株式会社

1．はじめに

知識処理システムでは、今後、大規模なデータを扱うような分析や診断問題・膨大な組み合わせを扱うような計画や設計問題・多大な計算を必要とするようなシミュレーション問題などを実時間で解く要求が高まってくるものと考えられている。これに答えるために、並列マシンの力を効率良く引き出すアプローチで、知識処理を高速化する研究が進められている。

知識処理は、人間の知識をデータとして知識ベースに登録し、①知識ベースから必要なデータを検索した後、②それを分析・加工する処理と見なすことができる。従来の並列処理では、これらを並列化するために、OR並列に力点を置く研究と AND並列に力点を置く研究が別々に進められてきた。

本報告では、知識処理システムをOR並列と AND並列の両方で並列化することに着目し、知識処理システムの代表的な機能である全解探索用推論エンジンを OR 並列と AND並列の両方で並列化する方法とその測定結果について報告する〔1〕。また、ここで使用したマシンは、並列マシンの実験用システムとして使っているSEQUENT 社のSymmetry(8 台のプロセサが共有メモリーで接続された構成)でありインプリメンテーション言語は、Commited-Choice 型言語 GHC 〔2〕である。

GHC は単一代入言語であるので、Prologのバックトラック機構に対応する変数の書き換え機構を持たない。これにより、知識ベースを持つ推論エンジンの実現が難しいとされていた。ここでは、GHC に新しいデータタイプとして、$ 変数を導入する事により、OR並列とAND 並列を旨く組合せる方法について述べる。

本研究は、制約型論理プログラミング言語・知識獲得や学習などで必要とされる知識ベース管理・確信度付きの推論エンジン・時間論理を持つ推論エンジン・意味ネットワーク用の推論エンジン・プロダクションシステム用推論エンジンなどを GHCで十分な並列効果を出しながらプログラミングする際の基本的な方式を提供していると言える。

2．全解探索用推論エンジン

知識処理は、知識ベース検索と検索された知識の分析・加工から構成されるが、これらの処理をまとめた例として論理型プログラミング言語 Prolog 〔3〕による推論エンジンの研究がある。このような推論エンジンを使うと、取敢えず解答を1つ求めることができるが、この処理は探索木のある一本の経路を辿る処理なので、沢山の経路を同時に辿るような意味での並列性がない。従って、これを拡張し、全解を求めるための推論エンジンbagof-solve を実現すると、それは、並列に複数解を求める処理として並列化が可能である。

2．1　並列化の問題点

全解を求める推論エンジンの並列化を GHCで行うためには、知識ベース検索機構として次の処理が必要である。

①ストリームで解答をデータ転送

②必要なデータを探すときの単一化処理

③変数名の新規作成

①の処理は、GHC がもつ機能を利用することにより実現できるが、②の処理は、GHC の単一代入制限により、実現できない。即ち、2件以上のデータを探すことが出来なくなる。また、③の機能は、GHC に存在しない機能であり、新機能として作成する必要がある。

以上から明らかなように、②が実現上の大きな問題点になる。この処理で、単一代入制限を回避する方法として、GHC に項のコピー機能を持たせる方法があるが、この方法では、項の中の変数をコピーする時に変数の排他制御が必要であり、オーバーヘッドが大きいと考えられる。

この問題点は、全解探索用の推論エンジンにおいて、複数のデータを分析していく過程でも生じるので、解決しなければならない点である。

2．2　解決策

変数を含む項のコピーはオーバーヘッドが大きいが、変数を含まない項のコピーは特に問題が生じない。これに着目し、データを表現する変数を特殊な定数で表現し、GHC の変数と区別する方法を考えた。この特殊な変数を$変数と呼び、知識ベースのデータ表現や知識ベース検索の条件指定の表現に利用することにした。図1に知識ベースの表現例を示す。知識ベース検索の条件指定は、次のように表される。

```
?- bagof-clause(ancestor($10, $11), Stream).
Stream= [(ancestor($12, $13):-parent($12, $13)),
         (ancestor($14, $15):-...)]
```

Knowledge Processing Oriented A Parallel Inference Mechanism

Hajime Kitakami, Haruo Yokota, Akira Hattori

Fujitsu Limited

データの変数を$変数で表すと、項の単一化処理をもはやGHC の単一化命令で行うことが

```
ancestor($1,$2):-
    parent($1,$2).
ancestor($1,$2):-
    parent($1,$3),ancestor($3,$2).
parent(f1,f2).
parent(f2,f3).
parent(f3,f4).
```

図１．知識ベースの例

出来なくなるので、以下のような$ 変数を含む項を扱うための命令をGHC で実現した。

① unify(TM1, TM2, NewTM, OutBinf)

TM1(項)とTM2(項)を単一化し、その結果をNewTM に返す。 OutBinfには、単一化の際に行われた$ 変数の代入状況を返す。単一化が終わっても、TM1 とTM2 の$ 変数は、書き換わることがない。

② substitute(InBinf, TM, NewTM, OutBinf)

InBinfの代入情報をもとにTM内$ 変数を書き換えNewTM にその結果を返す。OutBinf にはその時に行われた代入状況を返す。代入が終わっても、TM1 とTM2 の$ 変数は、書き換わることがない。

③ generate-newVL(InVList, OutVList)

InVList 中の$ 変数要素分だけ、新しい$ 変数を生成しそれらをOutVListに返す。

3．システム構成

図２は、GHC で作成した全解探索用の並列推論エンジンの構成図である。図２中の並列推論ドライバーは、与えられた問題（ゴール列）をOR並列や AND並列で解くための指示を行う。並列推論ドライバーがゴール列を受け取ると、それをAND 並列処理部に渡す。

AND 並列処理部では、ゴール列を分解し、ゴールを並列推論ドライバー経由でOR並列処理部に渡し、ゴールを実行する（OR並列処理部での実行の結果、複数の解答が得られる）。ゴール列の中に、p($1),q($1,$2)で示される共有変数($1)があるとp($1) の複数解(例: $1=1,2,3,...)をq($1,$2)に伝達しなければならないので、そのための処理を行う。以上により、ゴール列の複数解が求まるので、それらをマージする処理が行われる。

OR並列処理部では、並列推論ドライバーから渡されたゴールと単一化するデータ（知識）を知識ベースから検索し、検索結果として得られる複数のデータをストリームで受け取る。その後、個々のデータを解釈するために、個々のデータを並列処理できるように並列展開を行う。

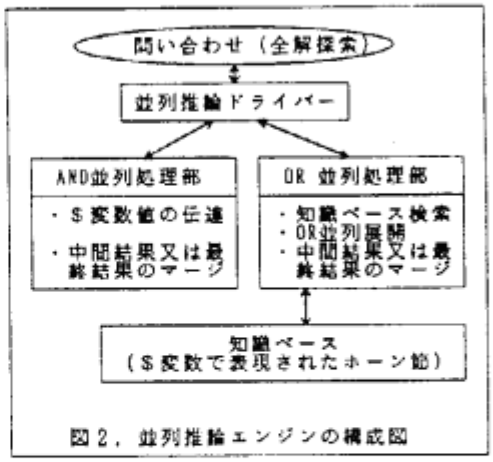

図２．並列推論エンジンの構成図

このようにして並列推論ドライバーは、OR並列と AND並列により、並列推論を進め、ゴールが無くなるまで実行を進める。求められる結果は、以下の組の集合となる。<解答，$変数の代入状況，推論履歴等>

4．測定結果

図１の知識ベースを使い、anc($10,$11)の全解を求める問い合わせの台数効果を測定した。この知識ベースでは、OR並列処理部で２１回の知識ベース検索の要求が出されており、８６%以上がOR処理であることが分かった。この問題では、８PEで、６倍程度の台数効果を得た。さらに、この問題は、データ量の増大に伴い、並列度が高くなることに着目し、並列性能がどの程度上がるかを測定した。測定結果の詳細については、発表時に報告する。

5．おわりに

GHC を使い、OR並列処理部とAND 並列処理部をもつ全解探索用の並列推論エンジンを試作し、並列効果が出ることを確認した。しかし、$変数を含む項の単一化や代入は、GHC 自身で書いているため、知識ベース検索のスピードに問題がある。また、知識ベースのデータ量が増大するにつれ、益々、重要な問題になってくる傾向がある。 現在、この知識ベース検索機能を、GHC よりも下位の言語で作成しており、これとGHC との結合を行っている。

〔 謝辞 〕

本研究に当たり、日頃から有益なコメントを下さったICOT第三研究室・伊藤室長、㈱富士通研究所・林人工知能研究部長、ICOTのKBM メンバーの方々に深謝致します。

〔 参考文献 〕

〔１〕北上，横田，服部：知識処理向き並列推論エンジン，CPSY88-50 (1988).

〔２〕淵 監修，並列論理型言語ＧＨＣとその応用，知識情報処理シリーズ６，共立出版 (1987).

〔３〕Bowen, D.L.: DEC-10 PROLOG USER'S MANUAL, University of Edinburgh (1981).