7D-3

汎用エキスパートシステムの高速化技術 (II) 推論方式

島田 優　　黒沢 憲一　　平山 洋一
(株)日立製作所　日立研究所

1．はじめに

エキスパートシステムの基本的なアーキテクチャであるプロダクションシステムは、システム規模が大きくなると、照合回数が爆発的に増加し極端に推論性能が低下する欠点がある。そこで照合回数を大幅に減すＲＥＴＥアルゴリズムが提案された。しかしこのＲＥＴＥアルゴリズムをベースとしたプロダクションシステムでも、十分な性能が得られないという問題が生じてきた。この問題に対して本報告では処理量評価に基づく新たなアルゴリズムを提案する。なお以下では知識の表現をプロダクションルールとフレームを用いた前向き推論システムについて話しを進める。

2．プロダクションシステムの概要

プロダクションシステムでは以下の認知－行動サイクルを繰り返すことにより推論を実行する。

1）照合：　全てのルールの条件部（ＬＨＳ）とフレームとのパターンマッチを行い、実行可能なルールとフレームの組合せ（ｲﾝｽﾀﾝｼｴｰｼｮﾝ）を全て求める。

2）競合解消：　全インスタンシエーションの中から予め指定した選択基準（戦略）に基づいて一つだけ選びだす。

3）動作：　選びだされたインスタンシエーションのルールの実行部（ＲＨＳ）に従い実行を行う。この時一般にはフレームのスロット値が変更される。

この原理に基づき実行を行った場合、照合フェーズでのパターンマッチの回数がルール及びフレームの増加に従い爆発的に増えてしまう。この問題点を解決するために生れたのが Forgy によるＲＥＴＥアルゴリズムである[1]。

3．ＲＥＴＥアルゴリズム

ＲＥＴＥアルゴリズムは現在大部分の商用ツールで基礎とされている高速化アルゴリズムである。このアルゴリズムはまずルールの条件部をＲＥＴＥネットワークと呼ばれる一種の弁別ネットに変換し、トークンとしてフレームをこのネットワークに流してネットワークの状態させ照合を行う。

ＲＥＴＥアルゴリズムの特徴としてパターンマッチの回数を減らすために前サイクルでの照合の途中結果を中間結果として全て保存しておくことが挙げられる。そして次のサイクルでの照合は、ルールの実行により変更されたものに関してだけパターンマッチを行い、無駄なパターンマッチを減らしている。これら中間結果には、

1）ルール中の一つの条件節内で完結したパターンマッチの結果を保存するαメモリ

2）ルール中の複数の条件節に渡って、変数の値の整合性を判定した結果を保存するβメモリ

の２種類がある。また複数のルール間で共通な条件を共有させてパターンマッチを減らしている。

以上のようにＲＥＴＥアルゴリズムでは大幅なパターンマッチ回数の減少が期待出来る。しかし実際には期待したほど性能が向上しない場合がある。その原因を探るため、まず実際のエキスパートシステムではどのようなルールを用いているのか、約１００ルール程度のエキスパートシステムについて調べてみた。その結果を図１に示す。これによると、条件部の約半数のフレーム条件節で照合対象となるフレーム名称が変数で指定されていた。これは各フレーム条件節毎に多数のαメモリを持つことを表している。また条件部に表れる変数の約半数が、複数の条件節に出現している。これは多くの場合照合時に他の条件節との間で変数値の整合性を調べる必要が有ることを示している。そのため多数のｊｏｉｎ操作が行われることになり、推論時に多量のβメモリ操作が行われることを示している。そこでＲＥＴＥアルゴリズムをベースとしたインタプリタ型推論エンジンついて、典型的なルールについてのベンチマークを実行し、その照合時に実行した命令数の内訳を実際に調べてみると、図２に示すように、スロット条件判定等のパターンマッチは照合全体の３０％程度であり、その他は中間結果管理のためのリスト処理等によるメモリ管理であった。以上のようにＲＥＴＥアルゴリズムでは照合回数の低下に対するメモリ管理のオーバーヘッドというトレードオフが存在し、これが推論性能の向上を妨げている。

	フレーム指定			変数	
ルール数	フレーム名指定	クラス名指定	不定	条件内変数	条件間変数
92	62.6%	37.4%	-	81.0%	19.0%
96	13.3%	86.7%	-	68.1%	31.9%
106	53.6%	36.7%	9.7%	41.8%	58.2%

図１　ルール表現分析

4．アルゴリズム改良方針

以上のようなＲＥＴＥアルゴリズムの問題点に対し、近年 Miranker によりＴＲＥＡＴアルゴリズムが提案されている[2]。これはＲＥＴＥアルゴリズムにおけるβメモリを保持せずにαメモリのみを保存する。その代わりに動的なＪｏｉｎ最適化によって各サイクル毎に必要なインスタンシエーションを作り出す。ＴＲＥＡＴアルゴリズムはこのβメモリの廃止によって動的に行うメモリ管理の量を減らし、照合の高速化を図っている。しかし依然αメモリは残っており、そのメモリ管理及びＪｏｉｎ操作の手間は大きなものになる。

そこで我々は、先に見たようにパターンマッチによる照合コストと、中間結果操作によるメモリ管理のコストがトレードオフの関係に有ることに着目し、これら２つのコストのバランスを取ることにより、全体としての照合時間を低下させることにした。

RETEｱﾙｺﾞﾘｽﾞﾑに基づくインタプリタの実行命令数内訳

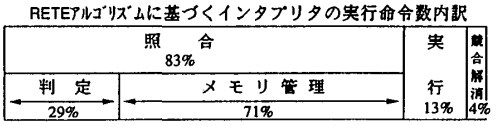


図２　実行命令数分析

A Performance Improvement Technology in Expert Systems (II) Algorithm
Masaru Shimada, Ken'ichi Kurosawa, Hirokazu Hirayama
Hitachi Research Laboratory, Hitachi, Ltd.

5．新アルゴリズム

以上の方針に従った我々の新しいアルゴリズムは、
1）中間結果の有効活用
2）照合コストとメモリ管理コストのバランスを取る
の2つをポイントとしている。

まず中間結果の有効活用であるが、ＲＥＴＥアルゴリズムのβメモリはαメモリの組合せによる判定であるためβメモリの操作には多くのコストを必要とする。そして複数のフレーム条件節に渡る照合判定の結果を残す場合、αメモリの管理と重複してしまう。しかしＴＲＥＡＴアルゴリズムのように一切βメモリを持たないようにすると、今度はサイクルごとのＪｏｉｎ演算コストが大きくなってしまう。そこで中間結果を保存した場合これをより有効に利用出来る保存単位を考える。従来のαメモリは言わば1つのフレーム条件節単位で管理する中間結果であり、βメモリは2つ以上のフレーム条件節単位で管理する中間結果であった。我々は、同一の変数が出現するフレーム条件は一つの大きな条件項であると考え、1つ以上の独立性の高いフレーム条件節単位に中間結果を保持することにする。即ち一度保持した中間結果は、他のフレーム条件節の影響をなるべく受けないような単位で中間結果を残すことにより、できるだけ中間結果の変更を少なくする。フレーム条件節間の影響は変数によって伝播して行く。そしてこの変数によって互いに密に結合しているフレーム条件節や、互いに疎に結合しているフレーム条件がある。この変数への代入と変数の参照による値の伝播方向に着目して、図3に示すようにルールの条件部を各フレーム条件節をノードとし、変数による値の伝播を有向辺とした有向グラフを作る。そしてこのグラフで強連結成分を調べ、それぞれの強連結成分をグループにまとめる。このグループを中間結果の保持単位とすることにより中間結果の独立性を増し、より有効な中間結果の利用が期待出来る。

次に照合コストとメモリ管理コストのバランスであるが、先の強連結成分グループにおいては中間結果を保持して管理を行う場合に比べ、毎回パターンマッチを行った場合の処理量の方が小さい場合、中間結果を保持するより直接フレームとのパターンマッチを行わせた方がルール全体としてより高速な推論が実行できる。そこで予め各グループ毎に更新されたフレームと各条件節がパターンマッチを行う確率を基に、直接フレームとのパターンマッチを行った場合の照合コストと、グループが中間結果を保持した場合の中間結果の生成，削除及びその中間結果の参照等の処理に要するメモリ管理コストを評価しておく。そして照合コストの方がメモリ管理コストよりも大きいグループについてのみ中間結果を保持するようにし、メモリ管理コストの方が照合コストよりも大きいグループの場合、再照合時にはフレームと直接パターンマッチを行う動作を割り当てる。例えばパターンマッチ対象のフレーム名が明示してあるものは中間結果の数が制限され、メモリ管理コストが小さいため一般に中間結果を保持する。一方フレーム名が変数で指定されているものは一般に中間結果の数が増えるため、条件節の判定内容が容易で一回当たりのパターンマッチに要する処理量が小さく、更新フレームとのパターンマッチを行う確率が大きい条件節を含む場合には中間結果を保持しない。これにより図4の例に示すようにグループ1は中間結果を保持し、グループ2は中間結果を保持しないとすると、グループ1に属するフレーム条件とパターンマッチ可能なフレームが変更された場合の再照合では、グループ1，2ともに直接パターンマッチを行う。この時グループ1の中間結果は更新される。反対にグループ2に属するフレーム条件とパターンマッチ可能なフレームが変更された場合の再照合ではグループ2のみ直接パターンマッチを行い、グループ1に関しては中間結果の参照を行うだけとなる。このように照合コストとメモリ管理コストのバランスを取ることにより全体としての照合は、全て中間結果を保持した場合や一切中間結果を保持しない場合に比べて高速になることが期待できる。

6．終りに

プロダクションシステムの代表的な高速推論アルゴリズムであるＲＥＴＥアルゴリズムの問題点を明らかにし、この問題点を解決するために強連結成分単位にグループ分割し、照合コストとメモリ管理コストをバランスさせる新しい高速推論アルゴリズムを提案した。今後、さらにＴＲＥＡＴアルゴリズムとの比較、及びより高精度な処理コストの評価方法の確立が必要である。また実際の性能を他のアルゴリズムと定量的に比較していく予定である。

参考文献

[1] Forgy,C.L.:RETE:A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artificial Intelligence,Vol.19,pp.17-37(1982)
[2] Miranker,D.P.:TREAT:A Better Match Algorithm for AI Production Systems,AAAI-87,pp.42-47(1987)
[3] 石田 亨,桑原 和宏:プロダクションシステムの高速化技術,情報処理,Vol.29,No.5,pp.467-477(1988)
[4] 平山 洋一,黒沢 憲一他:汎用エキスパートシステムの高速化技術(I)-(III),第39回情報処理全国大会,平成元年10月

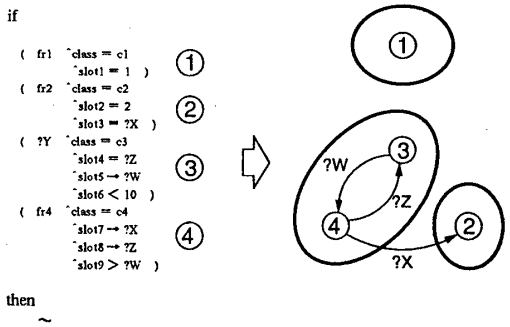
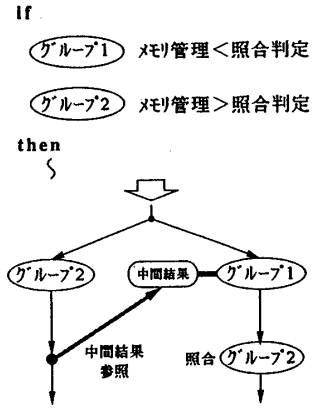


図3　グループ分割



第4図　処理量判定