

ルールの間接的関連性を利用した パターンマッチングアルゴリズムについて

藤井 和之^{*} 上野 晴樹^{**}

^{*} 清水建設 情報システム部 ^{**} 東京電機大学 理工学部

前向き型のプロダクションシステムの推論方法は認識-行動サイクルと呼ばれ、次のものである。

- (1) ルール群と作業領域の要素とのパターンマッチングを行い実行可能なルールを競合集合に入れる。
- (2) 競合集合のルール群から適当な競合解消を行い、実行ルールを1つ選択する。
- (3) ルールを実行し、作業領域の内容を変更する。

つまりあるサイクルで実行されるルールは次のサイクルの競合集合にはなんらかの関連性があると考えられる。これをルールの間接的関連性と呼ぶ。

本報告では、このルールの間接的関連性の利用と、Reteに基づく判別ネットワーク手法とを組み合わせたルールコンパイル手法を提案する。

A PATTERN MATCHING ALGORITHM USEING INDIRECT RELATIONSHIPS AMONG RULES

Kazuyuki FUJII^{*}, Haruki UENO^{**}

^{*} System Developmet and Data Processing dept., SHIMIZU CONSTRUCTION CO.,LTD
16-1,Kyobashi,2-chome,Chuo-ku.Tokyo,Japan

^{**} Tokyo Denki University

This paper discribes a new method for rule compilation by means of the combination of sorting network techniqes based on the Rete-mach algorism and the usage of indirect relationships among rules. The reasoning procedure called as "recognize-act cycle" for pure production systems consists of three steps as follows:

- (1) to select a set of rules by pattern matching against the working memory elements, and to assign them to a conflict set,
- (2) to choose one rule from the conflict set by conflict resolution processing, and
- (3) to execute the RHS of the rule to update the contents of the database, and to return to step 1.

It should be stressed that a rule to be executed at a cycle has a specific indirect relationship to a conflict set for the following cycle, and this indirect relationship has an important rule to minimize the excution time for pattern matching processes are presented in the paper.

1. はじめに

前向き型のプロダクションシステム (PS) の高速化手法の1つとしてルールのコンパイルがあげられる。特に Rete マatching アルゴリズム [1] はその基礎となっているものであり、OPS 5、ART をはじめとして多くのシステムが採用している。また、より高速な処理を目的とした Rete の改良研究 [2、3] も行われている。これは、主にネットワークの探索効率に関する研究である。従って、確実に効率かほはできるが、限界があると考えられる。大幅な効率化を行うためには、Rete 自信の見直し、新しい概念の導入を行うことが必要である。

本論文で提案する Matching アルゴリズムは、現在東京電機大学 (理工) の上野研究室で開発中の PSX (Production System eXtension) で採用されているものであり、

(1) Rete の考え方にに基づく判別ネットワーク手法

(2) ルールの間接的関連性の利用

という2つのコンセプトから成る。この手法を便宜上 PSX アルゴリズムと呼ぶ。また前者をステップ1、後者をステップ2と呼ぶことにする。

2. 制約条件

PSX アルゴリズムは、作業領域 (WM: Working Memory) にフレーム型のデータ構造を用いることを前提としている。従って Rete アルゴリズムと異なり、同じ名前のクラス名 (PSX ではフレーム名) をもつ要素 (WME: Working Memory Element) が WM 中に複数存在することは考慮されていない。

以下、この制約条件のもとに論法を進める。

3. ステップ1 — 判別ネットワーク手法 —

3.1 Rete の考え方と短所

Forgy は論文中に Rete アルゴリズムの2つの基本的な考え方を示している [4]

(1) 各認識-行動サイクルにおいて WM はそのわずかな部分のみが変化することが多い (一時的冗長性)。従って、前サイクルまでの状態を保持することにより、変化していない WME とのパターンマッチングを行う無駄を排除することができる。

(2) ルール群には類似する条件が多く出現する (構造的類似性)。従って、ルール群の条件部 (LHS) の共通部分をあらかじめ抽出し、一度にパターンマッチングを行うことで、同一のパターンマッチングを行う無駄を排除することができる。

Rete では、これらに基づいてルール群の LHS をネットワークで表現しパターンマッチングを行うことを提案している (図1)。ところが、この時点で基本的な考え方に対して欠損が生じており、それが Rete の短所となって現れている。それらは次である。

(i) 一時的冗長性を満足していない

Rete ネットワークにおいて前サイクルの状態は最終ノードで保持される。従って、

(B ↑ B1 V3 ↑ B2 V4 ↑ B3 V5)

というWMEが

(B ↑ B 1 V 3 ↑ B 2 V 4 ↑ B 3 V 6)

に変更された場合、実際に変化したのは属性B 3の値のみであるにもかかわらず

< - (B ↑ B 1 V 3 ↑ B 2 V 4 ↑ B 3 V 5) >

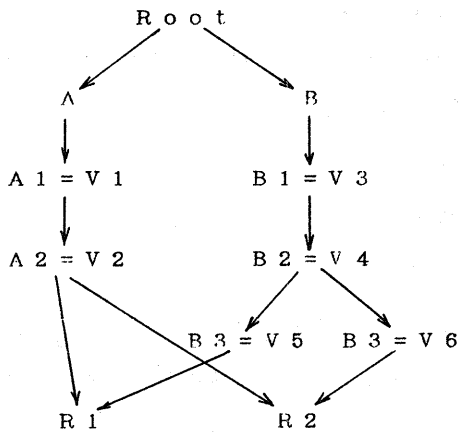
< + (B ↑ B 1 V 3 ↑ B 2 V 4 ↑ B 3 V 6) >

というトークンがネットワークへ送られることになる。つまり、変化していない属性B 1とB 2の削除のテスト、追加のテストが無駄に行われることになる。

(R 1 (A ↑ A 1 V 1 ↑ A 2 V 2)
 (B ↑ B 1 V 3 ↑ B 2 V 4 ↑ B 3 V 5)
 --> . . .)

(R 2 (A ↑ A 1 V 1 ↑ A 2 V 2)
 (B ↑ B 1 V 3 ↑ B 2 V 4 ↑ B 3 V 6)
 --> . . .)

(a) ルールR 1とR 2



(b) R 1とR 2のReteネットワーク

図1 ルールとReteネットワーク

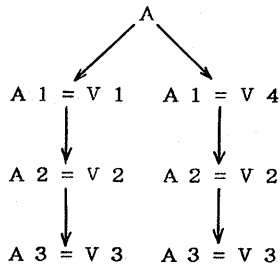
(ii) 構造的類似性を満足していない

Reteネットワークは規則の条件を1単位として構成される。またどのように展開されるかはコンパイラに依存する。つまり次の2つの条件

(A ↑ A 1 V 1 ↑ A 2 V 2 ↑ A 3 V 3)

(A ↑ A 1 V 4 ↑ A 2 V 2 ↑ A 3 V 3)

は



に展開される可能性がある。従ってこの場合、この条件間で属性 A 2 と A 3 のテストは同じであるにもかかわらず別々に行われる。このような事例はルール数の増大にしたがい多く現れると考えられる。

さらに、Rete のインプリメント上の短所として指摘されているものとして、

(iii) 縦型探索による処理効率の低下

があげられる。これは、ネットワークをアセンブラコードで表現実行するために生じるものである。探索はノードのテストが失敗するまで深く行われる。またあるノードのテストが成功しても同じ深さの他のノードのテストまで行われる。

以下で述べるステップ 1 (ネットワーク手法) ではこれら短所に対応するよう (Rete の考え方により忠実に) 考慮されている。

3. 2 ステップ 1 の基本的な考え方

ステップ 1 は Rete の考え方に次を追加する。

・知識の排他性の利用 (福岡工大 荒屋)

ここでは次のように定義する。

『あるフレーム名テストが成功したら他のフレーム名テストは必ず失敗する。同様にあるスロット値の EQVAL テストが成功したらそのフレームの他の EQVAL テストは必ず失敗する。』

3. 3 改良ネットワーク手法

改良ネットワークは、構造的類似性を満足するために、属性ごとにネットワークを構成する。またその属性ごとに前サイクルまでの状態を保存するラベル (SN: Slot Number) を、条件の前サイクルまでの状態を保存するラベル (CN: Condition Number) を設け、一時的冗長性を満足するようになっている。またルールの LHS の条件を CN に変換した変換ルールをネットワークとは別に設け、ルールの LHS が成立するか否かのテストは、これを用いて行う。これによって複雑な LHS への対応ができるようになっている。以下図 3 に図 1 の (a) のルールの改良ネットワークを、図 4 に変換ルールを示す。

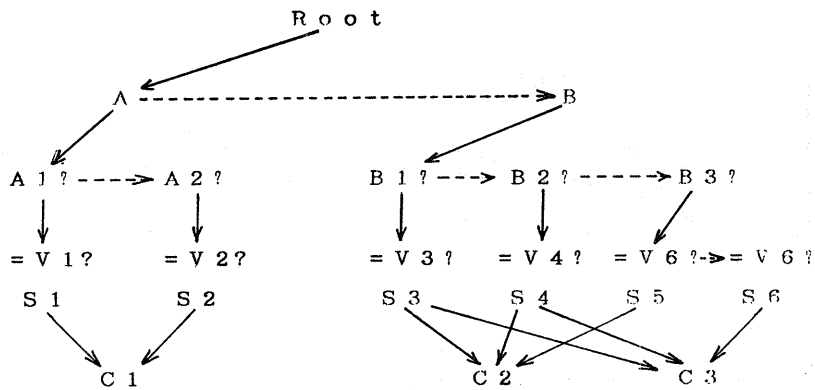


図3 ステップ1ネットワーク

(R 1 (AND C 1 C 2) --->)
 (R 2 (AND C 1 C 3) --->)

図4 変換ルール

図3でS1～S6がSNであり、C1～C3がCNである。点線は排他リンクを示す。

また、PSXアルゴリズムではネットワークを高級言語（現在はLisp）に変換実行する。これによりインタープリターへの負担を軽減する。

3. 4 改良ネットワークの長所・短所

以下にこのネットワークの長所短所を述べる。

長所

1. 一時的冗長性をより満足している。従って前述（3. 1）のWMEの変更のトークンも

< - (B ↑ B 3 V 5) >
 < + (B ↑ B 3 V 6) >

でよいことになる。これはテスト数の減少に貢献する。

2. 構造的類似性をより満足している従ってルール数の増加に対するノード数の増加が少ないことを意味する（図5）。

短所

1. 条件の成立を判定するプロセスがRet eに対して複雑である。同様にルールを導出する過程が複雑である。

これはテスト数の増大を示す。

これらの長所・短所を考えると、ルール数が少ない時は条件のテストがダイレクトに行えるRet eの方が有利である。しかし、推論実行時にはWMEの変更命令がかなりの比重をしめると考えられ、ルール数が増大する程PSXアルゴリズムが有利になる。

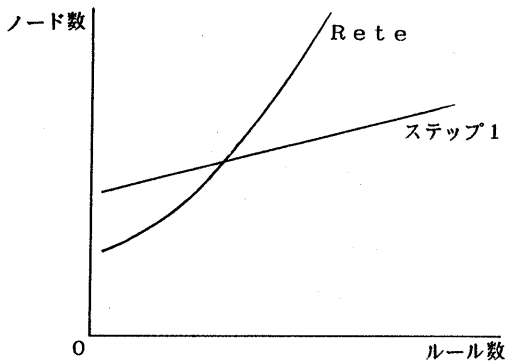


図5 Reteネットワークとステップ1のルール数に対するノードの変化(予想)

4. ステップ2 — ルールの間接的関連性の利用 —

4. 1 ルールの間接的関連性

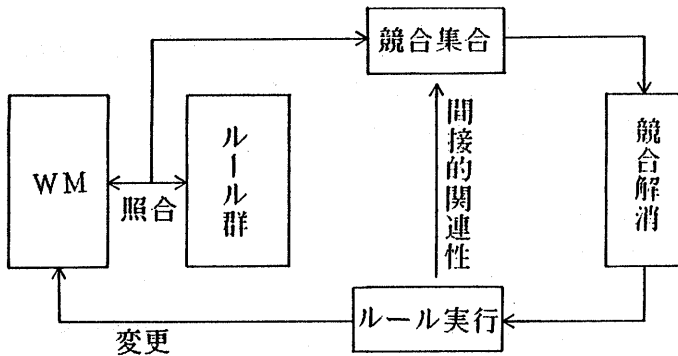


図6 認識-行動サイクルとルールの間接的関連性

前向き型PSの推論は認識実行サイクルと呼ばれ図6に示すものである。この図からわかるように、あるサイクルで実行されたルールによりWMが変更される。そのWMとルール群とのパターンマッチングを行い次のサイクルの競合集合が導かれる。つまり、これらには間接的な関連性があると考えられる。ステップ2ではこれらの間接的な関連性をより直接的な表現にすることでパターンマッチングの効率化を計る。つまりルールの実行部(RHS)に関するコンパイルである。

さて、ルールのRHSに用意されるWM変更情報は通常次の3つである。

1. 追加 WMに新しい情報を追加する。
2. 削除 WMから要素を削除する。
3. 変更 WMの要素の内容を変更する。

これらに対する処理としては次が考えられる。

・追加に対する処理

通常、変数を用いた値以外は、追加されるWMEは明示されている。それらのテストはコンパイル時にある程度行うことができる。この結果はRHSに明示する。

・削除に対する処理

この場合、クラス名、スロット名が明示されている。従ってネットワーク手法と組み合わせ、トークンをネットワークの途中フレームテストの後へ送ることでテスト数を減少させることができる。

・変更に対する処理

変更は、削除と追加の組合せで処理を行う。

これらは変換ルールで表現する。以下、実際の表現方法を示す。

4. 2 ステップ2の表現方法

追加に関する処理は、ラベルの利用により、大幅に簡略化を行う。

(# TRUTH (TになるSN) (TになりうるCN))

また変数を用いている場合は、

(# VT WMEの変数を用いた部分)

という形でネットワークの途中へ直接トークンを送る。

次に削除の場合は、

(# FOURTH フレーム名 スロット1 スロット2 . . .)

削除命令

(# MOD フレーム名 (スロット1 値) (スロット2 値) . . .)

変更の削除部

という形で行う。

4. 3 ネットワーク手法のみとのアルゴリズムの比較

図7にステップ1のみとステップ1にステップ2を加えたものとのテストの流れを比較する。ここに示すように処理が簡略化されていることがわかる。

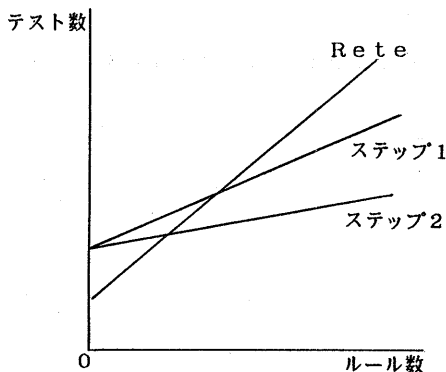


図7 ReteとPSXのテスト数の比較(予想)

5. Reteとの比較

ルールの間接的関連性の利用により、テスト数はステップ1のみよりよりも減少することが予想される。しかしネットワークの構造からルール数が少ないときは依然としてテスト数はReteの方が少ないことも考えられる。しかし、ルール数の増大に対するテストの数の増大は小さいことが考えられるため、図7の様な傾向があらわれることが予想される。

6. おわりに

PSXアルゴリズムはまだ荒削りなアルゴリズムでありいくつかの問題点をかかえている。さいごにそれらの課題を示す。

問題点

1. マッチングインタプリターが2通り（ステップ1とステップ2）必要なため冗長が生じる。
2. フレームに対する変数の使用が考慮されていない。

今後の課題

・同じ条件（同じ言語）上に構築したReteアルゴリズムとの大規模なモデルでの比較検証

参考文献

1. Forgy, C.L. ; Rete : A Fast Algorithm for the Many Pattern / Memory Object Match Problem, Artificial Intelligence, Vol.19, pp17-37, (1982).
2. 田野, 増位, 船橋 ; 推論高速化のための弁別ネットワークの動的変形法, 情報処理学会全国大会第33回, pp1417-1418, (1986).
3. 荒屋真二 ; 知識の排他性を利用したパターン照合アルゴリズム, 情報処理学会研究会知識工学と人工知能 49-4, (1986).
4. 伊藤秀昭 ; OPS5高速化, Computer today, NO.13, pp47-53, サイエンス社, (1986).
5. 藤井和之 ; 汎用プロダクションシステムPSXの研究, 昭和61年度東京電機大学大学院修士論文, (1987)
6. 上野晴樹 ; 知識工学入門, オーム社, (1985).