ルールベースの連想ＲＥＴＥネット表現

5H-7

荒屋 真二
福岡工業大学　工学部　通信工学科

１．まえがき

プロダクションシステムの高速化のためにルールベースを構造化する研究として，ルール条件部の類似性に関する知識を用いたＲＥＴＥネット[1]，排他性に関する知識も加味した排他ＲＥＴＥネット[2]などがある．これらはルール条件部の構造化である．また，ルール動作部と条件部の関係を利用した構造化にペトリネット方式[3]がある．一方，練習により問題解決速度が向上するという認知モデルで使用されているルール合成も[4]，プロダクションシステムの高速化に役立つ．

本論文では，ＲＥＴＥネット方式とペトリネット方式の両者の長所を合わせもち，かつルール合成と同じような効果をもつ連想ＲＥＴＥネット方式を提案する．連想ＲＥＴＥネットは，ルール相互の支持関係に関する知識を用いてＲＥＴＥネットを更に構造化したものであり，ＲＥＴＥネットをサブネットとして包含する．本方式はトークンをＲＥＴＥネットのルートノードからではなく，途中のノードから流すため，ルール条件部の照合範囲が局所化される．

以下では，連想ＲＥＴＥネットの概略を述べると共に，上述の既存手法との比較により本方式の有用性を示す．なお，本稿ではプロダクションシステムとしてＯＰＳ５を対象とする．

２．連想ＲＥＴＥネット

ＲＥＴＥアルゴリズムでは作業記憶（WM）の変化から競合集合の変化だけを求めるために，次の２種類の知識を利用してルールベースをＲＥＴＥネットの形に構造化する．

1)全ルールの条件部を構成する条件パタン間の類似性に関する知識
2)現在のWM要素によってルール条件部のどの範囲が支持されているかに関する知識

図１は，連想ＲＥＴＥネットとＲＥＴＥネットの一部を略記したものである．今，あるルールが発火し，その動作項（make C77 ^A2 3 ^A7 RED ^A3 <X>）が実行された場合を考える．変数<x>には７が束縛されているものとする．このとき，動作パタン（動作項からmakeを除去したもの）のインスタンス（C77 ^A2 3 ^A7 RED ^A3 7）がトークンとして流される．

まず，ＲＥＴＥネットでの処理を考える．トークンは n1+n2+n3+n4 回以上の失敗照合と３回以上の成功照合の後，２入力ノードA3=A9 の左ノードメモリM_Lに格納される．右ノードメモリM_Rが空のときにはバックトラックしながら n5+n6+n7+n8 回以上の失敗照合が行われ，トークン処理が完了する．推論実行時に以上のような照合を行うのは明らかに無駄である．なぜなら，変数<x>の値が確定していない推論実行前の時点で，この動作パタンの全てのインスタンスがノードメモリM_Lに必ず到達することはわかるからである．

連想ＲＥＴＥネットは，このような支持関係に関する知識をあらかじめ抽出し，動作パタンから，トークンを流し始めるノード（連想ノード）へのリンク（連想リンク）を付加しておく．故に，トークンをわざわざルートノードから流さずに，直接ノードメモリM_Lに入れてやることから処理を開始できる．図１の場合には，ＲＥＴＥネットに比べ，照合回数が n1+n2+･･･+n8+3 回以上少なくて済む．ＲＥＴＥネットでは，どの動作パタンの連想ノードも全てルートノードになっているとみなすことができる．

ノードＮが動作パタンＰの連想ノードであるための必要十分条件は次の三つである．

[1] ルートノードからノードＮまでのパスに含まれる全てのノード（ただしＮは除く）でのテストにＰは必ず成功する．
[2] ノードＮでのテストにＰは成功する可能性があ

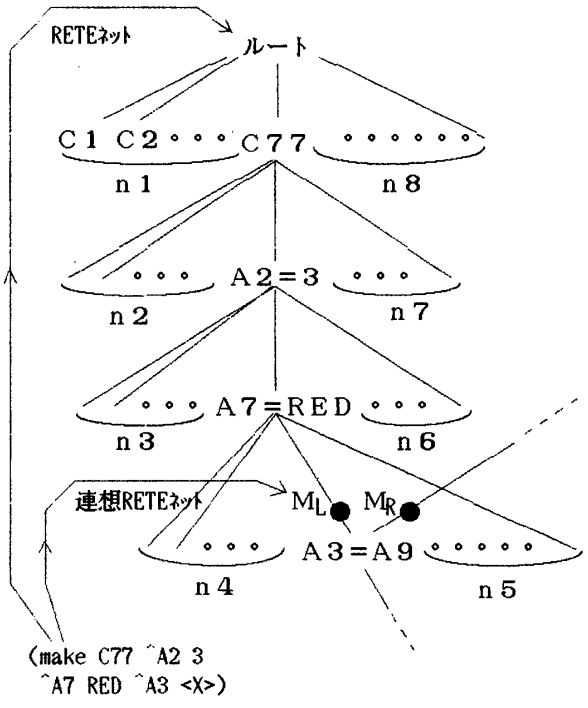

図１　連想ＲＥＴＥネットとＲＥＴＥネット

An Associable RETE Network Representation for Rule Bases, Shinji ARAYA
Fukuoka Institute of Technology

る（必ず成功あるいは失敗する場合を除く）．ただし，Nが端末ノードのときにはこの限りでない．

[3] ノードNから端末ノードに至るパスでのテストにPが成功する可能性があるような端末ノードが少なくとも一つ存在する．

連想ノードは，一つの動作パタンに対して一般に複数個存在する．連想ＲＥＴＥネットは，全ての動作パタンに対する全ての連想ノードを求め，それらを連想リンクで結合したものである．連想ＲＥＴＥネットの効率的生成法は別途報告する[5]．

３．従来法との比較

3.1 ＲＥＴＥネット方式との比較

1)提案方式はトークンをルートノードからではなく連想ノードから流すため，必ず成功あるいは失敗するような照合の大部分を回避できる．

2)提案方式の推論効率はノード共有度の影響を受けず，共有最大化を図った場合と同じ効率が得られる．

3)上記 2)より， 提案方式はＲＥＴＥネットの作成に時間効率の良い逐次構造化法[7]を利用しても推論効率が低下しない．

4)提案方式では，行き止まり動作パタンからは連想リンクが出ないので，それに起因する無駄な照合やメモリの使用を避けられる．

5)提案ネットを作成するには，ＲＥＴＥネットを作成してから連想ノードを求め連想リンクを付加するので，そのための処理時間並びに格納用メモリが余分に必要となる．

6)提案方式は動作パタンに含まれる変数の割合が増加するにつれて推論効率は低下する．ただし，ＲＥＴＥアルゴリズムよりも悪くなることはない．

3.2 排他ＲＥＴＥネットとの比較

排他ＲＥＴＥネットでは，分岐先のノードが互いに排他関係にあるならば，そのうちの一つのノードでのテストに成功すれば他のノードでのテストは必ず失敗するので，それらのテストを省略できる．ただし，成功ノードを見つけるまでの照合は不可欠なので，成功可能性の高いノードが早くテストされるように最適化される．これに対して提案方式では，必ず成功あるいは失敗するようなノードには始めからトークンが流れないのでより効率的である．排他方式では失敗照合回数だけが減少するのに対し，提案方式では成功照合回数も減少する．

3.3 ペトリネット方式との比較

この方式は，条件パタンや動作パタンを一つのノード（プレース）で表現し，条件パタンと動作パタンの定数部分が同じものをリンクで結ぶ．故に，動作パタンと条件パタンとの関連性を利用しているという点において提案方式と類似している．以下に主な相違点を示す．

1)推論時に使うペトリネットとは別に，トークンの初期配置を求めるための弁別ネットが必要である．

2)条件パタン間にまたがるテストは単にトランジションとしてモデル化されているので，ほとんど構造化されていない．

3)ＲＥＴＥネットにおける２入力ノードのノードメモリに相当するものがない．そのため条件パタン間にまたがる変数テストの結果は一部きり保存されないので同一照合が繰り返される．

4)ＲＥＴＥネットにおける１入力ノード及び２入力ノードの共有化が図られていない．

5)条件パタンや動作パタンに変数が含まれる場合でも照合結果があらかじめ確定する場合があることを考慮に入れていない．

3.4 ルール合成との比較

合成は，特定の問題解決に使用された複数のルールを編集して，同一効果を持つ単一のルールにまとめ上げる．主な相違点は以下の通りである．

1)合成は機能的に冗長なルールの数を増大させる．

2)合成ルールと原ルールが混在している状態では，ルールの追加／削除時の整合性維持が困難になる．

3)合成ルールの存在によって，生起すべき原ルールの発火系列が遮蔽され，結論が誤ることがある．

4)合成に当って中間結果を省略することができるが，場合によっては意味的に妥当でなくなることがある．

5)提案ネットが大局的編集によって生成されるのに対し，合成は特定の問題解決毎に行われる局所的編集である．故に，システム性能を全般的に向上させるのには多くの問題を与える必要があり，時間がかかる．

合成との比較の詳細は，文献６）を参照されたい．

４．あとがき

プロダクションシステム高速実行のための連想ＲＥＴＥネット方式を提案し，従来法との比較考察により提案方式の有用性を示した．本方式と逐次構造化[7]や逐次メンテナンスとの統合化は今後の課題である．

参考文献

1) Forgy,C.L.: Rete: A Fast Algorithm for Many Pattern/Many Object Pattern Match Problem, Artif. Intell., Vol.19, pp.17-37 (1982).

2) 荒屋ほか:プロダクションシステムのための高速パターン照合アルゴリズム,情報処理学会論文誌, Vol.28, No.7, pp.768-755 (1987).

3) 鶴田ほか：連想･選別型推論のアナロジーによるプロダクションシステムの高速実行方式,情報処理学会論文誌, Vol.26, No.4, pp.696-705 (1985).

4) Anderson, J.R.: Knowledge Compilation: The General Learning Mechanism, in Michalski, Carbonell and Mitchell (Eds): Machine Learning, Vol. 2, Morgan Kaufmann Pub.Inc. pp.289-310 (1986).

5) 荒屋：連想ＲＥＴＥネットの逐次編集，本大会 (1988)

6) 荒屋：知識編集に基づく学習，情報処理学会,知識工学と人工知能研究会，AI-59-12 (1988).

7) 荒屋ほか：知識ベースの逐次構造化 －ＲＥＴＥネットワークの逐次構築法－，電子情報通信学会論文誌，Vol.71, No.6 (1988).