

S T - N E T を用いた高速双方向推論アルゴリズム

5J-6

田野 俊一 \* 増位 庄一 \* 大森 勝美 \*\*

\* (株)日立製作所 システム開発研究所 \*\* おおみか工場

1. はじめに

エキスパートシステムの実用化の進展に伴い、推論機構の高速化が重要な課題となってきた。前向き推論に関しては、高速な処理方式<sup>1)2)</sup>がすでに提案されているが、双方向推論の高速化方式は、提案されていない。本論文では、S T - N E T (Semi bi-directional Transition NEtwork) アルゴリズムと呼ぶ高速双方向推論の実現アルゴリズムを提案する。

2. 双方向推論の実現に関する基本的考え方

まず、単純な知識を用いた双方向推論を形式的に捉えてみる。事象を、A, B, C, ...、推論に用いる知識を、if A THEN B のように表わす。この知識を用いれば、事象を、図1に示すネットワークにより関連付けることができる。事象間の関連図における矢印は、導出方向を表わしている。例えば、事象Aが真であれば、事象Bを導くことができることを表わしており、この方向に新たな事象を導出することにより、前向き推論を行なうことができる。また、矢印と逆の方向に処理することにより、後向き推論を行なうことができる。つまり、図1のように知識を表現すれば、単に情報の流れる方向を変えるだけで双方向推論が実現できるはずである。

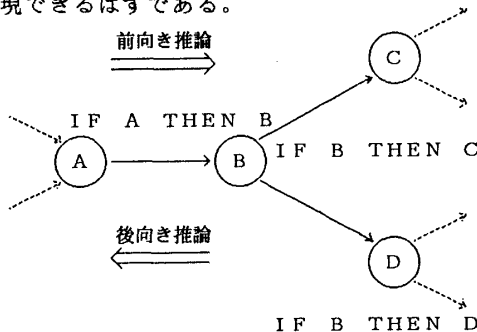


図1 事象間の関連ネットワークの例

しかし、実用的記述力を持つルールを、図1のような完全な事象間の関連ネットワークに変形することはできない。この問題点は次のように形式化できる。ルールを

if (Left Hand Side パターン)  
then (Right Hand Side パターン)

のように表わすと、それぞれのパターンは、非伝播パターンと伝播パターンに分解できる。即ち、  
LHSパターン = 非伝播LHSパターン + 伝播LHSパターン  
RHSパターン = 非伝播RHSパターン + 伝播RHSパターン  
のように分解できる。非伝播パターンとは、それぞれのsideで閉じているパターンであり、逆に、伝播パターンとは、それぞれのsideで閉じておらず、他のsideに影響されるパターンである。

例として、パターンを述語で表わした「if A(1,2,?X) then B(2,?X,4)」のルールを考える。?Xは変数を表わしており、このルールは、もし、述語名がAであり、第1引数が1、第2引数が2、第3引数が?Xであるならば、述語名がBであり、第1引数が2、第2引数が?X、第3引数が4である述語を導けることを意味する。

これは、以下のように分類できる。  
非伝播LHSパターン 述語名=A, 第1引数=1, 第2引数=2  
伝播LHSパターン 第3引数=?X  
非伝播RHSパターン 述語名=B, 第1引数=2, 第3引数=4  
伝播RHSパターン 第2引数=?X

伝播パターンは、ルールが、前向き、後向きに適用される時点になって初めて実体化するパターンである。例えば、事実 A(1,2,5) をLHSパターンにマッチさせることにより、伝播RHSパターンは、「第2引数=5」という実体になる。また、ゴール B(2,3,4) をRHSパターンにマッチさせることにより、伝播LHSパターンは、「第3引数=3」という実体になる。

このため、RHSパターンと、LHSパターンが等しいことを確認しようとしても、伝播パターンが実体化されていないため完全なパターンマッチは不可能であり、図1のような完全な関連図は、生成できない。

しかし、非伝播パターンのみに関して解析することにより、図2に示す関連図を生成することができる。図2において、斜線部分は、パターンとパターンの共通部分を示している。例えば、B をRHSパターンB(2,?X,4)、B' をLHSパターンB(2,3,?Y) と考えると、斜線で示す共通部分Xには、「述語名=B, 第1引数=2」が記憶され、残りのパターン「?Y=4」、「?X=3」は、それぞれ、非共通部分Y, Zに記憶される。この場合、?Xが、3となると、RHSパターンは、B(2,3,4)になり、LHSパターンを満たすことになる。この判定でわかるように、斜線で示す共通部分の条件チェックは、不要であり、単に、非共通部分Zの「?X=3」のチェックのみで、LHSパターンが満たされることが判定できる。即ち、このネットワークを用いれ

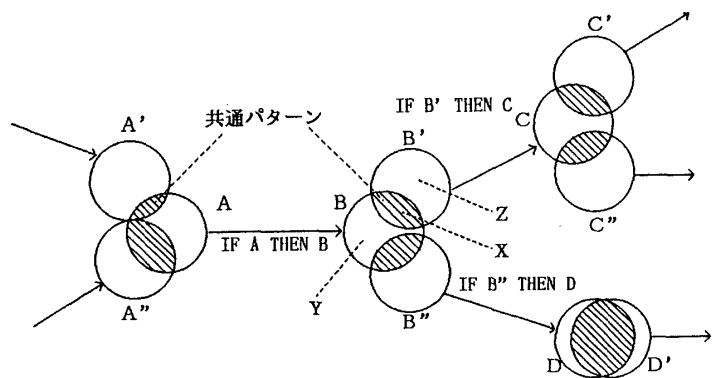


図2 非伝播パターン解析によるネットワーク生成

A fast algorithm for a bi-directional inference

Shun'ichi TANO \*, Shoichi MASUI \*, Katumi OHMORI \*\*

\* Systems Development Laboratory, HITACHI, Ltd.

\*\* Omika Works, HITACHI, Ltd.

ば、共通部分の処理は不要であり、かつ、判定処理は、非共通部分のパターンのみで限定できる。

以上のように、非伝播パターンに着目し解析することにより、図2のネットワークが生成でき、さらに、無駄なマッチ処理の大幅な削減が見込める。

次に、図2のネットワークを、処理、つまり、推論の観点で見つめる。まず、前向き推論を考える。前向き推論においては、矢印に沿って事象が流れる。ネットワーク内を流れる事象をトークンとして捉えれば、このネットワークは、ペトリネットとしてみなせる。同様に考えると、後向き推論では、ネットワーク内を矢印の逆方向に流れるゴールが、トークンである。つまり、図2のネットワークは、事象トークン、ゴールトークンの2種のトークンが流れるペトリネットとしてみる事ができる。ゴールトークンと事象トークンが衝突した場合、そのゴールは達成されたことを表わしている。推論の状態は、このトークンのネットワーク内での配置で表わされる。事象トークンの状態が、前向き推論の状態を、ゴールトークンの状態が、後向き推論の状態を表わしている。

また、このトークンの流れを制御することにより、多様な推論方式を実現できる。例えば、事象トークンの流れを制御することにより様々なコンフリクトリゾリューション戦略での前向き推論を、また、ゴールトークンの流れを制御することによりdepth-first, bread-first等の後向き推論を、事象トークン、ゴールトークンの流れを連携させ制御することにより高度な協調型の双方向推論を実現できる。さらに、事象、ゴールの両トークンをネットワーク内に記憶することにより、過去の処理履歴を利用した効率のよい処理を実現できる。

### 3. ST-NETアルゴリズムの概要

ルールを非伝播パターンに着目して解析することにより、状態遷移の方向を変えるだけで双方向推論の実現が可能なルール間関連ネットワークが生成できる。

まず、状態遷移の方向を変えるだけで双方向推論が可能な、ルールの表現方法について考えてみる。

前向き推論、後向き推論におけるルールの評価の違いについて考察する。前向き推論では、ルールのLHSを条件の成立判定に用い、RHSを事象の生成に用いる。一方、後向き推論では、RHSを適用できるルールの探索に用い、LHSをサブゴールの生成に用いる。即ち、LHSは、条件の成立判定、サブゴールの生成に用いられ、RHSは、適用できるルールの探索、事象の生成に用いられる。つまりLHS, RHSは、同様な2種の処理-条件判定、生成-の基本処理により、評価される。

基本処理の1つである条件判定を高速化する手段としては、LHS, RHSそれぞれを、弁別ネットに変換し、条件判定を行なう方式が考えられる。つまり、LHSの弁別ネットの頂点から、事象を処理、つまり、弁別ネット内のノードに記憶されている条件を満たせば次のノードに進み、ターミナルノードに到達すれば、実行可能ルールであると判断し、一方、RHSネットの頂点ノードからゴールを流す(上記LHSネットと同じ処理)ことにより、ゴールを導くルールを知ることができる。これにより、LHSでの条件判定、RHSでの適用できるルールの探索が実現できたことになる。逆に、RHSのターミナルノードから頂点ノードに向かって処理すればWMEが生成でき、LHSのターミナルノードから、頂点ノードに向かって処理すればサブゴールが生成できることになる。

この2つの弁別ネットのターミナルノードは、いずれもルールの終端を表わしており、これを結合点として、2つのネットを結合する。結合したネットワークは、LHSの頂点ノードを表わすf-root(*forward-root*)ノード、RHSの頂点ノードを表わすb-root(*backward-root*)ノードの2つの頂点ノードを持ち、中間部は、LHSネット、ターミナルノード、RHSネットの並びとなる。

つぎに、以下のようなshortcutアークを導入し、ルール間関連図とする。上記のネットワークを用いると、前向き推論は、RHSネットでは事象を生成し、得られた事象をf\_rootから処理し、LHSネットでは条件判定し、実行可能ルールを見つけ、さらに、RHSネットで、事象を生成し…の過程で進む。後向き推論も同様である。この処理過程の生成部分と条件判定部分において共通に現れる処理が、図2で示した共通パターンである。

そこで、同じ処理を飛び越す、即ち、共通部分を飛び越すshortcutアークを設ける。これにより、ネットワークは、図3に示す構造となる。この構造を持つネットワークをST-NETと呼ぶ。前向き推論でのshortcutアークは、RHSネットでは生成されたWMEを流すべきLHSネット、即ち、そのルール実行において条件が満たされる可能性のあるルールへのポインターである。後向き推論でのshortcutアークは、LHSネットでは生成されたゴールを流すべきRHSネット、即ち、ゴールを導く可能性のあるルールへのポインターである。

このshortcutアークの外側が図2で言う共通パターンを表わしており、このshortcutアークは、双方向推論におけるルール間の関連を示している。

図3に示した平面の両端は、図2におけるパターンの重なり部分を表わしている。中央の円筒部分の、右回りの方向の事象トークンの流れが後向き推論、左回りの方向のゴールトークンの流れが前向き推論となる。

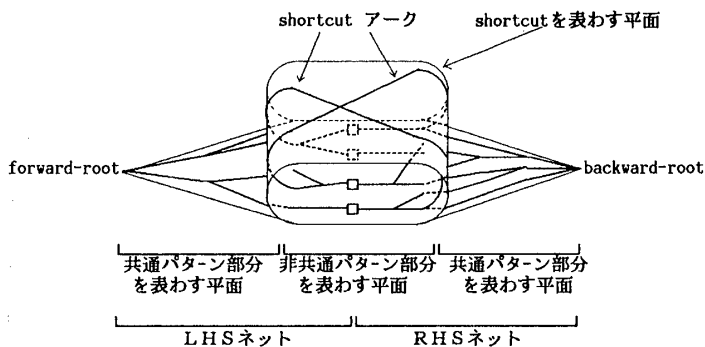


図3 ST-NETの構造

### 5. まとめ

ST-NETを用いた高速双方向推論アルゴリズムを示した。本方式は、単に、高速であるだけではなく、同一ルールの双方向での適用、および、前向き推論と後向き推論が協調した推論を可能とする方式である。

### 参考文献

- 1) Forgy, C.L. : Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, ARTIF. INTELL., Vol.19, No.1, pp.17-37(1982).
- 2) 田野 他 : 知識処理ソフトウェア EUREKA における高速処理方式、情報処理学会論文誌, Vol.28, No.12, pp.1255-1268(1987)