ST-NET アルゴリズム：双方向推論の高速処理方式†

田野俊一†† 増位庄一††

知識工学の実システムへの適用が進むにつれて，より高速な推論機構が望まれている．本論文では，高速な双方向推論を可能とするアルゴリズムを提案する．まず，双方向推論の実現に関し，① ルールの非伝播パターンに着目することにより，ルール間の関連図の生成が可能であり，これを用いて高速に推論ができること，② その関連図は，ワーキングメモリエレメント，ゴールの2種類のトークンを持つペトリネットとしてみなすことができ，このトークンの動きを制御することにより，様々なタイプの双方向推論を実現できること，を示す．次に，この解析に基づき得られた，ST-NET を用いた双方向推論の高速化アルゴリズムを提案する．ST-NET の特徴は，① ルール条件部，結論部を1つのネットワークで表現すること，② ネットワークは有向アークで構成され，部分的に双方向ネットワークとなること，③ 条件部，結論部に共通したパターンを飛び越す shortcut アークを有することである．本アルゴリズムは，ネットワーク内に，ワーキングメモリエレメント，ゴールの2種類の情報を履歴データとして保持すること，および，shortcut アークに沿って処理すること，により処理を効率化し，ネットワーク内の2種類のデータの流れを制御することにより，双方向推論を可能にした．最後に，簡明な知識表現を持つ推論システムに対する ST-NET の適用例を示す．

1. はじめに

知識工学は，人工知能で得られた研究成果を，実際の問題解決に応用することを目的とする新しい工学である[1]．知識工学を用いることにより，人間の持つ知識を計算機に移植し，専門家なみの能力をもつ計算機システムを構築することが可能となり，最近注目を集めている．

一般に，人間が行う問題解決においては，そこで用いられる知識の全体構成，相互関連を事前に詳細に把握することは困難である．このような知識は，断片的な知識の形でしか得られないことが多いため，ルールを用いた知識表現法が知識工学ツールの代表的手法と考えられている[2]．

ルールを用いた演繹推論方式は，前向き推論と後向き推論に大別できるが[3]，人間は，いずれか1つだけではなく，それらを相互に関連させ，問題解決を行うことが多い．そのため，上記2つの推論方式を持つことは，知識工学ツールの必須の条件である[4~6]．さらに，実用エキスパートシステムを得るには，これらの推論処理を高速に行うことが必須である．

我々は，前向き推論に関して，有効な高速処理方式をすでに提案した[15]．しかし，人間の知識の取り扱いの基本をなすと考えられる双方向推論の高速実現アルゴリズムについては，まだ提案されていない．

本論文では，双方向推論を実現でき，かつ，高速なアルゴリズムである ST-NET アルゴリズムを提案する．

以下，次章では，高速推論に関する従来の研究，第3章では，双方向推論の実現に関する基本的考え方，第4章では，ST-NET を用いた双方向推論アルゴリズム，第5章で，本方式の適用例を示す．

2. 高速推論に関する従来の研究

前向き推論の高速化に関しては，ルールが関係するワーキングメモリエレメント（WME）をあらかじめ関連図として解析しておき，これを用いて高速化する手法[7~10]，条件成立判定の履歴情報を用いる RETE アルゴリズム[11]，さらに高速化するためのアルゴリズム[12~16] 等多くの手法が提案されている．

しかし，双方向推論の高速処理アルゴリズムは，まだ提案されていない．このため，双方向推論を行う現在の知識工学ツールは，前向き推論に RETE アルゴリズムを用い，後向き推論は，後向き用のルールを前向き推論で用いられるように変換することにより，実現するものが多い[5),6]．

例えば，A, B,… を事象名として，「G を導くためには，A, B, を検証すればよい」ことを表す「$G \leftarrow A$, B」の後向き推論用のルールは，「もし，G の導出が要求されており，A, B が真ならば，G を導出する」のような前向き推論に適したルールに変換される．このルール条件節に現れる「G の導出が要求されており」の記述を，G' のように表すと，上述のルールは，

† ST-NET: A Fast Algorithm for Bidirectional Inference by SHUN'ICHI TANO and SHOICHI MASUI (Systems Development Laboratory, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所

「IF G', A, B THEN G」となる．すなわち，G' がワーキングメモリに書かれた場合に，このルールが起動されるようになる．後向き推論の特徴は，ゴールの成立判定に寄与するルールだけを用いることであるが，これを前向きで実現するため，このルールが今動くべきであることを，G' といういわばフラグのような状態量で表し，G' をワーキングメモリに書き出すことで，ルールの動きを制御しようとするものである．この方法では，推論機構は，後向き推論によって導かれる事象を完全に把握し，その事象の検証が必要であるか否かを常に監視する必要がある．つまり，ゴール G が与えられると，G' をワーキングメモリに書き出し，上記ルールを適用しようとするが，例えば，事象 A の真偽が未定である場合，A が，後向き推論で導くことのできる事象であるかを判定し，もし，後向きで検証すべき事象であれば，A' をワーキングメモリに書き出す必要がある．この「後向きで検証すべき事象であるかどうかの判定処理」は，ルール条件部のすべてのチェック過程で行う必要がある．

すなわち，前向き推論を用いて，擬似的に後向き推論を実現する場合には，上述の処理がオーバヘッドとして伴うことになる．

さらに，この方式では，ルールを前向き用，後向き用のように明確に定義する必要があり，同一ルールを状況によって前向き，後向き推論で用いる本質的な双方向推論の実現は困難である．

3. 双方向推論の実現に関する基本的考え方

本章では，双方向推論に関し考察を行い，その高速な実現のための3つの着眼点を示す．

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，$A, B, C, \cdots$，推論に用いる知識を，IF A THEN B のように表す．事象は，この知識を用いて図 1 に示すネットワークの形で相互に関連付けられる．事象間の関連図における矢印は，導出方向を表している．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している．矢印の向きに従い，新たな事象を導出することは，前向き推論に当る．また，矢印の逆の方向に処理することは，後向き推論に相当する．

つまり，図1の表現で知識を表せば，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ことになる．これが，第一の着眼点である．

しかし，一般に，ルールを，図1に示すような

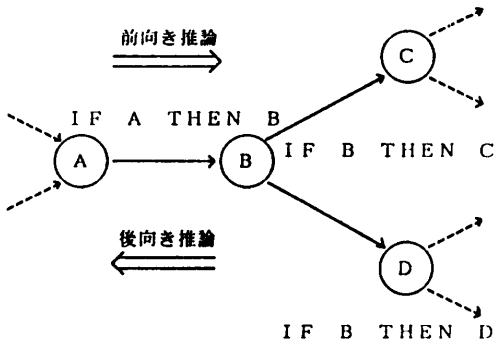


図 1 事象間の関連ネットワークの例
Fig. 1 An example of an event network.

strict な事象間の関連ネットワークに展開することは困難である．

この問題点は次のように形式化できる．ルールの条件部 (Left Hand Side) は，ルールが適用可能となるワーキングメモリの状態記述 (LHS パターン，あるいは，単に LHS と呼ぶ) であり，結論部 (Right Hand Side) は，ルールの適用後のワーキングメモリの状態記述 (RHS パターン，あるいは，単に RHS と呼ぶ) である．

それぞれのパターンは，2つの性質の異なるパターンに分解できる．

例として，「IF A (1, 2, ?X) THEN B (2, ?X, 4)」のルールを考える．?X は変数を表しており，このルールは，もし，第1引数が1，第2引数が 2，第3引数が ?X であり述語名が A である述語が存在するならば，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．

これらのパターンは，以下のように分解できる．

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，そのルールが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と，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と，伝播 LHS パターンは，「第3引数＝3」という実体になる．一方，非伝播パターンは，他 side のパターンとは独立である，つまり，そ

の side で閉じているパターンである．

以上のように，パターンは，それぞれの side で閉じている非伝播パターンと，他の side に影響される伝播パターンに分解できる．すなわち，

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．

このため，**図 2** に示すように，ルール1の RHS パターンと，ルール2の LHS パターンが等しいことを事前解析により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となる．

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 3** に示すような部分的関連図を生成することができる．図3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「?X＝3」「?Y＝4」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．この例からわかるように，このような部分的関連図展開を行うことにより，斜線で示す共通部分の条件チェックは，不要となり，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．すなわち，ルール適用時の判定処理を，非共通部分のパターンのみに限定できる．

以上のように，非伝播パターンに着目しルールの関連を，図3のネットワークに展開できれば，無駄なマッチ処理を削減でき，かつ，これを用いて，データの流れの方向を変えることで，前向き推論，後向き推論が容易に実現できる．これが第2の着眼点である．

次に，図3のネットワークを，処理，つまり，推論の観点で見てみる．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．後向き推論では，ネットワーク内に流れるゴールを，トークンと考えることができる．つまり，図3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ことができる．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になる．

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，双方向推論を実現できる．

したがって，事象，ゴールの両トークンをネットワーク内に記憶し，これを用いることにより，過去の処理履歴を利用した，効率のよい双方向推論を実現できる．これが，第3の着眼点である．

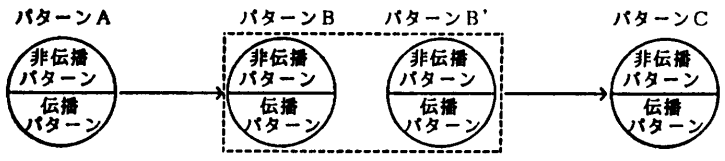


図 2 関連ネットワーク生成の問題点
Fig. 2 A problem of network generation.

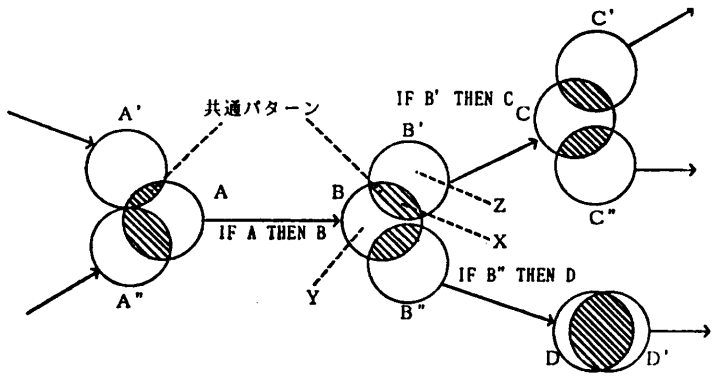


図 3 非伝播パターン解析によるネットワーク生成
Fig. 3 Network generation by non-transformational pattern analysis.

4. ST-NET アルゴリズム

本章では，前章で明らかにした3つの着眼点から生み出された ST-NET (Semi bi-directional Transition NETwork) アルゴリズムを提案する．

4.1 ST-NET の構造

第1，第2の着眼点に沿い，状態遷移の方向を変えるだけで双方向推論が可能な，ルールのネットワーク化の方法について考えてみる．

まず，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，RHS を WME 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，LHS をサブゴールの生成に用いる．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，ルール条件部を弁別ネットに変換し，これを用いて条件成立判定を行う方式があり効率がよい[15]．

これを用いると，LHS, RHS を，それぞれ弁別ネットに変換し（LHS ネット，RHS ネットと呼ぶ），条件判定を行う方式が考えられる．つまり，LHS ネットの頂点から，WME を流す，つまり，前向き推論においては，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する．一方，後向きの場合は，RHS ネットの頂点ノードからゴールを流すことにより，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る．

次に，WME，ゴールの生成について考察する．前向き推論においては，ルールの実行により得られる WME を生成するための情報は，RHS ネット内にある．RHS ネットの頂点から，ゴールを流す処理とは逆に，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，このルールを用いた場合に生成される WME を知ることができる．後向きの場合のサブゴールの生成も，LHS ネットのターミナルノードから，頂点ノードに向かって処理することにより，実現できる．つまり，LHS, RHS の弁別ネット内のアークを双方向にすれば，同一ネットワークを用いて双方向推論が行える．

この2つの弁別ネットのターミナルノードは，いずれも1つのルールの終端を表しており，このターミナルノードを結合点として，2つのネットを結合する．結合したネットワークは，LHS の頂点ノードを表す forward-root (f-root) ノード，RHS の頂点ノードを表す backward-root (b-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．このネットワークの構造を図 4 に示す．さて，このネットワークのアークは，事象，ゴールを流す方向に関する属性を持っている．f-root から，b-root への方向は，前向き推論，逆の方向は，後向き推論を表している．前向きにしか使えないルール，後向きにしか使えないルールは，単方向のアークにより表す．双方向に用いられるルールは，双方向のアークで表す．さらに，後向きに用いられるルールであっても，生成するサブゴールが，他のルールにより導けない場合は，生成しても無駄であるので，サブゴール生成方向のアークは存在させない．このようにすると，部分的な双方向のネットワークが形成できる．

以上により，双方向推論が可能なネットワークが得られるが，このネットワークは，ルール間の関連を直接表し得ておらず，第2の着眼点を実現できていない．RHS ネットで生成された WME を f-root から，あるいは，LHS ネットで生成されたゴールを b-root から流すことによって，はじめて，ルールの関連がわかるだけである．つまり，このネットワークは，図3における斜線部分がなく，ルールの RHS を表す○印にすべてのルールの LHS を表す○印が接している（つまり，共通パターンがない）ネットワークを表しているに過ぎない．

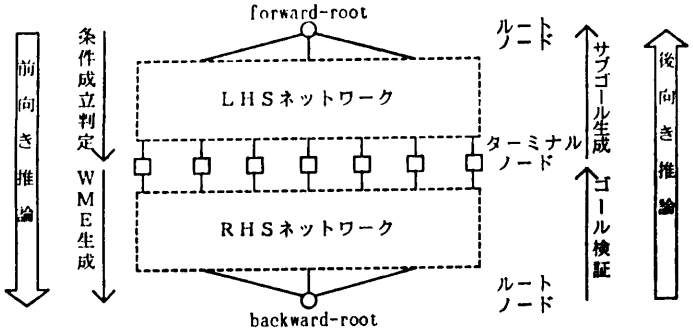


図 4 LHS ネットと RHSネットの結合
Fig. 4 Combination of LHSnet and RHSnet.

上記のネットワークでの生成—条件判定のサイクルにおいては，生成部分と条件判定部分で共通に現れる処理が，図3で示した共通パターンになる．そこで，同じ処理を飛び越す，すなわち，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5に示す構造となる．前向き推論での shortcut アークは，RHS ネットで生成された WME を流すべ

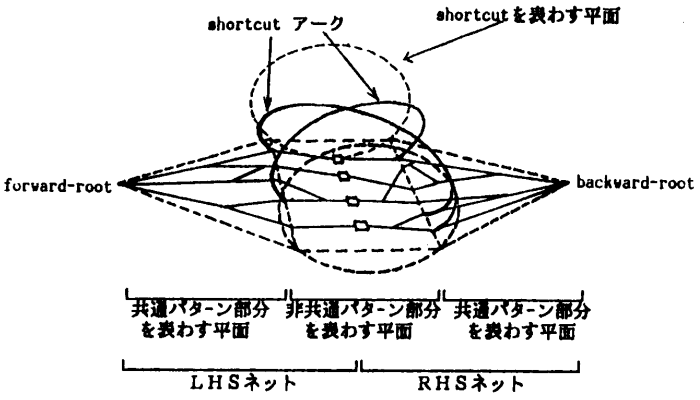


図 5 ST-NET の構造
Fig. 5 Structure of the ST-NET.

き LHS ネットを指し示すようにする．すなわち，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を設ける．後向き推論での shortcut アークは，LHS ネットで生成されたゴールを流すべき RHS ネットを指し示すようにし，ゴールを導く可能性のあるルールへのポインタを設ける．このように，shortcut アークにより，双方向推論におけるルール間の関連を示すことができる．

この構造を持つネットワークが ST-NET である．

4.2 ST-NET の処理

ST-NET の前向き推論処理は以下のようになる．まず，ST-NET のf-root ノードから，WME を流し，各ノードに記されている条件を満たすか否かをチェックする．満たす場合は，下位のノードに WME を流し，条件を満たさない場合は，処理を中止する．WME がターミナルノードに到達した場合，このルールが実行可能であると判定する．さらに，ターミナルノードに続く RHS ネットを b-root ノードに向かって，ノードに記されている条件を収集することにより，導くことのできる WME を生成する．

逆に，後向き推論処理は，導きたい WME（ゴール）を，前述の LHS ネットでの WME の処理と同様に，RHS ネットを b-root ノードから，f-root ノードに向かって処理する．ゴールが，ターミナルノードに到達した場合は，このルールを用いてこのゴールが導けることを判定し，さらに，ターミナルノードに結合している LHS ネットを f-root ノードに向かって処理し，WME を生成した場合と同様の処理により，サブゴールを生成する．

この2つの処理は，それぞれ，前向き推論，後向き推論の1つのルールの適用に対応する．2つの推論の生成処理（前向きでは，RHS ネット，後向きでは，LHS ネットの処理）において，shortcut アークが現れた場合は，このアークに従って処理を進める．推論の各時点で生成される完全な WME，ゴールが必要な場合は，生成を f-root あるいは，b-root ノードに到達するまで行う必要があるが，中間結果としての WME，ゴールが不要な場合は，shortcut アークのみに流せばよい．

このネットワークを流れる WME，ゴールをネットワーク内のアークに保持するようにし，変化した WME のみネットワークに流すことにすれば実行可能ルールを即座に知ることができる．これは，RETE の考え方と同じである．さらに，過去に導出を失敗したゴールをネットワーク内に記憶することにすれば，同じゴールがすでにネットワーク内に記憶されている場合，このゴールを直ちに達成不能であると判定することができ，無駄なゴールに関する後向き推論を避けることもできる．

5. ST-NET アルゴリズムの適用例

本章では，図 6(a)に示すような簡明な知識表現を持つ推論システムへの適用例を示す．

5.1 適用する推論システム

WME は，クラス名，および，スロット名，スロット値の複数の組より構成される．例を図6 (b) に示す．

1つの WME に関する条件を，条件節と呼ぶ．条件部は，複数の条件節で構成される．結論部は，1つの WME の記述よりなる．図6 (c)に示すルールは，*A* 系統のポンプで，回転数が高く，同じ系統のタービンが正常であるという条件を表している．"?*X*" は，変数を表し，同一ルール内では，同一値となる．図6 (c) のルールに図6 (b) の WME がマッチした場合は，(診断結果　$故障系統　*A* 系統　$スタート機器　ポンプ3号) の WME が生成される．

5.2 ネットワーク表現と記憶情報

スロットと定値との比較をコンスタントノード，変数との比較をバリアブルノードで表現する．このネットワークは，図 7 に示すノード階層となる．

図7に示す階層において，ノードの上にある枝を入力枝，ノードの下にある枝を出力枝とする．ターミナルノードは，*N* 入力枝，1出力枝を持つ．

〈ルール〉＝〈ルール名〉 IF 〈LHS パターン〉 THEN 〈RHS パターン〉
〈LHS パターン〉＝〈WME パターン〉。。。
〈RHS パターン〉＝〈WME パターン〉
〈WME パターン〉＝(〈クラス名〉 (〈スロット名〉 (〈値〉|〈変数〉))。。。)
〈クラス名〉＝〈文字列〉
〈スロット名〉＝$〈文字列〉
〈値〉＝〈文字列〉
〈変数〉＝?〈文字列〉
〈WME〉＝(〈クラス名〉 (〈スロット名〉〈値〉)。。。)
〈ゴール〉＝〈WME〉

(a) シンタックス

(ポンプ $系統 A系統 $回転数 高い $代替機器 ポンプ3号)
(タービン $系統 A系統 $状態 正常 $代替機器 タービン4号)

(b) WME の例

ルール1 IF (ポンプ $系統 A系統 $回転数 高い $代替機器 ?X)
(タービン $系統 A系統 $状態 正常)
THEN (診断結果 $故障箇所 A系統 $スタート機器 ?X)

(c) ルールの例

(診断結果 $故障箇所 A系統 $スタート機器 ポンプ3号)

(d) ゴールの例

図 6 推論システムの知識表現
Fig. 6 Knowledge representation for an inference system.

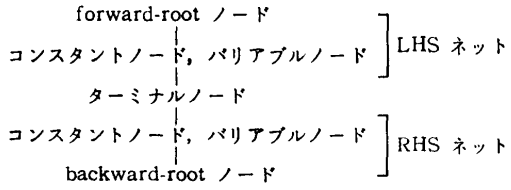


図 7 ネットワークを構成するノードの階層
Fig. 7 Hierarchy of network nodes.

ターミナルノードの入力枝には，それぞれの条件節を満たす WME を記憶し，出力枝には，それぞれの枝から流れてきた WME の組を記憶する．つまり，この WME の組は，そのルール条件部を満たすインスタンシエーションである．未達成のゴールは，ターミナルノードの出力枝に記憶する．

ルールの例を図 8(a)に示す．すべてのルールを双方向に使う場合の対応する ST-NET を図8(b)に示す．

5.3 ネットワーク処理

まず，WME をネットワークに流す処理を説明する．コンスタントノードでは，そのノードに記憶されている条件を満たすかをチェックし，バリアブルノードでは，そのスロットがあることを確認し，変数に値を代入する．ターミナルノードに到達した場合は，入力枝に WME を記憶し，他のすべての入力枝に WME が記憶されている場合は，それらの組を作成し，出力枝に記憶する．

例えば，初期状態（つまり，ネットワーク内に記憶されている情報がない状態）の図8(b)のネットワークに

WME1 (*A* $*A*1 *a*1 $*A*2 *a*2 $*A*3 *a*3)

を流すと，ルール1，ルール2のターミナルノードの入力枝に到達し，その枝に WME1 が記憶される．ルール1のターミナルノードは，1つの入力枝しかないので (WME1) を出力枝に記憶する．ルール2のターミナルノードは，2つの入力枝を持つが，他の入力枝には，何も記憶されていないので出力枝に記憶されない．この状態で，

WME2 (*B* $*B*1 *b*1 $*B*2 *b*2 $*B*3 *b*3)

を流すと，ルール2，3のターミナルノードに到達し，その入力枝に記憶される．ルール2のターミナルノードの他の入力枝には，WME1 が記憶されているので，(WME1, WME2) の組を作りその出力枝に記憶する．

以上の過程では，ルール 1, 2 のターミナルノードの出力枝に，それぞれ，(WME1)，(WME1, WME 2) の組が記憶され，この組により条件部が満足されたことがわかる．

つぎに，ターミナルノードの出力枝に記憶されている WME の組を，RHS ネットに流す．RHS ネットでは，コンスタントノードに記憶されている条件および，バリアブルノードに記憶されている条件に従い WME を順次生成していく．shortcut アークが現れた場合は，その時点で生成された WME を，shortcut アークに沿って流す．例えば，ルール1の (WME 1) を RHS ネットに流した場合，?*X* は，*a*3 となり，shortcut アークに到達した時点で生成されている ($*c*3 *a*3) を shortcut アークに沿って流す．

つぎに，後向き推論について説明する．与えられたゴールを b-root ノードから流す．コンスタントノードでは，そこに記憶されている条件の成立判定を行い，バリアブルノードでは，変数の制約を作成する．例えば，ゴールを，

goal1 (*D* $*D*1 *d*1 $*D*2 *d*2 $*D*3 *a*3)

とすると，「$*D*3＝?*Y*」のバリアブルノードでは，「?*Y*＝*a*3」という制約を生成する．

ゴールがターミナルノードに到達した場合，その

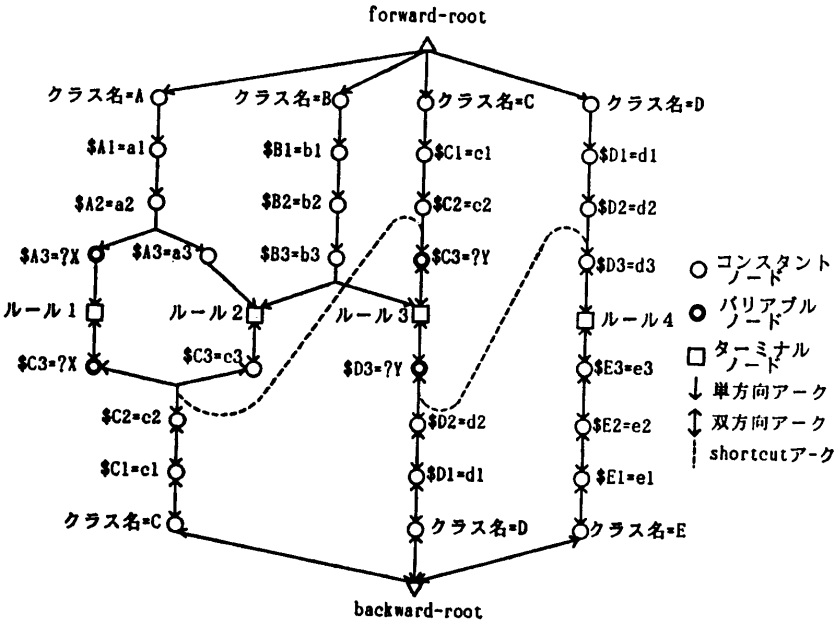
```
ルール1   IF   (A  $A1 a1  $A2 a2  $A3 ?X)
          THEN (C  $C1 c1  $C2 c2  $C3 ?X)

ルール2   IF   (A  $A1 a1  $A2 a2  $A3 a3)
               (B  $B1 b1  $B2 b2  $B3 b3)
          THEN (C  $C1 c1  $C2 c2  $C3 c3)

ルール3   IF   (B  $B1 b1  $B2 b2  $B3 b3)
               (C  $C1 c1  $C2 c2  $C3 ?Y)
          THEN (D  $D1 d1  $D2 d2  $D3 ?Y)

ルール4   IF   (D  $D1 d1  $D2 d2  $D3 d3)
          THEN (E  $E1 e1  $E2 e2  $E3 e3)
```

(a)ルールの例



(b)対応するST-NET表現

図 8 ST-NET 表現の例
Fig. 8 An example of ST-NET.

ゴールを制約条件とともにターミナルノードの出力枝に記憶する．例えば，上記の例では，(goal1, ?Y=$a3$) が記憶される．

つぎに，これらのゴールを LHS ネットへと流す．上記の例では，goal1 をさらに上へと流し，新たなゴールを生成する．この生成は，WME の生成と同じ処理である．コンスタントノード，バリアブルノードで，新たなゴールを順次生成していく．shortcut アークに到達した場合，その時点で生成されているゴールを，それぞれの枝に流す．上記の例では，goal2 (\$$C3$ $a3$) が shortcut アークに沿って流れ，RHS ネットの \$$C2$=$c2$ のノードの上位ノードからの処理になる．\$$C3$=$c3$ ではないので，ルール2のターミナルノードには，到達しないが，ルール1のターミナルノードには，「?X=$a3$」という制約とともに到達する．ゴールがターミナルノードに到達した場合，そのノードの出力枝に記憶されている WME の組があり，かつ，制約を満たす場合，ゴールは，この WME の

組でルールを実行することにより導けることを表している．例えば，goal2 がルール1のターミナルノードに到達した場合，この出力枝には，(WME1) が記憶されており，制約「?X=a3」を満たすので，goal2 は達成可能であると判定できる．

6. 評　価

まず，本アルゴリズムを機能の面から評価する．本方式では，① ルール条件部，結論部を，1つの部分的に双方向なネットワークにより表現し，② ネットワーク内を流れる情報として，WME およびゴールの2種類を設けた．このネットワーク構造とネットワーク内情報の構成により，単に，ネットワーク内を流れる情報の動きを制御することにより双方向推論の実現が可能となった．

さらに，前向き推論において必要な WME が未知である場合，自動的にそれを後向き推論により確認する推論のような，前向き推論と後向き推論が混在した，柔軟な双方向推論も，ネットワーク内を流れる情報の動きを制御することで実現できる．つまり，前章で示した，ネットワークにおいて，「WME がターミナルノードに到達したが，他の枝に WME がなく，かつ，その枝が双方向である場合，その枝から，ネットワークを，f_root 方向に辿り，そこに記述されている条件を収集し新たなゴールを生成し，shortcut アークの指す RHS ネットへ流すこと」で実現できるのである．

つぎに，処理性能の面から評価する．性能を向上させる要因は，① アークの持つ方向性，② shortcut アークの導入，③ 履歴情報の利用の3点である．①により，ネットワークに流す情報の方向性を規定することができる．つまり，後向きに適用しても無駄であるような場合，この方向での推論を禁止することが明示的にでき，これにより，無駄なゴールの生成を少なくすることができる．②により，WME あるいはゴールの生成，条件チェックでの無駄な処理を省くことができる．例えば，図8においてルール3を前向きに適用した場合，「$D2=d2, $D1=d1, クラス名=$D$」の生成，「$D2=d2, $D1=d1, クラス名=$D$」「クラス名=$A$」「クラス名=$B$」「クラス名=$C$」を削減できる．③による前向き推論における効率化はすでに報告されているので，ここでは，後向き推論における利点を述べる．ネットワーク内に保持されているゴール情報は，未達成のゴールであり，ネットワーク処理において，生成されたゴールがすでにネットワーク内に保持されている場合は，直ちに，達成不能であると判定でき，無駄な推論を回避できる．

しかし，これらは，解析可能な非伝播パターンの割合に依存している．つまり，解析可能な非伝播パターンが全くない，あるいは，少ない場合，推論処理の高速化の程度は低い．

そこで，実際のルール記述における非伝播パターン出現の割合，および ST-NET を用いることによりどの程度効率化が図れるかを，推論速度評価でよく用いられている，「サルとバナナ」[17]を用いて解析する．

図9(a) にルールの例を示す．"〈", "〉" で囲まれた文字列は，変数を表す．図中で，"^目標未達成" は，非伝播パターンを，"^物体名 〈obj〉" は，伝播パターンを表しており，これを1パターンとして数える．例えば，このルールの条件部は，7つの非伝播パターンと，3つの伝播パターンで構成されている．

図9(a)の結論部の意味は，条件部の1番目の条件節にマッチした WME の "^目標" を，"達成" に変更することを explicit に表しているが，隠された情報として，「"^動作" は，"上に乗る"」のままであることも implicit に表現しており，同図(b)に示すように，ルールを変形し，共通パターンの解析を行っ

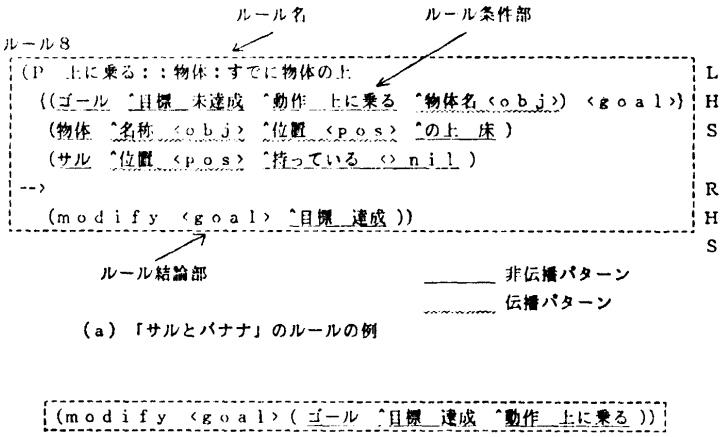


図9 「サルとバナナ」のルールの例
Fig. 9 An example of a rule appeared in "monkey and banana".

表 1 「サルとバナナ」における伝播・非伝播パターンの割合
Table 1 Proportion of propagative/non-propagative pattern.

	条件部 (割合 %)	結論部 (割合 %)	ルール全体 (割合 %)
非伝播パターン	178 (68%)	148 (92%)	321 (77%)
伝播パターン	85 (32%)	12 (8%)	97 (23%)
合 計	263 (100%)	155 (100%)	418 (100%)

表 2 ルール実行における解析可能 LHS パターンの割合
Table 2 Proportion of LHS patterns to be analyzed.

ルール	解析可能パターン数	処理要パターン数		解析可能パターンの割合	
		全パターン	非伝播パターン	全パターン比 (%)	非伝播パターン比 (%)
1	—	—	—	—	—
2	—	—	—	—	—
3	119	270	222	44.1	53.6
4	86	156	131	55.1	65.6
5	108	270	222	40.0	48.6
6	86	156	131	55.1	65.6
7	86	156	131	55.1	65.6
8	78	156	131	50.0	59.5
9	150	406	321	36.9	46.7
10	86	156	131	55.1	65.6
11	157	406	321	38.7	48.9
12	86	156	131	55.1	65.6
13	86	156	131	55.1	65.6
14	157	406	321	38.7	48.9
15	86	156	131	55.1	65.6
16	86	156	131	55.1	65.6
17	86	156	131	55.1	65.6
18	78	156	131	50.0	59.5
19	133	406	321	32.8	41.4
20	86	156	131	55.1	65.6
21	78	156	131	50.0	59.5
22	78	156	131	50.0	59.5
23	127	270	222	47.0	57.2
24	141	406	321	34.7	43.9
25	86	156	131	55.1	65.6
26	86	156	131	55.1	65.6
27	323	814	618	39.7	52.3
平均	—	—	—	48.6	58.7
最大	—	—	—	55.1	65.6
最小	—	—	—	32.8	41.4

た.

この変形は，非伝播パターンの割合を見掛け上ふやすための変形ではなく，これらルールが陽に表現する情報に加えて，暗に持つ性質も合わせて解析するための変形である.

表 1 に，「サルとバナナ」(27 ルール) に現れる非伝播パターン，伝播パターンの出現個数と，その割合を示す．ルール全体では，約 8 割が，非伝播パターンであり，ルール中には，非伝播パターンがかなりの頻度で現れることがわかる.

表 2 に，各ルールの実行において，不要となる条件成立判定の割合を示す．「解析可能パターン数」とは，shortcut アークの導入により，条件成立判定処理が不要となる非伝播パターンの数，「処理要パターン数」とは，shortcut アークを導入しない場合に条件成立判定が必要なパターン数 (「全パターン」は，伝播・非伝播パターンを合わせた数，「非伝播パターン」は，その中で，処理が必要な非伝播パターンの数を示す)．「解析可能パターンの割合」とは，「処理要パターン数」に対する「解析可能パターン数」の比を示している (「全パターン比」とは，「処理要パターン数」の「全パターン」に対する「解析可能パターン数」の割合であり，「非伝播パターン比」とは，「処理要パターン数」の「非伝播パターン」に対する「解析可能パターン数」の割合である).

例えば，上述のルール 8 の実行により，伝播・非伝播パターン合わせて 156 パターンの条件成立判定が必要であり，shortcut アークを用いれば，このうち，78 のパターンの成立判定を不要とすることが示されている.

この表から，平均では，処理が必要なパターン全体の 48.6%の成立判定を不要にできることがわかる.

現在，性能評価のためのプロトタイプシステムを用いて評価を進めているが，文献[17]のサルとバナナの例では，1 つの答を出力するために必要な条件成立判定の 41%が shortcut アークの導入により不要となっている.

7. まとめ

本論文では，双方向推論の高速処理アルゴリズムを提案した.

まず，双方向推論の実現に関して，①ルールの非伝播パターンに着目することにより，ルール間の関連図が生成できること，②この関連図は，WME，ゴールの 2 種のトークンが流れるペトリネットとしてみなすことができ，このトークンの動きを制御することにより，「前向き推論と後向き推論とが多様な連携を持つ推論」が実現できること，を明らかにした．この関連

図は，LHS，RHS をそれぞれ弁別ネットに変換し，それぞれを終端ノードで結合し，さらに，共通パターンを飛び越す shortcut アークを設定した ST-NET で表現できることを明らかにした．

次に，簡明な知識表現を持つ推論システムへの本アルゴリズムの適用例を示した．

さらに，本アルゴリズムの評価を，機能，性能の両面から行った．これより，本アルゴリズムは，前向き推論と後向き推論が密に結合した推論を実現可能であり，高速性を有することが得られた．

謝辞 本研究の場，およびその方向付けを与えてくださった，(株) 日立製作所システム開発研究所，堂免信義所長，春名公一副所長，石原孝一郎部長，ならびに，有益な御意見を頂いた同大みか工場，林利弘部長，森清三主任技師に深く感謝いたします．

参考文献

1) Feigenbaum, E. A.: The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering, *IJCAI-77*, pp. 1014-1029 (1977).
2) 小林：プロダクションシステム，情報処理，Vol. 26, No. 12, pp. 1487-1496 (1985).
3) Barr, A. and Feigenbaum, E.: *The Handbook of Artificial Intelligence*, ch. 2, William Kaufmann, Inc., California (1981).
4) 田野：知識処理モデルに基づく推論及び推論制御方式，人工知能学会全国大会（第1回），pp. 139-142 (1987).
5) 礒貝：第2世代の知識工学言語 ART，AI（人工知能）：実用化の夜明け，日経コンピュータ別冊，pp. 105-115，日経マグロウヒル社，東京 (1985).
6) 末田：エキスパートシステム構築支援ツール IREX (2)—知識表現—，第 34 回情報処理学会全国大会論文集，pp. 1567-1568 (1987).
7) McDermott, J., Newell, A. and Moore, J.: The Efficiency of Certain Production System Implementations, in Waterman, D. A. and Hayes-Roth, F. (eds.), *Pattern-Directed Inference Systems*, pp. 155-176, Academic Press, New York (1978).
8) Konolige, K.: An Inference Net Compiler for the Prospector Rule-Based Consultation System, *IJCAI-79*, pp. 487-489 (1979).
9) Mark, W.: Rule-Based Inference in Large Knowledge Bases, *AAAI Aug. 1980*, pp. 190-194 (1980).
10) 鶴田，能見，宮本：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).
11) 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).
12) Miranker, D. P.: TREAT: A Better Match Algorithm for AI Production Systems, *AAAI-87*, pp. 42-47 (1987).
13) 荒屋，百原，田町：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. 768-775 (1987).
14) 荒屋：知識の構造化に関する考察，情報処理学会研究報告，Vol. 87, No. 3 (1987).
15) 田野，増位，舩橋：推論高速化のための弁別ネットワークの動的変形法，第 33 回情報処理学会全国大会論文集，pp. 1417-1418 (1986).
16) 田野，増位，坂口，舩橋：知識ベースシステム構築用ツール 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).
17) 吉村：ルール・ベース・エキスパート・システム構築ツールの「基本形」OPS5，AI（人工知能）：実用化の夜明け，日経コンピュータ別冊，pp. 73-88，日経マグロウヒル社，東京 (1985).

(昭和 63 年 3 月 10 日受付)
(昭和 63 年 7 月 15 日採録)

田野 俊一（正会員）

昭和 33 年生．昭和 56 年東京工業大学工学部制御工学科卒業．昭和 58 年同大学院総合理工学研究科システム科学専攻修士課程修了．同年，(株) 日立製作所に勤務．同システム開発研究所所属．主として，人工知能，知識工学の研究に従事．人工知能学会，計測自動制御学会各会員．

増位 庄一（正会員）

昭和 25 年生．昭和 47 年京都大学工学部電子工学科卒業．昭和 49 年同大学工学研究科電気工学第 2 専攻修士課程修了．同年 (株) 日立製作所入社．システム開発研究所にて制御関連および知識工学の研究に従事，現在同所主任研究員．昭和 56 年 8 月より一年間，カーネギメロン大学客員研究員．計測自動制御学会，人工知能学会，電気学会，IEEE，AAAI 各会員．