

高速双方向推論のための ST-NET 生成アルゴリズム†

田 野 俊 一^{††} 増 位 庄 一^{††} 大 森 勝 美^{†††}

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。我々は、ST-NET アルゴリズムと呼ぶ高速推論方式を提案した。その実現のためには、ルールの条件部、結論部をそれぞれ弁別ネットに交換し、それらをターミナルノードで結合し、条件部、結論部に共通したパターンを飛び越す short-cut アークを設定した構造を有するルール関連図 (ST-NET) を生成する必要がある。本論文では、高速推論が可能な ST-NET を効率良く生成するアルゴリズムを示す。まず、パターン間には、Difference, Matchable, Super-sub, Equal の 4 種の関係があることを明らかにした。次に、パターンを Super-sub 関係によって関連付けた階層グラフ内に Matchable 関係を部分的に表現すれば、これら 4 種の関係を得ることができ、このグラフから ST-NET が直接生成できることを示した。さらに、パターンの数を N とすると、パターンの階層を用いることにより、 $O(N \log N)$ の計算量で、ST-NET が生成できることを示した。

1. はじめに

知識工学は、人工知能研究で得られた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、専門家並みの能力を持つ計算機システムを構築することが可能となる。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、前向きと後向き推論の 2 方式に大別できるが³⁾、人間は、それらを交互に関連させ問題解決を行うことが多い⁴⁾。このため、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要件であり、さらに、実用エキスパートシステムを得るためには、これらの推論を高速に行うことが必須である^{5), 13), 14)}。

前向き推論に関しては、Rete アルゴリズムをはじめとする多くの高速処理方式があり^{9)-11), 15)}、我々も、数種類のアルゴリズムを提案してきた⁶⁾⁻⁷⁾。一方、双方向推論の高速実現アルゴリズムについて、我々は、ST-NET アルゴリズムと呼ぶ双方向推論の高速処理アルゴリズムを提案した⁸⁾。

我々の提案した ST-NET アルゴリズムは、高速双

方向推論のために ST-NET と呼ぶルール関連図を必要とする。ST-NET は、従来の前向き推論の高速アルゴリズムで用いていたネットワークとは構造が全く異なり、複雑化しているため、従来の変換方式を用いることはできず、新たな変換アルゴリズムが必要である。

また、これら高速推論アルゴリズムは、実行効率を高めるが、弁別ネット、Rete ネットの生成等の事前処理が大きくなり、ルールの追加、削除、修正が容易であるという知識工学ツールのよさが失われる可能性が大きい。このため、事前処理の効率化の研究が、最近進められるようになってきた¹²⁾。

つまり、ST-NET の生成においては、どのように ST-NET を生成するかに加え、生成処理自体の効率化が必要である。

本論文では、ルール条件部、結論部に出現するパターン間の性質を用いて、効率的な ST-NET を効率的に生成するアルゴリズムを提案する。

以下、次章では、ST-NET アルゴリズムの概要を説明する。第 3 章では、解決すべき課題を明確にし、パターン間に成立する関係の性質を示し、生成の基本的考え方を述べる。第 4 章では、ST-NET 生成アルゴリズムを明らかにする。第 5 章で、評価を行う。

2. ST-NET アルゴリズムの概要

2.1 知識表現

知識は、対象の状態を表すワーキングメモリエレメント (WME) およびルールで表現する。

WME は、述語形式を用いて表す。ルールは、WME の状態に関する記述よりなる条件部 (Left Hand Side) と、条件部が成立した場合に生成できる WME 記述

† ST-Network Generation Algorithm for Fast Bidirectional Inference by SHUN'ICHI TANO, SHOICHI MASUI (Systems Development Laboratory, Hitachi, Ltd.) and KATUMI OHMORI (Omika Works, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所

††† (株)日立製作所大みか工場

ルール 1	IF	A (a1 , a2 , ?X)	THEN	C (c1 , c2 , ?X)
ルール 2	IF	A (a1 , a2 , a3) B (b1 , b2 , b3)	THEN	C (c1 , c2 , c3)
ルール 3	IF	B (b1 , b2 , b3) C (c1 , c2 , ?Y)	THEN	D (d1 , d2 , ?Y)
ルール 4	IF	D (d1 , d2 , d3)	THEN	E (e1 , e2 , e3)

図 1 ルールの例

Fig. 1 Examples of rule.

よりなる結論部 (Right Hand Side) より構成される。1つの WME に関する記述をパターンと呼ぶ。LHS は、複数のパターンより構成され、RHS は、1つのパターンより成る。図 1 に示すルール 1 は、述語名が A、第 1 引数が a1、第 2 引数が a2 である WME が存在した場合に、述語名が C、第 1 引数が c1、第 2 引数が c2、第 3 引数が ?X である WME を生成できることを示している。“?X” は、変数を表し、同一ルール内では、同一値となる。

2.2 ST-NET アルゴリズムの概要

ST-NET アルゴリズムは、以下に示す 2 つのアイデアを実現したものである。

(1) 事前解析によるパターンマッチの効率化

図 1 のルールを用いて、前向き推論を考える。ルール 3 が実行された場合、その結果、D(d1, d2, ...) が生成される (“...”の値は、実行時に決定される)。これにより、新たに実行可能になるルールを捜す必要があるが、明らかに、ルール 1, 2, 3 の条件部は無関係であり、ルール 4 に対しては、第 3 引数が d3 であることだけ確認すれば、実行可能であると判定できる。後向き推論においても同様のことが言える。例えば、ルール 3 を用いた場合、C(c1, c2, ?Y) のサブゴールが生成され、これを導くルールを捜す必要があるが、ルール 3, 4 は適用不能、ルール 1 は、常に (何も確認することなく) 適用可能、ルール 2 は、?Y が c3 であれば適用可能であると判断できる。すなわち、ルール間の関連を解析しておけば、前者の場合「第 3 引数 = d3」、後者の場合「?Y = c3」以外のすべての判定処理が不要になる。

これを実現するためには、ルールのパターン間を解析し、不要なパターンマッチ条件を含まないルール関連図を生成し、これを用いて推論を行えば良い。

(2) ルール関連図内に 2 種のトークンを持たせることによる双方向推論の多様化、高速化

上記で得られるルール関連図は、WME、ゴールの 2 種類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る。WME を関連図内に流すことにより前向き推論が、ゴールを関連図内に流すことにより後向き推論が、2 種のトークンの流れを協調させることにより様々なタイプの双方向推論が実現可能である。

さらに、ペトリネットと同様に、2 種のトークンを推論の状態として記憶すれば、過去のパターンマッチの処理結果を用いることに相当し、処理を効率化できる。

上記の処理に必要なルール関連図を、①forward-root ノードを頂点ノードとする弁別ネットでルール条件部を表現 (LHS ネットと呼ぶ)、②backward-root ノードを頂点ノードとする弁別ネットでルール結論部を表現 (RHS ネットと呼ぶ)、③LHS ネットと RHS ネットを、それぞれの弁別ネットの終端ノードであるターミナルノードで結合、④LHS ネットと RHS ネットで共通に現れる条件を飛び越す shortcut アークを設定、のような構造で表したネットワークが ST-NET である。

ST-NET の構造を図 2 に示す。上記(1)で説明した不要な条件判定は、LHS と RHS に共通して現れる条件部分であり、これを飛び越すように、SCUT アークを設定してある。

図 1 のルールを表す ST-NET を図 3 に示す。例えば、ルール 3 の条件部は f-root に続く「述語名 = C」「\$1 = c1」「\$2 = c2」「\$3 = ?Y」の並びで表され、結論部は、b-root へと続く「\$3 = ?Y」「\$2 = d2」「\$1 = d1」「述語名 = D」の並びで表現されている (「\$1 = c1」は、第 1 引数が c1 であることを表している)。結論部を表す「\$3 = ?Y」と「\$2 = d2」のノード間にルール 4 の「\$3 = d3」への SCUT アークが設定され

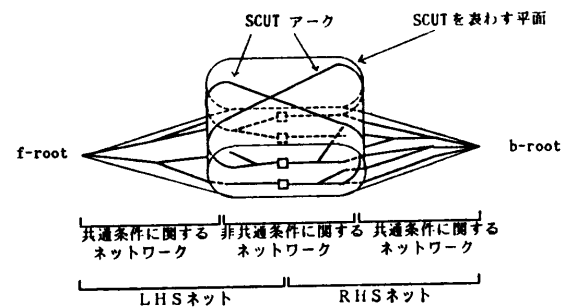


図 2 ST-NET の構造

Fig. 2 Structure of the ST-NET.

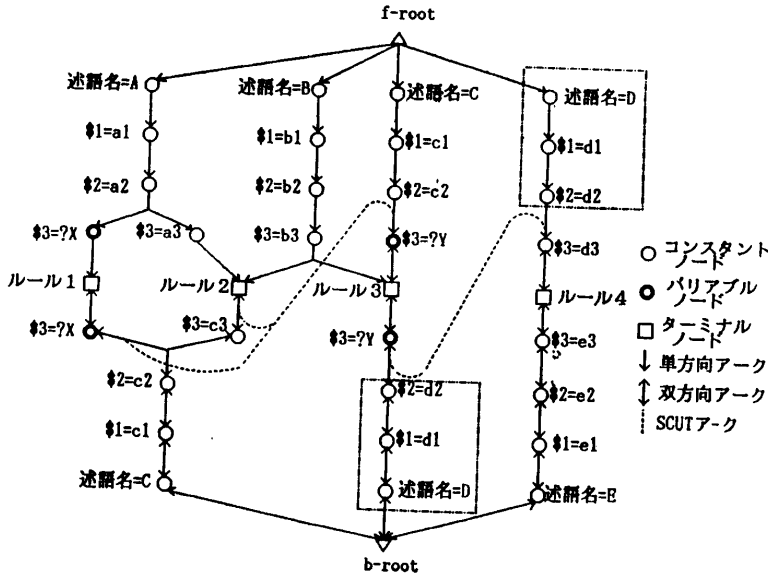


図 3 ST-NET 表現の例
Fig. 3 Example of a ST-NET.

ており、ルール 3 の実行により、生成される WME は、「\$3=d3」だけの条件チェックで、ルール 4 の条件部を満たし、他の条件部には無関係であることを示している。

すなわち、図 3 において一点鎖線で囲んだ部分が、ルール 3 の結論部とルール 4 の条件部の組での共通パターンであり、これを飛び越すように SCUT アークが設定されている。

3. ST-NET 生成の基本的考え方

3.1 ST-NET 生成のために解決すべき課題

ST-NET アルゴリズムの高速性は、

- ①いかに効率的な LHS ネットを作るか
- ②いかに効率的な RHS ネットを作るか
- ③いかに効率的な SCUT アークを設定できるかに依存する。

LHS ネット, RHS ネットは、弁別ネットワークであるから、効率化するという事は、少ない条件判定で、パターンの成立・不成立判定が行えること、すなわち、共通に現れる条件を 1 つにまとめることに対応する。SCUT アークは、事前解析により、パターンマッチが可能と判定されたネットワークにのみ、データを流すように設定しなければ、効率的であるとは言えない。

SCUT アークは、LHS ネット, RHS ネットを結合するアークであるから、効率的なアークができるかどうかは、LHS ネット, RHS ネットの構造に影響される。つまり、まず、効率の良い RHS, LHS ネット

を生成し、次に、これらのネットを結ぶ SCUT アークを生成しても、効率の良い SCUT アークが設定できるとは限らないのである。すなわち、三者を統合し生成する必要がある。

さらに、SCUT アークを設定するためには、ルール間の関連を解析する必要がある。しかし、この判定処理は、基本的に、LHS パターンと RHS パターンのすべての組に対するマッチング処理であり、処理量が莫大になる。

以上のように、ST-NET 生成アルゴリズムでは、①LHS ネット, RHS ネット, SCUT アークの生成を統合して行うこと、かつ、②LHS パターンと RHS パターンの部分的な組合せでのパターンマッチ処理で ST-NET

が生成できること、の 2 点が課題となる。

3.2 パターン間の関係とその性質

パターンは、対象の状態（すなわち、ワーキングメモリの状態）を規定するものであり、対象の状態が満たすべき条件記述である。すべての状態の集合を T とするとパターン Pa, Pb の示す意味は、パターンを満足する状態の集合で表すことができる。パターン Pa, Pb を満たす状態の集合をそれぞれ Sa, Sb とすると、

$$Sa = \{x | x \in T, x \text{ は } Pa \text{ を満足する}\}$$

$$Sb = \{x | x \in T, x \text{ は } Pb \text{ を満足する}\}$$

と定義できる（以下では、 $Sa, Sb \neq \phi$ である、つまり、恒偽パターンが存在しないとする）。この集合の関係は以下の 4 種に分類できる。

交わりがない場合： $Sa \cap Sb = \phi$

部分的に交わる場合： $Sa \cap Sb \neq \phi, Sa \cap Sb \neq Sa, Sa \cap Sb \neq Sb$

包含される場合： $Sa \supset Sb \text{ or } Sa \subset Sb$

等しい場合： $Sa = Sb$

この分類に基づき、パターン間の関係を定めることができる。

(1) Difference 関係 ($\leftarrow D \rightarrow$)

$Sa \cap Sb = \phi$ の場合、 Pa と Pb には、D 関係があるといい、 $Pa \leftarrow D \rightarrow Pb$ と記述する。例えば、 $A(1, ?X, 2)$ と $A(2, ?Y, 2)$ は D 関係である。

(2) Matchable 関係 ($\leftarrow M \rightarrow$)

$Sa \cap Sb \neq \phi, Sa \cap Sb \neq Sa, Sa \cap Sb \neq Sb$ の場合、 Pa

と Pb には, M関係があるといい, $Pa \leftarrow M \rightarrow Pb$ と記述する. 例えば, $A(1, 2, ?X) \leftarrow M \rightarrow A(1, ?X, 3)$ である.

(3) Super-sub 関係 ($\leftarrow S \rightarrow$)

$Sa \supset Sb$ or $Sa \subset Sb$ の場合, Pa と Pb には, S関係があるといい, $Sa \supset Sb$ ならば $Pa \leftarrow S \rightarrow Pb$, $Sa \subset Sb$ ならば $Pb \leftarrow S \rightarrow Pa$ と記述する.

$Pa \leftarrow S \rightarrow Pb$ の場合, Pa は, Pb の上位パターン, Pb は, Pa の下位パターンと呼ぶ. 例えば, $A(1, ?X, ?Y) \leftarrow S \rightarrow A(1, ?X, ?X)$ である.

(4) Equal 関係 ($\leftarrow E \rightarrow$)

$Sa = Sb$ の場合, Pa と Pb には, E関係があるといい, $Pa \leftarrow E \rightarrow Pb$ と記述する. E関係を持つパターンは, 意味的にみて全く等しい. 例えば, $A(?X, 1, ?Z)$ は, $A(?Y, 1, ?Z)$ とE関係があるが, $A(?Y, 1, ?Y)$ とはE関係はない.

これら4種の関係には, 以下のような性質がある.

(a) パターン間には, D関係, M関係, S関係, E関係のいずれか1つの関係が必ず成り立つ.

これは, パターンを満たす状態の集合 (Sa, Sb) の関係による定義から明らかである.

(b) S関係は, transitive である.

$Pa \leftarrow S \rightarrow Pb, Pb \leftarrow S \rightarrow Pc$ ならば, $Sa \supset Sb, Sb \supset Sc$ であり, $Sa \supset Sc$ となるので, $Pa \leftarrow S \rightarrow Pc$ が成立する.

(c) $Pa \leftarrow M \rightarrow Pb$ であり, $Pc \leftarrow S \rightarrow Pb$ を満たす Pc と Pa にS関係がないならば, $Pa \leftarrow M \rightarrow Pc$ が成立する.

$Pa \leftarrow M \rightarrow Pb$ より $Sa \cap Sb \neq \emptyset, Pc \leftarrow S \rightarrow Pb$ より $Sc \supset Sb$ であるので, $Sa \cap Sc \neq \emptyset$ となる.

また, Pa, Pc にS関係がないので, Sa と Sc 間には, 包含関係はなく, $Sa \cap Sc \neq Sa, Sa \cap Sc \neq Sc$ である.

したがって, $Sa \cap Sc \neq \emptyset, Sa \cap Sc \neq Sa, Sa \cap Sc \neq Sc$ となり, $Pa \leftarrow M \rightarrow Pc$ が成立する.

(d) $Pa \leftarrow D \rightarrow Pb$ であり, $Pb \leftarrow S \rightarrow Pc$ ならば, $Pa \leftarrow D \rightarrow Pc$ が成立する.

$Pa \leftarrow D \rightarrow Pb$ より, $Sa \cap Sb = \emptyset, Pb \leftarrow S \rightarrow Pc$ より $Sb \supset Sc$ であるので, $Sa \cap Sc = \emptyset$ となり, $Pa \leftarrow D \rightarrow Pc$ が成立する.

(b)の性質より, パターン間のS関係を, 上位パターンと下位パターンを有向アークで結合した階層グラフで表せることがわかる. また, このS関係の階層グラフにおいて, (c)の性質により, Pa と Pb にM

関係が存在すれば, その上位パターン間にも, S関係が存在しない限り, M関係が成立する. つまり, M関係が, 階層グラフの上位に伝播できることを示している. さらに, (d)の性質により, Pa と Pb にD関係が存在すれば, その下位パターン間にも, D関係が成立する. つまり, D関係が, 階層グラフの下位に伝播できることを示している.

次に, 4種の関係と, パターンマッチとの関連について説明する.

$Pa \leftarrow E \rightarrow Pb$ であれば, Pa を満たす状態は, 常に Pb を満たし, Pb を満たす状態は, 常に Pa を満たす.

$Pa \leftarrow D \rightarrow Pb$ であれば, Pa を満たす状態は, 常に Pb を満たさない. また, Pb を満たす状態は, 常に Pa を満たさない.

$Pa \leftarrow S \rightarrow Pb$ であれば, Pa を満たすデータは, Pb を常に満たすとは限らないが, Pb を満たすデータは, 常に Pa を満たす.

$Pa \leftarrow M \rightarrow Pb$ であれば, Pa を満たすデータは, Pb を常に満たすとは限らない. また, Pb を満たすデータは, 常に Pa を満たすとは限らない.

以下, E関係を持つパターンは, 1つのパターンであるとみなし, E関係については考慮しない.

3.3 IM グラフを用いた関係表現

基本的に, パターン間の関係を知るためには, すべてのパターンの組合せに対して, その関係を判定しなければならず, 処理量がパターン数の2乗に比例して増加してしまう. ところが, 前節で示した性質(c)(d)より, D関係, M関係は, S関係より計算できる部分がある. また, (b)より, S関係は, transitive であるから, この性質を用いれば, S関係の階層を表すグラフは, 効率良く生成できる.

そこで, すべての状態が満足する仮想的なパターンを頂点とするS関係を用いたパターンの階層グラフを考える. このグラフは, 一意に定まる.

S関係を表すアーク (Sアークと呼ぶ) を実線で

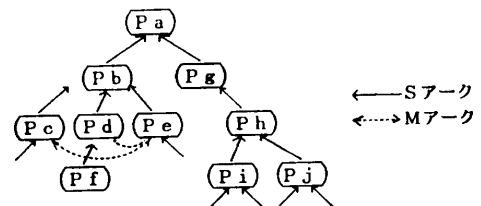


図4 パターンの階層グラフの例
Fig. 4 Example of pattern hierarchy.

表した例を図4に示す。例えば、このグラフから、 $P_a \leftarrow S \rightarrow P_b \leftarrow S \rightarrow P_c$ であることが得られる。

次に、あるパターンに直接結合した下位パターンが複数あり、それらのパターン間に、M関係がある場合、それらをアーク (Mアークと呼び、点線で表す) で結ぶ。例えば、 P_b に直接結合した下位パターンは、 P_c, P_d, P_e であり、M関係が、 P_c と P_e, P_d と P_e にあれば、それらのパターン間をMアークで結合する。

図4の例では、 P_b の直下のパターン P_c, P_d, P_e 間のMアークから、 $P_c \leftarrow M \rightarrow P_e, P_d \leftarrow M \rightarrow P_e$ であると判定できるとともに、 $P_c \leftarrow D \rightarrow P_d$ であることも表している。また、 P_a の直下のパターン P_b, P_g 間、 P_h の直下のパターン P_i, P_j 間、には、Mアークがないので、 $P_b \leftarrow D \rightarrow P_g, P_i \leftarrow D \rightarrow P_j$ であることも示している。すなわち、あるパターンの直下のパターン間にMアークが存在しない場合、D関係であることを示している。

S関係による階層グラフに、直下パターン間に成立するM関係を表すMアークを加えたグラフを Implication/Matchable グラフと呼ぶ。

3.4 IM グラフと ST-NET の関連

IM グラフは、S関係の階層グラフを含むから、弁別ネットを直接生成することができる。

すなわち、IM グラフにおいて、パターン P_a とパターン P_b が、直接S関係で結合されている場合、つまり、 $P_a \leftarrow S \rightarrow P_b$ のアークが存在する場合、このアーク上に、パターン P_a と P_b の条件の差を表すノードを生成し、パターンが成立したことを表すターミナルノードを生成すると、IM グラフ内に存在するパターンに関する弁別ネットが生成できる。

例えば、図5(a)のIMグラフにおいて、パターン P_1 と P_2 を結ぶアーク上に、パターン P_1 と P_2 の条件の差、つまり、「 $\$2=2$ 」を生成し、さらに、パターン P_1, \dots, P_7 が成立したことを表すターミナルノードを生成することにより、図5(b)の弁別ネット

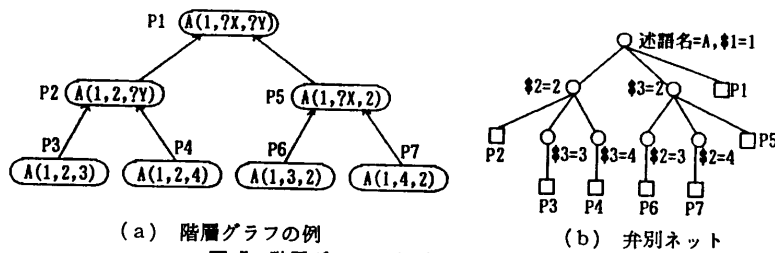


Fig. 5 Generation of a discrimination network.

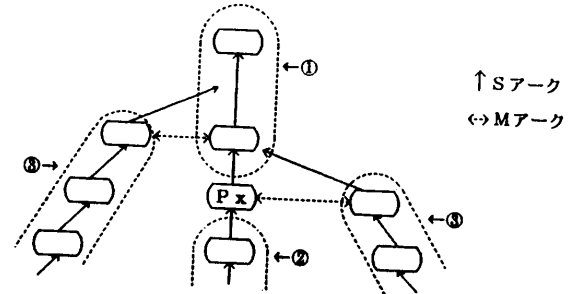


図6 IM グラフにおけるパターン間の関係
Fig. 6 Relation among patterns.

に直接変換できる。

上位のパターンは、下位のパターンにとって、共通の条件を表している。すなわち、IM グラフの構造に沿って、直接弁別ネットを生成することは、共通条件を1つのノードで表すことに対応し、同一の条件チェックを複数回判定することを避けることができ、効率的ネットワークとなっている。

また、LHS パターン、RHS パターンをすべて1つのIM グラフとして表すことにより、SCUT アークの設定に関する情報が得られる。

図4の例を用いて説明する。Pd を RHS パターン、他を LHS パターンとする。あるルールの実行により Pd にマッチする状態が生成された場合、Pd の上位パターンである Pb, Pa は必ず満たされ、また、Pd の下位パターンである Pf は、Pd と Pf のパターンの差をチェックするだけでパターンが満足されるかを判定できることを表している。すなわち、Pd の上位パターン Pb, Pa はパターンマッチの処理なしに、Pd とパターンマッチすると判定できるパターンであり、SCUT アークによって直接結合でき、Pd の下位パターン Pf は、部分的なパターンマッチの処理後、パターンマッチできるかどうか判定できるパターンであり、SCUT アークによって、必要となるパターンマッチ条件を表すネットワークに結合する。

また、Pd は、Pe とMアークで結ばれているため、Pd にマッチする状態は、Pe 以下のパターンを満たす

可能性がある。そのため、Pf と同様に、SCUT アークによって、必要となるパターンマッチ条件を表すネットワークに結合する。一方、Pc 以下のパターンおよび Pg 以下のパターンとは、SCUT アークで結合する必要はないことがわかる (Pc と Pd とはD関係であり、また、

Pg は Pb と D 関係であるから Pb の下位パターンである Pd とも D 関係である).

以上を図 6 を用いてまとめる. あるパターン Px に関する SCUT アークは, (a) 上位パターン①が成立したことを表すノードへ, (b) 下位パターン②の成立判定に必要となるチェックを表すノードへ, (c) 上位パターンおよびそのパターン自身と M アークで結合されているパターン③の成立判定に必要となるチェックを表すノードへ, データが流れるように設定すればよい.

以上のように, IM グラフは, SCUT アークに関する情報も表している.

4. ST-NET 生成アルゴリズム

ST-NET の生成は, ①LHS, RHS パターンを構成要素とする IM グラフを生成, ②得られた IM グラフを用いて, LHS ネット, RHS ネット, SCUT アークを生成, の 2 段階で行う.

4.1 IM グラフの生成法

図 7 に生成フローを示す. すべてのパターンの上位となる仮想的パターンとしてルートパターンを設け, これをパターン 0 とする. NOW は, 新たに階層グラフに登録するパターン番号 (1~) を, POS は, 解析中の階層グラフ内の位置を示す. 直接結合している下位パターンを直下パターン, 直接結合している上位のパターンを直上のパターンと呼ぶ.

まず, 初期化として, ルートパターンをパターン 0 とし, NOW=1, POS=0 とする. 次に, POS のパターンの直下パターンを BELOW に入れる. 例えば, 図 8 (a) のグラフが既に生成されており, 現在の POS が 1 である場合, BELOW は, {2, 4, 7} となる. BELOW が, 空である場合は, POS パター

ンがグラフのリーフ(葉)であることに対応し, NOW パターンと POS パターンを結合する. 例えば, 図 8 (a) で, POS=7, NOW=8 の場合, 図 8 (b) のようになる.

BELOW が空でなければ, BELOW 内のパターンと NOW パターンでパターンマッチを行う (パターン記述から解析的にパターン間の関係を導くためには文献 16) の方式がある). BELOW の中には, NOW の上位であるパターンと下位であるパターンが混在して現れることはないので, 以下の 3 つの場合を考えればよい.

- (1) NOW の上位に位置すべきパターンがある場合
上位に位置すべきパターンから, 1 つ選択し POS

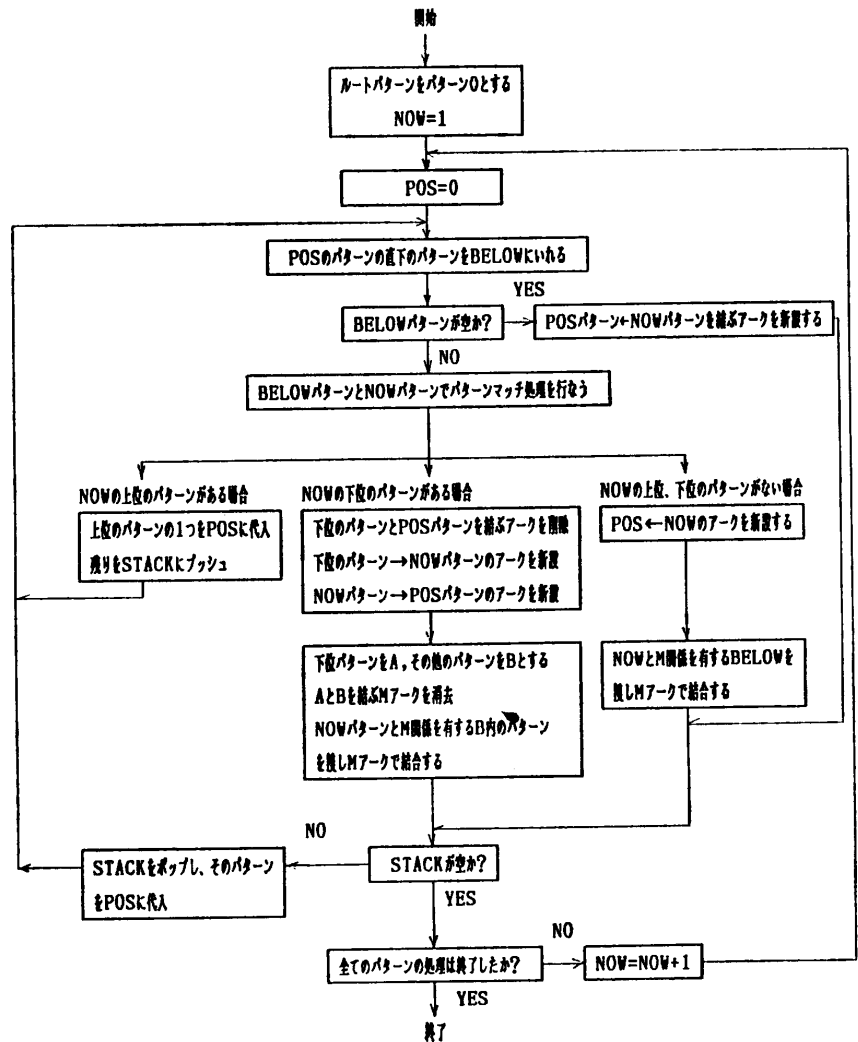


図 7 IM グラフの生成フロー
Fig. 7 Generation flow of IM graph.

に代入し、残りを STACK にプッシュする。例えば、2, 4, 7 のうち 2, 4 が, NOW の上位に位置すべきパターンである場合, POS=2 とし, STACK に 4 をプッシュする。

(2) NOW の下位に位置すべきパターンがある場合

下位に位置すべきパターンと POS パターンを結ぶアークを削除し、それらのパターンと NOW を結び、NOW と POS を結ぶ。例えば、2, 4, 7 のうち 2, 4 が NOW の下位に位置すべきパターンである場合、図 8 (c) のように結合する。BELOW 中で下位に位置すべきパターンを A とし、以外のパターンを B とする。A に含まれるパターンと B に含まれるパターン間の M アークを消去し、NOW パターンと M 関係を有する B 内のパターンを M アークで結合する。例えば、上記の例では、パターン 2 と 4、パターン 4 と 7 を結ぶ M アークが消去され、パターン 8 と 7 に M 関係がある場合は、パターン 8 と 7 を結ぶ M アークが生成される。

(3) NOW の上位あるいは下位に位置すべきパターンがない場合

NOW と POS を結合する。例えば 2, 4, 7 と上位、下位の関係がない場合、図 8 (d) のように結合する。NOW パターンと M 関係を有する BELOW 内のパターンを M アークで結合する。

これらの処理を、すべてのパターンに対して行えば IM グラフが生成できる。

4.2 IM グラフの ST-NET への変形法

ST-NET の生成アルゴリズムを図 9 に示す。前節で説明したアルゴリズムにより、LHS パターン、RHS パターンを構成要素とする IM グラフを生成し、各パターンにパターンが成立したことを示すターミナルノードを対応付ける。LHS パターンを○、RHS パターンを□、ターミナルノードを◆印で示した例を図 10 (a) に示す。

まず、SCUT アーク設定のために、IM グラフ内に SC アークを張る。SC アークは、

(i) ターミナルノードへの枝と、そのターミナルノードを持つパターンの直下パターンへの枝を結合するアーク

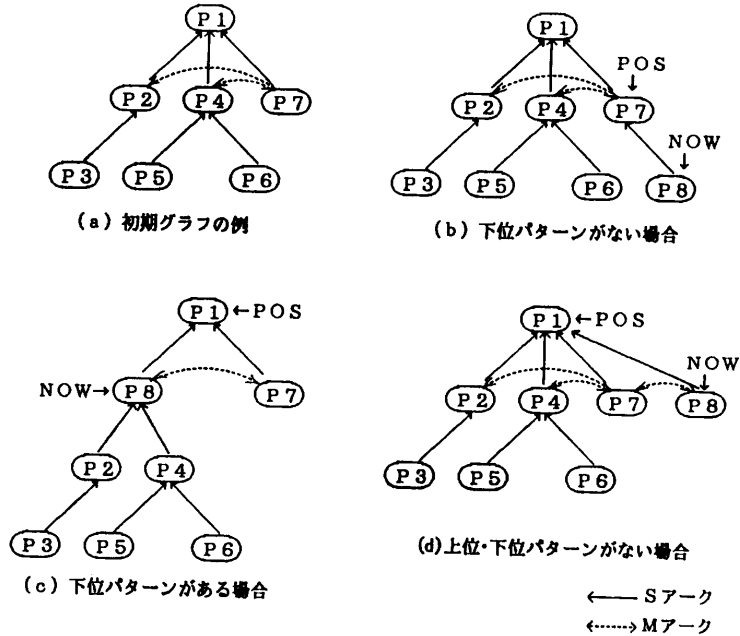


図 8 IM グラフの生成法
Fig. 8 Generation of IM graph.

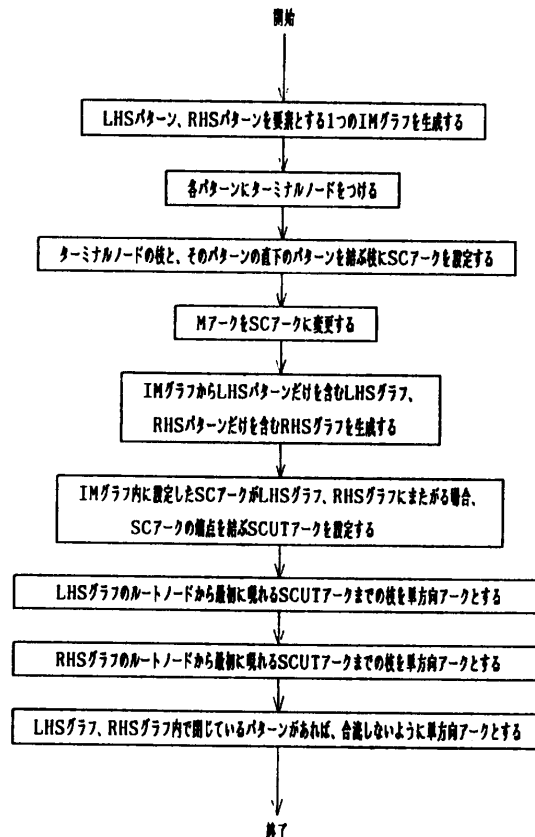


図 9 ST-NET の生成フロー
Fig. 9 Generation flow of ST-NET.

(ii) Mアークで結合されたパターンへの枝どうしを結合するアーク

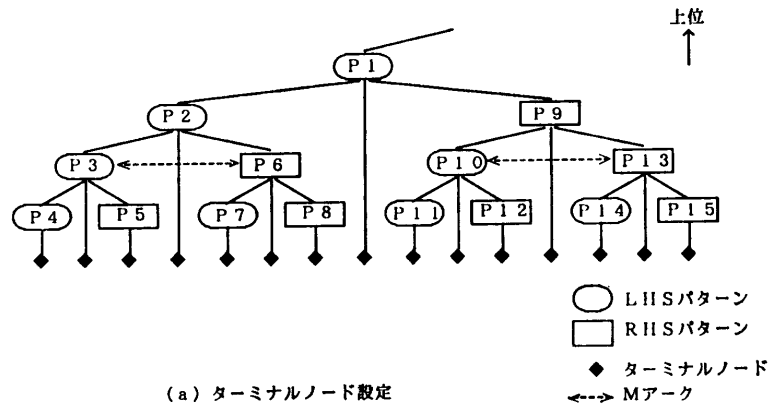
である。図 10(a) の IM グラフの場合は、図 10(b) のように SC アークが設定される。例えば、SC アーク X は、P3 のターミナルノードへの枝と、P3 の直下パターン P4 への枝を結ぶ上記(i)のアークであり、SC アーク Z は、Mアークで結合された P3 と P4 への枝を結ぶ上記(ii)の SC アークである。

次に、このグラフを2つに分解する。LHS ネットを生成するためには、LHS パターンだけを含む IM グラフが必要であるので、IM グラフのリーフより LHS パターンを表すノードが現れるまで、ノードを消去、すなわち、RHS パターンを表すノードがリーフノードまで連続している部分を消去し、RHS パターンのターミナルノードおよびそのアークを消去する。このグラフを LHS グラフと呼ぶ。同様にして RHS グラフを生成する。図 11 において、ターミナルノードより上のグラフが LHS グラフ、下が RHS グラフに対応している。

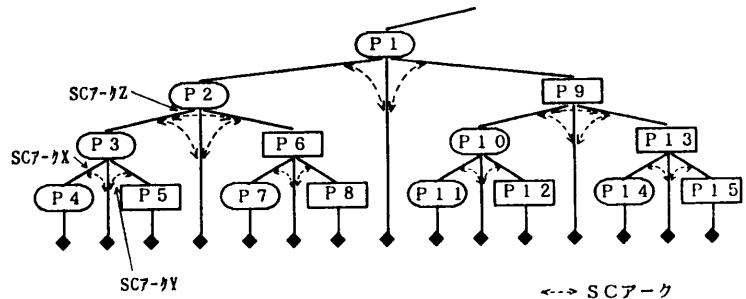
次に、SCUT アークの生成について述べる。

上記(i)のタイプの SC アークは、図 6 における ① ② の関係を、(ii)のタイプの SC アークは、図 6 における ③ の関係を表している。

例えば、図 10(b) の (i) のタイプである SC アーク Y は、P5 が生成されれば P3 が成立すること(すなわち、図 6 での ① の関係)、逆に、P3 が生成されれば P5 以下の条件成立判定が必要であること(すなわち、図 6 での ② の関係)を表しており、(ii)のタイプである SC アーク Z は、P3 が生成されれば P6 以下の条件成立判定が必要であること(すなわち、図 6 での ③ の関係)を表している。



(a) ターミナルノード設定



(b) SCアークの生成

図 10 IM グラフの ST-NET への変換過程
Fig. 10 Transformation from IM graph to ST-NET.

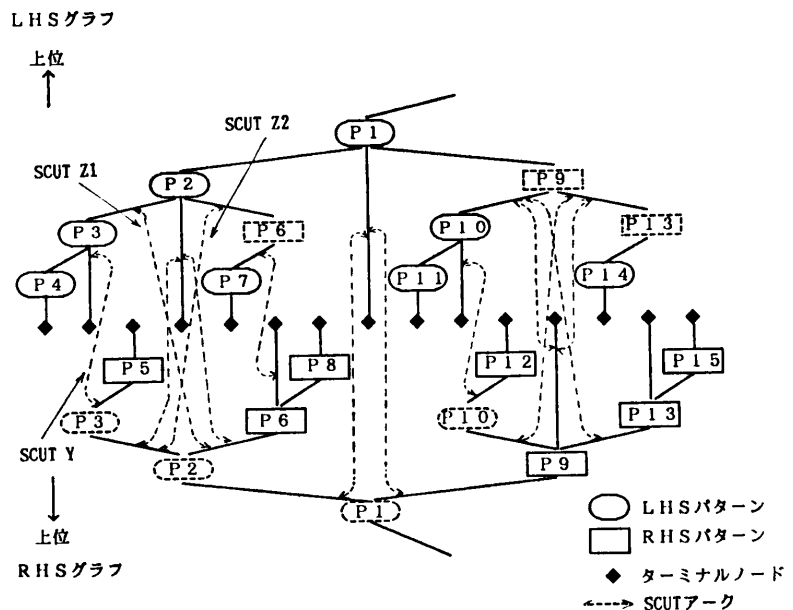


図 11 SCUT アークの設定

Fig. 11 Generation of shortcut arc.

したがって、IM グラフ内に設定した SC アークの端点が、それぞれ LHS グラフ、RHS グラフ内であれば、つまり、両方のグラフにまたがれば、それを

SCUT アークとして設定することにより SCUT アークが生成できることになる。

この操作により図 11 に点線で示した SCUT アークが生成される。例えば、図 10 (b) の SC アーク X の両端、つまり、P3 のターミナルノードへの枝と P3 の直下パターン P4 への枝は、どちらも LHS グラフ内にしかないので SCUT アークとはならない。SC アーク Y の両端、つまり、P3 のターミナルノードへの枝と P3 の直下パターン P5 への枝は、それぞれ、LHS グラフ、RHS グラフにあるので、SCUT アーク Y となる。SC アーク Z は、P3 への枝、P6 への枝をその両端としており、この枝は、LHS、RHS グラフの双方にあるので、LHS グラフ内の P3 への枝と RHS グラフ内の P6 への枝、および、LHS グラフ内の P6 への枝と RHS グラフ内の P3 への枝を結ぶ 2 つの SCUT アーク Z1, Z2 が生成される。

例えば、P5 のパターンが生成された場合 (WME の生成となる) は、図 10(a) のグラフから、P3, P2, P1 の LHS パターンは無条件に成立し、P6 以下のパターンは、条件成立の可能性があるため、条件成立判定が必要となり、それ以外のパターンは、成立しないことがわかる。図 11 に示したグラフでは、RHS グラフ内の P5 から P1 の間の 4 つの SCUT アークがあり、それぞれ、P3 のターミナルノード、P2 のターミナルノード、P6 以下の条件判定ノード、P1 のターミナルノードの処理を行うことになる。逆に、P10 のパターンが生成された場合 (ゴールの生成となる) は、図 10(a) のグラフから、P9 の RHS パターンは無条件に成立し、P12 と P13 以下のパターンは、条件成立の可能性があることがわかる。図 11 のグラフでは、LHS グラフ内の P10 から P1 の間に 2 つの SCUT アークがあり、それによって、RHS グラフの処理を行うと、それぞれ、P9 ターミナルノード、P13 以下の条件判定ノードの処理を行うことになる。このように、図 11 の SCUT アークは、図 10 (b) が表している SCUT アーク設定に関する情報を正しく表現していることがわかる。

最後に、このグラフを図 5 (a) を (b) に変形したように、上位パターンとのパターンの差をノードとして枝に付加し、アークにデータの流れる方向に関する情報を付加することにより、ST-NET が得られる。

5. 評価

推論の高速化に関しては多くのアルゴリズムが提案

されているものの、高速推論に必要となるネットワークの生成法に関する報告は、ルールの追加・削除・更新に対し、もとのネットワークを部分的に修正することで対応する方式¹²⁾が提案されているだけである。

さらに、ST-NET アルゴリズムは、従来方式とは異なる新たなネットワークを必要とするので従来方式との比較は困難である。

本アルゴリズムの評価のポイントは、推論効率の良いネットワークを生成できるか (機能の評価)、および、生成のために要する処理量はどのくらいか (性能の評価) である。そこで、これら 2 つの観点から評価を行う。

5.1 生成した ST-NET の質

まず、LHS ネット、RHS ネットの効率について評価する。第 3 章で示したように、LHS ネット、RHS ネットは、弁別ネットであるから、効率化することは、少ない条件判定で、パターンの成立・不成立が行えること、すなわち、共通に現れる条件を 1 つにまとめることに対応する。LHS ネット、RHS ネットは、それぞれ、パターンの S 関係に基づいた階層グラフから生成され、その結果、上位パターンの条件は共通化され、効率的なネットワークとなっている。

効率的な SCUT アークとは、事前解析によりパターンマッチが可能と判定されたネットワークにのみ、データを流すように設定されたアークである。前章で示したように、上位パターンに対しては、条件チェックなしでパターンの成立が判定できるように、下位パターンに対しては、必要となる条件を表すノードからの処理となるように、SCUT アークが設定されている。また、パターンマッチ不可能な (M アークがない) ネットワークへは、SCUT アークは設定されていない。すなわち、条件判定処理が必要となる部分にだけ、データが流れるように、SCUT アークが設定されている。

以上のように、本アルゴリズムを用いれば、効率の良い LHS ネット、RHS ネット、SCUT アークを生成できる。

5.2 ST-NET 生成に要する処理量

ST-NET を生成するには、不要パターンマッチ処理を削除するために、パターン間においてマッチするかどうかの判定を行う必要があり、パターンの個数を N とすると、処理量は、少なくとも $O(N^2)$ になる。

本アルゴリズムは、図 7, 9 で示したように、IM グラフ生成と、IM グラフから ST-NET への変換の

2ステップより構成される。

IM グラフから ST-NET へは直接行うことができるので、 $O(N)$ の処理量となる。また、IM グラフ生成の処理量は、階層グラフ生成の処理量と等しい。

階層グラフ生成の計算量は、階層グラフが1本の直線となり、最上位パターンから順々に図7のフローで処理した場合最高となり、 $O(N^2)$ となる。また、階層グラフが1本の直線となり、最下位パターンから順々に図7のフローで処理した場合最低となり、 $O(N)$ となる。

しかし、ルール条件部、結論部に現れるパターンの包含関係は、「推論が、ルール結論部のパターンにより WM (対象世界) の状態を変化させ、それをルール条件部のパターンで判定することにより進む」ことを考えれば、上記のようになることは考えられず、階層グラフは、1本の直線ではなく、バランスの良い木になると考えられる。

そこで、各ノードでの平均分岐数を m とすると、階層グラフを生成するために必要なパターンマッチの回数は、階層グラフの深さが $\log mN$ に比例することから考えて、 $O(N \log N)$ になる。

つまり、本アルゴリズムを用いれば、 $O(N \log N)$ の処理量で ST-NET を生成することができる。

従来のルール条件部のみを対象とする方式においても、共通に現れる条件を1つのノードで表現するためには、ルール条件部のパターンを構成要素とする階層グラフを生成する必要がある。すなわち、結論部のパターンに関する弁別ネット、および、SCUT の生成が不要な従来のネットワーク生成においても、階層グラフ生成に要する処理量は必要なのである。

つまり、ST-NET の構造は複雑であるにもかかわらず、本方式を用いれば、その生成のための計算量は、オーダ的には、従来方式のネットワークを生成するのに要する処理量と同程度ですむことになる。

6. ま と め

高速双方向推論方式である ST-NET アルゴリズムは、従来の高速推論アルゴリズムで用いられているルールネットワークとは構造の大きく異なる ST-NET と呼ぶルール関連図が必要である。本論文では、ST-NET を生成するアルゴリズムを提案した。

まず、ルール条件部・結論部を構成するパターン間には、Difference, Matchable, Super-sub, Equal と呼ぶ4種の関係があることを示し、それらの基本的性質

を明らかにした。次に、これら基本性質を用いて

- ①パターンを Super-sub 関係によって関連付けた階層グラフ内に Matchable 関係を部分的に表現した IM グラフを生成
- ②IM グラフ内に SC アークを設定
- ③グラフを2つに分解し、LHS ネット、RHS ネット、SCUT アークを生成

の処理フローにより ST-NET を生成するアルゴリズムを示した。

さらに、本方式を用いれば、パターンマッチ処理が必要なネットワークにしかデータが流れないように、SCUT アークを設定でき、かつ、 $O(N \log N)$ 程度の計算量で、ST-NET が生成可能であることを示した。

謝辞 本研究の場合、およびその方向付けを与えて下さった、(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致します。

参 考 文 献

- 1) Feigenbaum, E. A.: The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering, *Proc. of 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. (1981).
- 4) 田野, 増位: 知識処理モデルに基づく推論および推論制御方式, 第1回人工知能学会全国大会, pp. 139-142 (1987).
- 5) Tano, S. et al.: EUREKA-II: A Programming Tool for Knowledge-based Real Time Control Systems, *Proc. of the Int. Workshop on Artif. Intell. for Industrial Applications*, pp. 370-375 (1988).
- 6) 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, 第33回情報処理全国大会論文集, pp. 1417-1418 (1986).
- 7) 田野, 増位, 坂口, 船橋: 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).
- 8) 田野, 増位: ST-NET アルゴリズム: 双方向推論の高速処理方式, 情報処理学会論文誌, Vol. 29, No. 10, pp. 944-953 (1988).
- 9) Forgy, C.L.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, *Artif. Intell.*, Vol. 19, No. 1, pp.

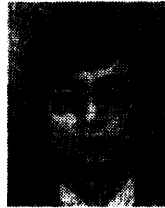
- 17-37 (1982).
- 10) Miranker, D. P.: TREAT: A Better Match Algorithm for AI Production Systems, AAAI-87, pp. 42-47 (1987).
 - 11) 荒屋ほか: プロダクションシステムのための高速パターン照合アルゴリズム, 情報処理学会論文誌, Vol. 28, No. 7, pp. 768-775 (1987).
 - 12) 荒屋ほか: 知識ベースの逐次構造化—Rete ネットワークの逐次構築法, 電子情報通信学会論文誌 D, Vol. J 71-D, No. 6, pp. 1100-1108 (1988).
 - 13) 磯貝: 第二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 (1985).
 - 14) 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. 1568-1569 (1987).
 - 15) 石田, 桑原: プロダクションシステムの高速度化技術, 情報処理, Vol. 29, No. 5, pp. 467-477 (1988).
 - 16) 坂口, 田野, 増位: ST-NET (ルール関連図) 生成のためのパタン間関係解析アルゴリズム, 第 38 回情報処理学会全国大会論文集, pp. 533-534 (1989).

(昭和 63 年 10 月 14 日受付)
(平成 元年 6 月 13 日採録)



田野 俊一 (正会員)

昭和 33 年生. 昭和 50 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, IEEE 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.



大森 勝美 (正会員)

1963 年生. 1985 年宇都宮大学工学部情報工学科卒業. 1987 年同大学院修士課程修了. 同年(株)日立製作所入社, 大みか工場勤務. 知識工学, ファジィ理論に興味をもつ. 電子情報通信学会, 人工知能学会各会員.