ATMSによるRete-Like ネットワークの逐次構築

2F-3

太田 好彦、井上 克巳
(新世代コンピュータ技術開発機構)

1．はじめに

従来、仮説推論システムの例として，（1）がある。これは、ATMS（2）とReteアルゴリズム（3）とを融合しラベル計算とパターン・マッチングをReteネットワーク内で同時に行い推論速度の向上を図っている。しかしながら、これは、Reteネットワークの構築法については論じていない。もともと、仮説推論は、不完全な知識等を取り扱うものでありルールの追加・削除が頻繁に起こると考えられる。したがって、Reteネットワークの構築に逐次性が要求される。

そこで、ルールのインクリメンタル・コンパイル機能を持ち、さらに仮説推論をサポートしたESツールとして、ART（4）があるが、その具体的アルゴリズムは公表されていない。一方、（5）は、Reteネットワークの逐次構築法にReteアルゴリズムを用いるというところに着目した研究である。

本稿では、主に仮説推論システムにおける、独自のRete-Likeネットワークの逐次構築法を提案する。この逐次構造化法は、ATMSに着目しており、（5）とは異なったルール・ベース管理機構が提供されている。

2．逐次構築法とATMS

もともと、ATMSは矛盾成立処理において逐次性がある。また、ATMSはあるデータが成立している世界を環境で表現して管理している。一般に、複数の環境は束というデータ構造になる。この環境束とRete-Likeネットワークの構造類似性に着目したのが本Rete-Likeネットワークの逐次構築法の基本的な考えである。

以下、本研究で用いたATMSについて述べる。

ATMSが管理するデータは、アトム、述語である。述語の場合、変数を含まないものとしている。データに対応してATMS内部では、ノードが生成される。ここに、ノードは、データで一元的に管理されている。すなわち、等しいデータのノードは、だだ一個に限定される。図1に、ノードのデータ構造を示す。

また、環境は図2に示すデータ構造であり、ビット・ベクタで一元的に管理されている。すなわち、等しいビット・ベクタでの環境は、だだ一個に限定している。これを管理するのが、環境表であり、ハッシュ・テーブルとなっている。

他に、Nogood Databaseがあり、これは矛盾する環境を最小（Minimal）で管理している。

Datum：データ
Label：環境の集合（環境へのポインタの集合）
Justifications：理由付けノードAND結合の集合
（ノードへのポインタの集合の集合）
Consequents：Justificationsの逆ポインタ
（ノードへのポインタの集合）
Contradictory：0＝IN，1＝OUT

図1 ノードのスロット

Assumptions：ビット・ベクタ
Nodes：Labelの逆ポインタ（ノードへのポインタの集合）

図2 環境のスロット

3．逐次構築アルゴリズム

まず、先に述べたATMSモジュールを以下のように修正する。ATMSの環境のデータ構造に図3に示すスロットを追加する。次に、環境の組み合わせを生成するオペレーションorに図4に示す操作を追加し、環境束がSuper，Subリンクで表現されるようにする。

次に、このように修正したデータ構造を利用しての構築法について述べる。対象とするルール・ベースは、IF-THEN型であり、IF部には、複数の条件要素とテスト手続きが記述できる。条件要素は、アトム、述語である。述語の場合、変数を含んでもよいとしている。THEN部は、条件要素と同じ構文とする。

まず、rootノードとしてPremiseノードを1個生成しておく。

次に、1入力ノードであるが、各条件要素をDatumとしたAssumptionノードを生成する。このAssumptionノードのLabelにある環境（1入力ノード）とrootノードにある環境とをSuper，Subリンクで結合する。ここで、ATMSのDatumは、命題（変数を含まない述語）のレベルでなければならないが、ルールの条件要素は変数を含む述語となっている。これを扱うために、ここでは変数として扱わずその位置に変数がある述語であると解釈する。先に述べた、ATMSノードのDatum一元性は、Rete-Likeネットワークの構築に用いられる時は、このようなユニフィケーション原理に基付き行われる。これにより、生成されるRete-Likeネットワークは、自然にファクタリングの処理がなされる。しかし、この原理に従えば、1つのルールでユニフィケーションする条件要素が複数ある場合は、Super，Subリンクのループができてしまうので、別のノードとして扱うことにしている。

次に、2入力ノードであるが、条件要素のAND結合をJustificationにもつテスト・ノードを生成すればよい。そのテスト・ノードの生成過程で生成される中間的な環境が2入力ノードとなる。このとき、同時にS

Incremental Building of Rete-Like Network by ATMS
Yoshihiko Ohta, Katsumi Inoue
(Institute for New Generation Computer Technology)

Ｓｕｐｅｒ：上位環境へのポインタの集合
Ｓｕｂ：下位環境へのポインタの集合
Ｗｍ：分散ワーキング・メモリ（分散ＷＭ），ＡＴＭＳノードの集合

図３　環境の追加スロット

環境Ｅ１のスロットＳｕｐｅｒに環境Ｅ３へのポインタを追加する。
環境Ｅ２のスロットＳｕｐｅｒに環境Ｅ３へのポインタを追加する。
環境Ｅ３のスロットＳｕｂに環境Ｅ１へのポインタを追加する。
環境Ｅ３のスロットＳｕｂに環境Ｅ２へのポインタを追加する。

図４　ｏｒ（Ｅ１，Ｅ２，Ｅ３）のときの追加操作

ｕｐｅｒ，Ｓｕｂリンクも張られる。

以下には、ルールの逐次的な追加や削除のアルゴリズムについて示す。追加する場合は、新規追加と全く同様である。ＡＴＭＳ　は、以前生成したＤａｔｕｍのノードは、再び生成されないように管理されている。これで、１入力ノードの共有がなされる。また、環境表は以前生成した環境が再び生成しないように管理しているので２入力ノードの共有がなされる。ルールの逐次的な削除については、削除する条件要素またはそれらの組み合わせを指定する。指定された条件要素またはそれらの組み合わせを含むルールは全て削除される。システムは、Ａｓｓｕｍｐｔｉｏｎとしてのこれらのノードまたはそれらの組み合わせのＬａｂｅｌを計算し矛盾な環境とする。すると、これを含む環境が矛盾となり、Ｒｅｔｅ－Ｌｉｋｅネットワークから全て削除される。このため、矛盾となる環境を全てのノードのＬａｂｅｌから削除するときに、それとＳｕｐｅｒ，Ｓｕｂリンクで結合されている環境のＳｕｐｅｒ，Ｓｕｂリンクから、その削除する環境へのリンクを切る操作を行う。今までに矛盾となった環境は、Ｎｏｇｏｏｄ　Ｄａｔａｂａｓｅにより、最小（Ｍｉｎｉｍａｌ）で管理されており、以後矛盾となるような条件要素またはそれらの組み合わせを含むルールは追加されない。

４．適用例

本方法の適用例を示す。等しい容量のタンクａ，ｂ，ｃ，ｄに液体が入っている。タンクａ，ｂ，ｃに入っている液体を混合しその濃度が３より大きければタンクｐに入れる。また、タンクａ，ｂ，ｄに入っている液体を混合しその濃度が２より大きければタンクｑに入れる。この問題のルール・セット例を図５に示す。また、本方法によるＲｅｔｅ－Ｌｉｋｅネットワークの例を図６に示す。図において、ａ（Ａ）とｂ（Ａ）の組み合わせは、ｒ１およびｒ２で共有されているので、２入力ノード［３］が共有されている。この例でタンクｄがもう使われなくなったとすると、条件要素ｄ（Ａ）を含むルールを削除すべきである。本方法によるルールの削除は、このｄ（Ａ）を指定する。すると、１入力ノード［８］、２入力ノード［１１］およびテスト・ノード＃ｒ２が削除される。以後、条件要素ｄ（Ａ）を含むルールは、ネットワークに反映されない。

５．まとめ

以上、ＡＴＭＳに着目した　Ｒｅｔｅ－Ｌｉｋｅネットワークの逐次構築法について述べた。本方法は、ルール削除が特徴的である。つまり、ルール単位で行うのではなく、ルールの条件要素またはそれらの組み合せを指定する。そ

```
r1 : : a (X), b (Y), c (Z),
        {W= (X+Y+Z) ／3．0, W>3．0}
－>
        p (X, Y, Z, W).

r2 : : a (X), b (Y), d (Z),
        {W= (X+Y+Z) ／3．0, W>2．0}
－>
        q (X, Y, Z, W).
```

図５　ルール・セットの例

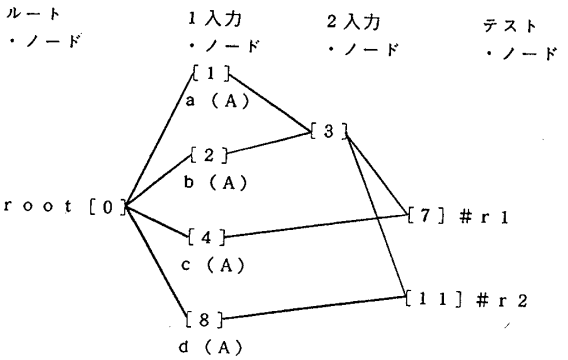

図６　ネットワークの例
（図５のルール・セット．Ａは、変数を意味する。
［］は、ビット・ベクタ十進表現した環境を示す。）

れによって、それを条件に含むルールが全て削除される。しかも、これらの条件要素を削除した効果は残り、これらの条件要素を含むルールを追加しても以後ネットワークには反映されないという、いわば削除ルールに関する学習効果がある。この点に関して本方法は、ルール・ベース・管理とルール・コンパイルを同時に行う手法である。

謝辞

本研究の機会を与えて下さり常に御指導頂いたＩＣＯＴ淵所長、第五研究室藤井室長に深く感謝致します。

参考文献

（１）飯島　他：”仮説ネットワークを用いた仮説推論器”、人工知能学会研究会研究会資料ＳＩＧ－ＦＡＩ－８７０１－２，ｐｐ．７－１４（１９８７）
（２）ｄｅ　Ｋｌｅｅｒ，　Ｊ．：”Ａｎ　Ａｓｓｕｍｐｔｉｏｎ－ｂａｓｅｄ　ＴＭＳ”，Ａｒｔｉｆｉｃｉａｌ　Ｉｎｔｅｌｌｉｇｅｎｃｅ　２８，　ｐｐ．１２７－１６２（１９８６）
（３）Ｃ．Ｌ．Ｆｏｒｇｙ：”Ｒｅｔｅ：Ａ　ｆａｓｔ　ａｌｇｏｒｉｔｈｍ　ｆｏｒ　ｍａｎｙ　ｐａｔｔｅｒｎ／ｍａｎｙ　ｏｂｊｅｃｔ　ｐａｔｔｅｒｎ　ｍａｔｃｈ　ｐｒｏｂｌｅｍ”，Ａｒｔｉｆｉｃｉａｌ　Ｉｎｔｅｌｌｉｇｅｎｃｅ１９ｐｐ．１７－３７（１９８２）
（４）Ｂ．Ｃｌａｙｔｏｎ：”ＡＲＴ　Ｐｒｏｇｒａｍｍｉｎｇ　Ｔｕｔｏｒｉａｌ”，Ｉｎｆｅｒｅｎｃｅ　Ｃｏｒｐｏｒａｔｉｏｎ（１９８５）
（５）荒屋　他：”知識ベース逐次構築法―Ｒｅｔｅネットワークの逐次構築法―”、信学論Ｄ　Ｖｏｌ Ｊ７１－Ｄ　Ｎｏ．６　ｐｐ．１１００－１１０８　（１９８８）