6J-8

仮説の選択が可能なＡＴＭＳ

山之内　徹　　渡辺　正信
日本電気㈱　Ｃ＆Ｃシステム研究所

１．はじめに

非単調な推論を可能とする一般的な推論の枠組みとして、ＴＭＳ[1]やＡＴＭＳ[2]が提案されている。ＴＭＳは、常に１つの環境（仮説の集合）に沿って推論を行い、矛盾が発見されると、バックトラックして、別の環境に沿った推論を再開する。したがって、１つの解を求める場合には効率が良いが、いくつかの可能性（環境で表現される）の中から最適な解を選択することは、そのままでは困難である。

一方ＡＴＭＳは、複数の環境を同時に管理し、それぞれの環境に沿った推論を並行して進める。したがって、全ての解を求める必要がある場合には効果的だが、１つの解だけに興味がある場合や、多数の解のうち、いくつかの解が非常に大きな計算を必要とする場合には、非常に効率が悪くなる。

本論文では、ＡＴＭＳに仮説の中断、復活機能を追加し、推論を並行して行う環境を、ユーザまたは問題解決機構が制御することを可能にした、ＳＡＴＭＳ(Selective Assumption based TMS) について報告する。

２．ＳＡＴＭＳの構成

2.1 ＳＡＴＭＳを含む問題解決機構

図１にＳＡＴＭＳを含む問題解決機構の構成を示す。制約伝播やルールベース推論などからなる問題解決部は、ワーキングメモリにたいし、事実データの登録(assert)、仮説データの登録(assume)、データの導出(deduce)などを直接行う。一方、ワーキングメモリは問題解決部に対し、その内容を直接見せることはなく、ＳＡＴＭＳを通して提示する。これにより、問題解決部は、事実データと現在信じられている仮説に依存するデータだけを参照することができる。

問題解決部がワーキングメモリを更新したとき、他のデータとの間に矛盾を起こすことがある。この場合には、矛盾を引き起こした１つ以上のデータが依存している全ての仮説の集合を nogood-set に加え、これらの仮説に依存している推論結果を無効とする。以後、無効となった推論結果は、問題解決部からは見えなくなる。問題解決部が既に登録してある仮説の中断(suspend)、復活(resume)を決めた場合、ＳＡＴＭＳに対して指示を出す。

2.2 仮説の中断と復活

ＳＡＴＭＳは仮説の中断(suspend)を指示されると、その仮説をsuspended-setに追加し、compound nogood-setをsuspended-setとnogood-setの和集合の最小表現として再計算する（compound nogood-setの計算は、suspended-set、nogood-setのいずれかが変更されるたびに必ず行われる）。またＳＡＴＭＳは、現在どの仮説を信じているかのチェックをcompound nogood-setを利用して行うため、中断されている仮説及びその仮説に基づく推論結果は矛盾を導く仮説と同様、問題解決部からは見えなくなる。したがって、以後の推論においては、その仮説が復活されるまで、その仮説が登録されなかった場合と同一の結果をもたらす。

ＳＡＴＭＳは、中断されていた仮説の復活(resume)を指示されると、その仮説をsuspended-setから削除し、compound nogood-setを再計算する。さらに、復活された仮説およびその仮説から既に得られていた推論結果のデータを問題解決部に通知し、これらのデータに基づく問題解決が再開されるきっかけを作る。これにより、仮説が中断されていたことによる、その環境に沿った問題解決の遅れを取り戻すことが可能となる。特に問題解決部が制約伝播によって記述されている場合、推論の順序が最終的な推論結果と無関係なため、一度中断された仮説がresumeによって復活されれば、その仮説が一度も中断されなかった場合と同一の推論結果を得ることができる。

2.3 いくつかの環境に沿った問題解決の並行実行

ＳＡＴＭＳは、基本的にＡＴＭＳに仮説の中断、復活の機能を付加したものであるため、問題解決部が仮説の中断を行わなければ、ＡＴＭＳと同一の結果をもたらす。つまり、全ての環境に沿った問題解決を並行して実行す

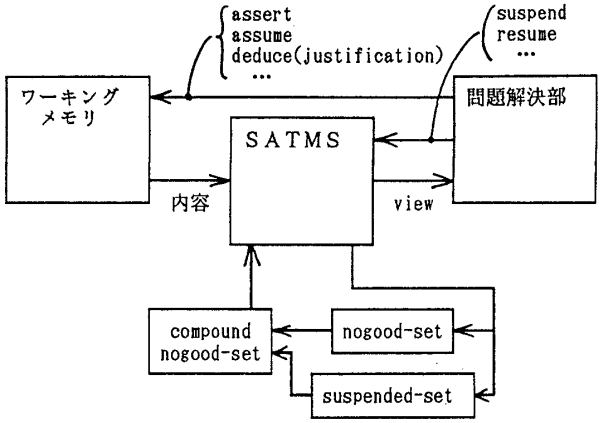

図１　ＳＡＴＭＳを含む問題解決機構の構成図

SATMS：Selective Assumption based Truth Maintenance System
Toru YAMANOUCHI and Masanobu WATANABE
NEC Corp. C&C Systems Research Laboratories

る。一方、複数の環境が互いに矛盾していることが判明した場合、問題解決部が、１つの環境だけを残し、他を中断することによってＴＭＳをシミュレートすることができる。推論が進んで、残された環境が別の矛盾を導き、その環境が削除の対象となった場合、中断されていた仮説のうち、１つを選択して復活し、推論を再開することで、依存に基づいたバックトラックが行われる。

さらにこの枠組みは、ユーザまたは問題解決部が、ヒューリスティクスにより有効と判断した、限られた環境に沿った推論だけを並行して行うことが可能である。したがって、ヒューリスティクスが妥当であった場合にはＡＴＭＳよりも効率良く解を求めることができる。また、ヒューリスティクスによる判断が誤っていることが後で判明した場合には、無効と判断され、中断されていた仮説を復活させることにより、別な環境に沿った推論を再開することができる。

３．論理回路の最適化問題の場合

次に簡単な論理回路の最適設計問題で、ＳＡＴＭＳがどのように利用されるかを示す。式１は１つの大きな論理回路の一部分の機能を示す論理式、R1,R2 は、論理回路を最適化するための論理式変換規則である。

```
OUT1 = (E + A)*C + A*D                (式１)

OUT1 = E*C + A*C + A*D                (式２)

OUT1 = E*C + A*(C + D)                (式３)

[X * (Y + Z)]   ==> assume [X * Y + X * Z]  (R1)
[X * Y + X * Z] ==> assume [X * (Y + Z)]    (R2)
```

式１の論理式に R1 を適用すると式２に、式２の Aに着目して R2 を適用すると、式３のようになる。ここで、式１と式３は４つのゲート（ANDとOR）で実現できるが、式２は５つのゲートが必要となる。したがって、この小さな論理回路だけを考えると、式１または式３が最適解となるが、この論理式を含む大きな回路全体でみると、必ずしもローカルな最適解が実際の最適解になるとは限らない。同じ大きな回路の中に式４の論理式で表される仕様をもつ部分 part2 が別に存在したとする。この論理式も、変換規則 R1,R2 で式５、式６のように変換され、式４、式６が４つのゲートで、式５が５つのゲートで実現されることがわかる。

```
OUT2 = (E + B)*C + B*D                (式４)

OUT2 = E*C + B*C + B*D                (式５)

OUT2 = E*C + B*(C + D)                (式６)
```

ＳＡＴＭＳを利用した問題解決では、ＳＡＴＭＳに対し、式１～式６のいずれも仮説として登録するが、式２と式５は最適解である可能性が低いものとの判断から、仮説の中断を行う。これにより、ＳＡＴＭＳは、式１と式４、式１と式６、式３と式４、式３と式６のそれぞれ２つの仮説を信じる４つの環境に沿った問題解決を並行して行う。part1 と part2 におけるゲートの共通化から、式３と式６に基づく環境に沿った問題解決が、図２のような６個のゲートによる論理回路を最適解として与える。

これを、従来のＴＭＳの枠組みで解決しようとすると、式１、式４とも、R1 による変換後の論理式、式２、式５よりも少ないゲート数となるため、一般には、式１、式４に基づく解として、８個のゲートによる論理回路を導出することになる。また、従来のＡＴＭＳでは、仮説の中断機能がないため、仮説の組合せは全部で９通りとなり、９個の解を同時に求めることになる。最適な解としては、図２と同じものが得られるが、処理量は本方式と比較して２倍以上となる。

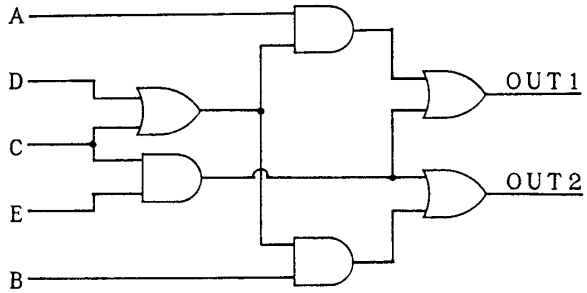

図２　最適解として得られた論理回路

４．まとめ

ＡＴＭＳに仮説の中断、復活機能を付加し、並行して問題解決を実行する環境を、ユーザや問題解決機構が制御可能な、ＳＡＴＭＳについて述べた。同様なシステムに、ＴＭＳの依存型バックトラックにＡＴＭＳの”より一般的な環境に基づく推論を先に行う”性質を組み込んだ deKleer[3]や、環境に対して重みを導入し、重みの高いものの順に推論を行う、劉[4]があるが、どちらも、推論を並行して行う環境を、状況に応じて動的に制御できる本システムとは異なる。

現在、プロトタイプの実現を終了し、いくつかの例題で本方式の有効性を確認している。

今後は、①値の継承を持つようなフレーム型のワーキングメモリに対してＳＡＴＭＳを適用すること、②制約伝播、ルールベース、フレーム上のデモンなどの複合環境（ＡＩツールＣＬ[5]など）による問題解決機構との結合、などを検討して行きたい。

最後に、本研究の機会を頂いた、Ｃ＆Ｃシステム研究所大野直哉部長、小池誠彦課長に感謝致します。

参考文献

[1] McAllester,D., An outlook on truth maintenance, MIT AI Memo No.551(1980).

[2] deKleer, J., An assumption-based truth maintenance system, Artificial Intelligence 28(1986) 127-162.

[3] deKleer, J. and Williams, B., Back to backtracking: controlling the ATMS, AAAI 1986, PP.910-917.

[4] 劉学敏、他，ＴＭＳの統合的自然言語理解への応用に関する考察,情処:知識工学と人工知能研究会,87-AI-53-1(1987)

[5] 渡辺正信、他，ＣＬにおけるルール指向プログラミング,情処:知識工学と人工知能研究会,86-AI-46-3(1986)