ＡＴＭＳの並列化に関する一提案

6C－3

°原田　拓　　溝口文雄
（東京理科大学　理工学部）

1．はじめに

意味，真理値などに関して曖昧性を伴った知識を扱う場合，一般に膨大な計算量を伴う．ＡＴＭＳ[1][2][3]はこの曖昧性のうち不完全性に焦点を当て，この不完全な知識を仮説として扱い，無矛盾な仮説の組合せを求めることによって，その真理値を維持しようとするものである．ところが，この場合，この仮説として扱われた知識を組み合わせることによる組合せ爆発が生じ，探索空間の爆発的増加のために，実行効率の低下という問題点が生じる．そこで，本稿では，ＡＴＭＳを並列化することによって実行効率を向上させるための方法について，一つの提案を行うものである[4][5][6]．

なお，試作したシステムは，計算機としてSun3/260，言語として並列論理型言語ＧＨＣを用いている．

2．並列化に対する指針

まず，ＡＴＭＳを並列化するために，以下の点に着目する．

①プロセス・ダイナミックス
②データ依存関係
③制約充足

このうち，本論文では①と②について述べる．

①のプロセス・ダイナミックスは，ある処理単位をＧＨＣのプロセスとして表現し，このプロセス間においてメッセージを通信し合うことによって，全体の処理を進めていくものである．

②のデータ依存関係は，ＡＴＭＳの保持するＡＴＭＳノードと呼ばれるデータ構造におけるデータ間の結合関係に着目して，ＡＮＤ並列やストリーム並列などを有効的に利用することができるというものである．

3．データ依存関係

Problem SolverからＡＴＭＳに対して，新しく導出されたデータとその導出過程における依存関係をJustificationとして渡す．このJustificationは一般に

$$X_1, X_2, \ldots, X_i, \ldots \rightarrow \mathrm{Datum}$$

という形式をしており，この論理的結合関係は，

$$X_1 \wedge X_2 \wedge \ldots \wedge X_i \wedge \ldots \vdash \mathrm{Datum}$$

となる．また，同じConsequent Nodeを持つJustificationは互いにＯＲ関係にある．従って，このようなデータ依存関係に着目することによって，効率的な並列処理を行うことができる．

このようなデータ依存関係を有効的に活用した並列処理を行うために，試作したシステムでは図1に示すようなネットワークを構築する．つまり，同一のデータをネットワーク上の同じノードとして表現する．もし，Justificationごとにネットワーク上のノードとして表現すれば，同一データのラベルが異なるノードにおいて異なる値で具体化されることがあり，この整合性を保つための処理に大きな計算量がかかってしまうことになる．また，このネットワークのノードは各々ＧＨＣのプロセスとして表現されている．これは並列化に対する指針①のプロセス・ダイナミックスを用いたものである．

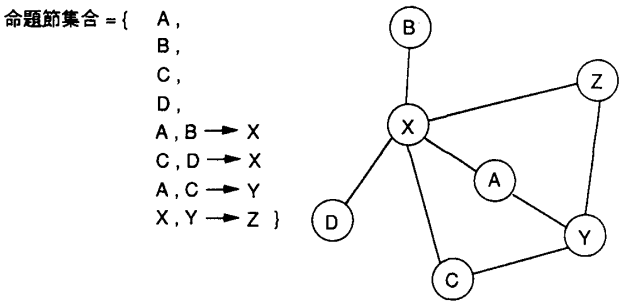

図1．Justificationに基づくネットワーク

このようなネットワーク上では，例えば仮説Ａが他の仮説から導かれるDerived Nodeとしても扱われることになった場合には，その影響による再ラベル計算が，並列伝播(Parallel Propagation)を利用して，Derived NodeＸとＹにおいて各々独立に行われることになる．

4．ネットワーク上の処理

4.1 ラベル計算

ＡＴＭＳでは以下のようなＡＴＭＳノードと呼ばれるデータ構造でその依存関係を表現している．

< Datum , Label , Justification >

Datumはデータであり，LabelはDatumが依存する仮説の集合，JustificationはDatumが成立する理由付けである．これに対して，試作したシステムでは，この情報をネットワーク上のノードに持たせ，このうちJust

---

A Proposal for Parallel ATMS
Taku HARADA , Fumio MIZOGUCHI
Science Univ. of Tokyo

ification部分はAntecedent NodeとConsequent Node間におけるネットワークチャネルとして表現している．即ち，あるノードは，そのノードをConsequent NodeとするJustificationの各Antecedent Nodeに対するチャネルと，さらにこれに加えて，このノードをAntecedent Nodeに含む全てのJustificationのConsequent Nodeに対するチャネルを保持している．

ネットワーク上のノードは，そのラベルが変化した場合，このデータをAntecedent Nodeに持つノードに対して，再ラベル計算を行うように指示する．このメッセージを受け取ったノードは，各Antecedent Nodeに対してストリーム並列を用いて環境（ラベル）を探索する．そして，探索した環境（ラベル）を用いてラベル計算を行うが，この際，ラベルの満たすべき性質のうちの極小性を保つように注意する．この結果，新しい環境が生成されれば，このノードをAntecedent Nodeに含むConsequent Nodeに対して，同様のメッセージを送る．もし，このConsequent Nodeか複数あれば，各々のノードにおける処理は独立に行われることになる．例えば，図1のデータXとYがこれに相当する．

4.2 nogoodの扱い

Problem SolverからJustificationが与えられるたびに，極小nogoodをプールしたモジュールに対して，そのJustificationのConsequent Nodeが無矛盾かどうかを確かめる．しかし，この結果にかかわらず，このJustificationに基づいたネットワークの更新は必ず行う．これは，この時点ではnogoodとして扱われていたデータが，後で無矛盾なものとして扱われる可能性があり，この場合そのデータをAntecedent Nodeに含むJustificationのConsequent Nodeも無矛盾なものとして扱われる．従って，あるデータの真理値が変化するたびにネットワークの生成・削除を行っていたのでは，これに対する計算量が膨大なものとなってしまうのである．よって，あるデータが一度nogoodとして扱われた場合でも，ネットワーク上のノードからは削除されない．言い換えれば，ネットワーク上のノード数は，Problem Solverから与えられるデータ数に対して，単調に増加するものである．

4.3 終了判定

Problem Solverと本システムとの間では，JustificationやBeliefなどをストリームを介して通信し合うが，このストリームが閉じられると，本システムに対してProblem Solverからはいかなるデータもジャスティファイされないので，この時点でネットワーク上の通信ストリームを閉じ，各プロセスを終了させる処理を開始する．

ここで注意しなければならない点は，ネットワーク上における全てのメッセージに対する処理を終了させてから通信ストリームを閉じなければならないということである．そうしなければ，通信ストリームを閉じた後で，新たなメッセージを受け取り，それによって新たなnogoodが生成される可能性があるからである．そこで，本システムでは，ショートサーキット・テクニックを用いて通信ストリームの終了時期を判定している．具体的には，次に示すようにショートサーキットのためのフラッグ変数(Left,Right)とともに，それが通過したネットワーク上のノードの履歴(History)をメッセージとして送る．

< Left , Right , History >

これを受け取ったノードは，その履歴の中に自分自身があればフラッグ変数を短絡(Left=Right)させ，そうでなければフラッグ変数を隣接するノードに分散させる．1本の通信ストリームにおいては，後から送られたメッセージが先に送られたメッセージを追い越すことはないので，このようにすることによって，真理値維持に関する全てのメッセージに対する処理を終了させてから，ネットワーク上の通信ストリームを閉じることができる．ただし，この方法が最も効率のよい方法である保証はなく，さらに検討の余地はある．

5．まとめ

ＡＴＭＳの並列化のための方法に関して1つの提案を行った．

試作したシステムでは，環境数をnとすると，nogoodがない場合ではサイクル数はおよそO(n)となるという結果を得ている．また，nogoodがある場合，それを命題節集合の最初にジャスティファイした場合は特に特性はなく，中間にジャスティファイした場合ではnogoodがない場合と同様におよそO(n)となり，最後にジャスティファイした場合は，サイクル数は環境数にあまり影響を受けないという結果を得ている．

また，試作したシステムを用いて静的な制約充足問題を解く場合，命題節集合において，制約をnogoodとして宣言し，制約に含まれる変数領域を仮説として宣言した場合では，制約を変数領域よりも先にジャスティファイした方が効率よく解けるという解析結果も得ている．

参考文献

[1]de Kleer J.:An Assumption-based TMS,Artificial Intelligence 28 (1986) pp.127-162

[2]de Kleer J.:A GENERAL LABELING ARGORITHM FOR ASSUMPTION-BASED TRUTH MAINTENANCE,Proc. of AAAI-88 (1988) pp.188-192

[3]de Kleer J. and Williams B.C.:Diagnosing Multiple Faults,Artificial Intelligence 32 (1987) pp.97-130

[4]原田　拓，溝口文雄：ＡＴＭＳに基づく真理維持の並列処理方式，日本ソフトウェア科学会第6回大会（1989) pp.73-76

[5]Michael Dixon and de Kleer J.:Massively Parallel Assumption-based Truth Maintenance,Proc. of AAAI-88 (1988) pp.199-204

[6]奥乃　博：動的特性解析によるＡＴＭＳの並列処理の検討，人工知能学会第3回全国大会（1989) pp.171-174