

2F-2

データ依存関係に基づく
無矛盾性管理の並列処理方式

原田 拓 溝口 文雄
(東京理科大学 理工学部)

1. はじめに

我々を取りまく知識には、曖昧性を伴ったものが多い。この曖昧性を大きく分類すると、不確実性、不完全性、多義性に分けることができる。本論文では、これら曖昧性のうち知識の不完全性に焦点を当て、知識全体の無矛盾性を効率的に管理するために、並列処理を行う方法について述べる。

不完全な知識に関する論理体系としてDefault Reasoning, Nonmonotonic Logic, Circumscription等を挙げることができる。一方、このような知識に関する管理のためのシステムのアプローチとしてde KleerのATMS (Assumption-based Truth Maintenance System)[1][2][3]がよく知られている。ここではこのATMSを基本概念として採用しており、並列ATMSの実現方法を提案するという意義もある[5][6]。

なお、使用計算機としてSun3/260ワークステーションを用い、システムはQuintus Prolog上のFGHCを用いてインプリメントされている。

2. 全体の構成

本論文で述べるシステム(Parallel Consistency Maintenance System:PCMS)はその基本的概念としてATMSを取り入れている。ATMSはProblem Solverに対して互いに情報を通信しあうことによって動作するインタラクティブな関係にある。PCMSもATMSと同様に、Problem Solverにおける各プロセスに対して各々割り当てられる。Problem Solverにおける各プロセスは互いに通信し合うことによって全体としての解を求める。具体的には、Problem SolverからPCMSへはオブジェクトの推論過程において導出された新しい状態に関する情報が与えられ、逆にPCMSからProblem Solverへは無矛盾性を調べられた状態に関する情報が与えられる。これを示したのが図1である。

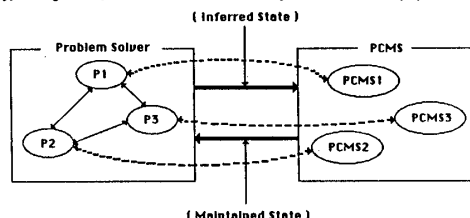


図1. 全体の構成

3. データ構造

ATMSでは次の形式でデータを保存しており、この形式で保存されたデータをATMSノードと呼ぶ。

$\langle \text{State}, \text{Environment}, \text{Justification} \rangle$
Stateは対象とするオブジェクトの状態(データ)であり、EnvironmentはStateが成立する環境(仮説のリスト)を表現している。またJustificationはStateの理由付けの集合を表現している。

PCMSでは現在のところEnvironmentとJustificationの両方の性質を一部考慮したものをEnvironmentとしており、次のようなデータ構造を採用している。

$\langle \text{State}, \text{Environment} \rangle$

1つのPCMSは一般的にこのデータ構造を同時に複数個保持する。

Node : $U_i \{ \langle \text{State}_i, \text{Env}_i \rangle \mid 1 \leq i \leq v \}$

State_i : $U_j \{ \text{Data}_{ij} \mid 1 \leq j \leq r \}$

Env_i : $U_m \{ \text{Env}_{im} \mid 1 \leq m \leq s \}$

Env_{im} : $U_n \{ \text{Env}_{imn} \mid 1 \leq n \leq t \}$

ここで、ATMSと同様にState_iは、対象とするオブジェクトの状態を表現している。また、Env_iはその状態(State_i)が導出されるまでの過程(依存関係)を表現しており、これを環境と呼ぶ。一般に、1つの状態に対して環境は複数個存在しており、これは同一状態が異なる依存関係によって導出されたことを意味している。なお、この環境の集合をラベルと呼ぶ。

4. PCMSの並列性と一般性

4.1 並列性

3で述べたノード集合における各要素の関係を詳しく記述すると以下のように表現することができる。

Node : $\{ \langle \text{State}_1, \text{Env}_1 \rangle \vee \langle \text{State}_2, \text{Env}_2 \rangle \vee \dots \vee \langle \text{State}_n, \text{Env}_n \rangle \}$

Env_i : $\{ \text{Env}_{i1} \vee \text{Env}_{i2} \vee \dots \vee \text{Env}_{ik} \}$

Env_{im} : $\{ \text{Env}_{im1} \wedge \text{Env}_{im2} \wedge \dots \wedge \text{Env}_{imt} \}$

すなわち、異なるノードは互いに選言関係にあり、また、ラベルに含まれる環境同士も選言関係にある。これに対して、1つの環境に含まれる各要素同士は連言関係にある。

従って、Problem Solverによって得られた新しい状態に関する情報を用いて、それまで保持していたデー

タとの無矛盾性を調べる場合に、各ノードに対してはOR並列に調べることができる。またこの際、各ノードにおけるラベルに含まれる環境に対してはOR並列に調べることができる。これに対して、各環境に含まれる各要素に対してはAND並列に調べることになる。

また、ノードの変更の他に環境の束の探索も並列に処理することができる。この環境束の探索は、組合せ問題として捉えることができる。最下位レベルの環境は空集合 $\{\}$ であり、最上位レベルの環境は全ての環境の集合となる。もちろん、この環境のレベルを逆転させても意味的には変わりはない。上位レベルの環境に対して下位レベルの環境は連言関係にあり、逆に、下位レベルの環境に対して上位レベルの環境は選言関係にあると言える。これを示したのが図2である。

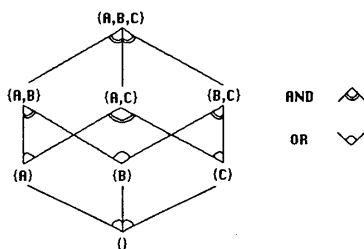


図2. 環境束

4.2 一般性

PCMSが保持するオブジェクトのデータ構造は3で述べた通りであるが、PCMSはその対象領域として制約充足問題、分散協調問題、故障診断等を考えているため、ある1つのノードの保持するオブジェクトの状態には様々な形態が考えられる。つまり、一般には複数個の要素が組み合わされて初めてある1つの状態を表現できるわけであるが、この要素数は任意であり、Problem Solverに依存したものである。例えば、回路の故障診断を対象領域とした場合[4]、回路におけるデバイスの各端子に対して各々PCMSを割り当てるのが自然であり、この場合、状態を表わす要素数は1個である。これを示したのが図3である。

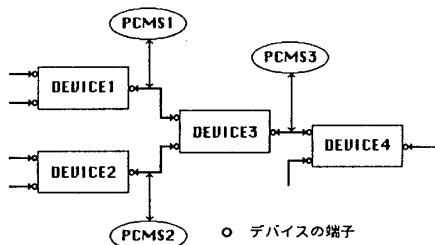


図3. デバイスとPCMS

これに対して、状態を表わす要素数が複数個ある場合の例としてN-Queen問題を挙げることができる[7]。4-Queen問題を2つに分割して解く場合、1つのPCMSが保持する状態は $\{\text{position}(A,B), \text{position}(C,D)\}$ と表現されることになる。この任意の要素数を一般的に

扱うためにオブジェクトの状態をリスト構造で表現している。

また、ATMS自身は問題解決戦略を持たないが、PCMSでは問題に含まれる制約を用いてデータのnogoodを判定する。しかし、このnogood判定対象はその問題に依存したものである。例えば回路の故障診断を対象領域とした場合では、異なるノード間の状態同士を比較してnogoodの判定を行なうが、N-Queen問題では1つのノードにおける状態においてnogoodを判定する場合もある。これに対しては、Problem Solverにおいてある一定の形式でnogood判定の対象およびその基準を定義すれば、PCMSがそれを認識して処理するようになっている。これも状態をリスト構造として扱っているために容易に扱うことができる。

5. おわりに

データ依存関係に基づいて対象領域全体の無矛盾性を管理する際の並列処理方式について述べた。これはATMSの並列処理可能性を示唆していることにもなる。具体的には、データ構造における依存関係および環境束の連言、選言関係に注目することにより各々AND並列、OR並列を行えることを述べた。また、システムの一般性を保つために、対象オブジェクトの状態をリスト構造として表現した。実際に、6-Queen問題において並列処理の有効性を調べた結果、サイクル数で比較して約7割の実行効率の向上を観測している。

今後は命題論理の演繹体系の並列性の考察、および大規模問題への適用による並列処理の有効性をさらに検討して行きたい。

参考文献

- [1]de Kleer J.:An Assumption-based TMS,Artificial Intelligence 28 (1986) pp.127-162
- [2]de Kleer J.:Problem Solving with the ATMS,Artificial Intelligence 28 (1986) pp.197-224
- [3]de Kleer J.:A GENERAL LABELING ALGORITHM FOR ASSUMPTION-BASED TRUTH MAINTENANCE,Proc. of AAAI-88 (1988) pp.188-192
- [4]de Kleer J. and Williams B.C.:Diagnosing Multiple Faults,Artificial Intelligence 32 (1987) pp.97-130
- [5]原田 拓, 溝口文雄: GHCによる並列問題解決, 日本ソフトウェア科学会第5回大会論文集 (1988) pp.53-56
- [6]Michael Dixon and de Kleer J.:Massively Parallel Assumption-based Truth Maintenance,Proc. of AAAI-88 (1988) pp.199-204
- [7]横尾 真, 石田 享: 分散協調問題解決におけるATMSの利用, 人工知能学会第2回全国大会論文集 (1988) pp.141-144