

仮説に基づいた組合せ問題解決機構

加藤 文之

松下電器産業株式会社 東京研究所

1. はじめに

次世代の知識情報処理システム構築のために、各種の高次推論技術の研究開発が行われている。その中でも仮説推論は、適合する知識や真偽が不明の知識のように不完全な知識を取り扱う事ができる推論の枠組みである。

この仮説推論を実現するために必要な要素技術の一つとして、知識の論理的な一貫性を保持するための知識管理の技術がある。知識管理機構の代表的なものとしてはTMS[1]やATMS[2]がある。このうちATMSでは、仮説の組合せに基づいて複数の環境(仮説の集合)が同時に扱われ、論理的な一貫性はそれぞれの環境において管理されている。

ここで設計型問題の事を考えてみると、設計型問題は基本的には組合せ問題であり、矛盾のない解候補の組合せを効率的に生成できる事が重要である。また複数の解候補の中から最適な解を容易に選択できる事も考慮すべき事項である。これに対してATMSは、前述のような枠組みを持っており、このような問題の解決に適した枠組みであると考えられる。しかしATMSをそのまま問題解決に適用するには、いくつかの問題が存在する。例えば仮説の数が非常に多い問題の場合、組合せ的爆発が発生する危険性がある。またすべての解を得る場合には適しているが、複数の解の中のどれでも良いから一つだけ求めれば良いような場合には、逆に無駄な処理が増加してしまう。

本論文では、ATMSを利用する上で上記のような欠点を補い、組合せ問題の効率的な解決が行えるようにする事を旨とした問題解決機構HYPROS(Hypothesis-based problem solver)について報告する。

2. 仮説管理機構

HYPROSにおける仮説管理機構は、基本的には前述のようにATMSである。ただしここでは対象を組合せ問題に限定して考える事にした。そして組合せ処理を効率的に実行する事を目的として、理由付け等にある程度の制限を加えたMATMS(Micro ATMS)を用いる事にした。MATMSでは、仮説の組合せ処理によって生成された環境はテーブルで管理を行い、理由付けを伴う導出ノードでは管理していない。またMATMSはビットベクタやハッシュテーブルを用いる事により、仮説管理処理の高速化を試みている。

MATMSには以下のような機能がある。

- (1)与えられたデータを持つ仮説を登録する。

- (2)仮説の集合である環境を生成し登録する。

- (3)環境に導出データが伴う場合には、環境をラベルとして持つデータノードを生成し登録する。

- (4)矛盾を登録する。

- (5)登録された矛盾に基づいて、仮説や環境の無矛盾性を検査する。

3. 問題解決機構

図1にMATMSを含むHYPROSの構成を示す。

MATMSを含むHYPROSは、ESPで記述されている。ESPで記述されたATMSを利用したシステムとしてはASTRON[3]等がある。HYPROSもASTRONと同様に、オブジェクト指向の特徴を取り入れ、仮説・環境・ノード・矛盾等をオブジェクトで表現し利用している。以下にHYPROSの問題解決機構の構成と、その特徴を述べる。

3.1 構成

仮説推論は、仮説の生成・検証すなわちgenerate and testを基本的な枠組みとしている。HYPROSの問題解決機構はこのgenerate and testの枠組みに基づいて構成されている。具体的には以下のようなモジュールからなる。

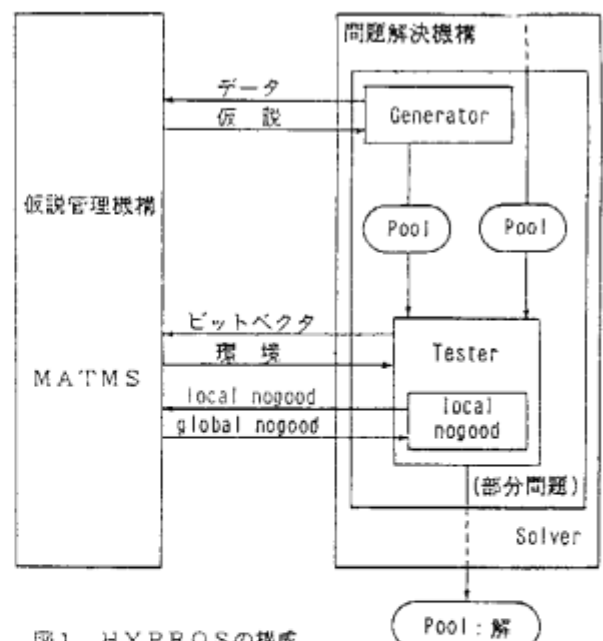


図1 HYPROSの構成

A hypothesis-based problem solver for combinatorial problem

Fumiyuki KATO

Tokyo Research Laboratory, Matsushita Electric Industrial Co., Ltd.

(1) Pool

- 同時に処理を行う対象となる複数の仮説や環境を保持・管理する。

(2) Generator

- 与えられたデータの集合に基づき、M A T M S に対して仮説の登録を要求する。そして得られた仮説を Pool に登録する。

(3) Tester

- 与えられたデータや仮説・環境等に基づいて、当該 Tester のみで有効な local nogood を生成する。
- 与えられた Pool に保存されている仮説・環境を組合せて、新たな環境を生成する。新たな環境の生成に際しては、local nogood に基づいて無矛盾性の検査を行い、矛盾を含まない環境のみを新たな Pool に登録する。
- 終了条件に基づいた検査を行い、その条件を満たすものを解テーブルに登録する。

(4) Solver

- 上記モジュールの管理・操作を行う。
- 特に Tester に対しては、以下の処理を行う。
 - 各 Tester が保存している local nogood を収集し、それらを矛盾の候補として M A T M S に与える。
 - M A T M S において複数の Tester の local nogood から生成された nogood (minimal な形式で保存されている) を global nogood と呼ぶ。
 - 各 Tester が保存している local nogood を、global nogood に基づいて更新する。

ユーザは、問題解決における仮説の生成・検証の流れに応じて、上記の Generator および Tester を呼出せば良い。生成された仮説および環境は、Pool によって処理の中を運ばれて行く。Pool にはその時点において無矛盾な仮説あるいは環境が保存されている。

3. 2 特 徴

HYPROS は上記のような構成により、以下のような特徴を持つ。

- (1) 部分問題に展開できる問題の場合には、部分問題ごとに Generator および Tester を割当てて、その部分問題の解決に必要な仮説や矛盾のみを扱う事が可能である。したがって一度に扱う仮説や矛盾が少なくて済む。これにより無矛盾性の検査等におけるマッチング処理が減少し、処理効率を向上させる事が可能となる。
- (2) 部分問題ごとに Tester による環境の絞り込みを行うので、探索空間が一度に広がってしまう事を防ぐ事が可能である。
- (3) 複数の Tester の local nogood の間で、重複や包含関係にあるものが多数発生してしまうような場合、処理効率の低下が考えられる。しかし Tester の local nogood を収集して global nogood を計算し、さらにこの global nogood に基づいて、各 Tester の local nogood を更新する事により、処理全体の nogood を適正配置し直す事が可能である。
- (4) local nogood の与え方により、探索範囲の動的な制御が可能である。これは優先探索的な処理を行う事が可能である事を示している。また探索範囲を絞り

込みすぎた場合、すなわち Tester の制約がきつすぎて環境がすべて nogood になってしまうような場合でも、その Tester の制約をゆるめた上で、途中経過の環境を保存してある Pool を用いて、改めて処理を再開する事が可能である。このように処理の途中で Tester の local nogood を変更する事ができるが、この場合でも global nogood 計算をやり直す事により M A T M S が管理している nogood の更新を行う事が可能である。すなわち nogood の非単調増加をサポートする事ができる。この枠組みは、事例の一部を変更して設計を行うような場合にも適している。

4. 適用例

基本的な適用例として、N クイーン問題の場合における HYPROS を用いた問題解決機構の構成例について概略を述べる。

まず N クイーン問題を、 $i-1$ 行目まで競合なしにクイーンが置かれている時、 i 行目に新たなクイーンを置くという部分問題の繰返しと考える。この時 Generator と Tester の機能は以下のようなものとなる。

- queen generator: i 行目のクイーンの位置を新たに仮説として生成し Pool に保存する。
- queen tester: 新たに生成されたクイーンの位置が、すでに生成されているクイーンの位置と競合を起さないかどうかを検査し、競合を起さないと判断されたものを Pool に保存する。

N クイーン問題の場合、Tester のオブジェクトは $N-1$ 個生成し、 i 行目のクイーンの位置の検査を行う Tester に対しては、 $1 \sim i-1$ 行目と i 行目との間の競合状態を local nogood として設定しておけば良い。

このような、M A T M S を含む HYPROS による処理を、単純な探索と A T M S とを組合せた処理と比較してみると、逐次型推論マシン P S I - II 上での実行時間は、8 クイーンの場合で約 $1/10$ に改善された。

5. おわりに

A T M S を利用する際に A T M S が持つ欠点を補い、組合せ問題の効率的な解決が行えるようにする事を目標とした問題解決機構 HYPROS の構成と特徴について報告した。今後は具体例に対する適用を通して検討・改良を行い、より汎用性の高い枠組みを構築して行く予定である。また多重世界の枠組みを生かした並列処理についても検討を加えて行きたい。

なお本研究は、第五世代コンピュータプロジェクトの一環として進めているものである。

【参考文献】

- [1] Doyle, J., A Truth Maintenance System, Artificial Intelligence 12, pp.231-272, 1978.
- [2] de Kleer, J., An Assumption-based TMS, Artificial Intelligence 28, pp.127-162, 1986.
- [3] 藤原道, 飯島啓美, 井上克己, ESP による A T M S と問題解決器を融合した仮説推論機構 - A T S R O N -, 情報処理学会第 37 回全国大会講演論文集, 1988.