

5C-6

仮説セットの無矛盾性管理を
備えた高速仮説推論システム

陳慶表 牧野俊朗 石塚満

(東京大学)

1. はじめに

論理に基づく仮説推論は次世代知識処理システムに向けて重要な手法である。^[1,2] 仮説推論は多くの実用問題に適用可能であるが、推論の高速化が必要とされる。J.de Kleerの提案による"An Assumption-based Truth Maintenance System" (ATMS)^[3]を提案した。ATMSは並列性をもつため、探索の高速化の一つの有効な方法である。本稿では、ATMSの考え方を利用しながらも、異なるデータ構造、推論の進め方採ることによって、論理に基づく仮説推論の高速化に利用する方法について記す。

2. 論理に基づく仮説推論

論理に基づく仮説推論とは、対象世界で常に成り立つ完全な知識の集合をF、常に成り立つとは限らない仮説の集合をHとしたとき、観測あるいはゴールOを証明するために必要な次のようなHの部分集合hを求めることである。

$h \subseteq H$ hはHの部分集合である
 $F \cup h \not\vdash \square$ $F \cup h$ は無矛盾または制約に違反しない
 $F \cup h \vdash O$ FとhからOが証明できる

選定する仮説をどのように限定するか、またゴールを満たすのに必要な仮説をいかに速く見つけるかが高速化に向けての課題となる。

3. ATMS

ATMSはn個の要素仮説を組合わせた2^n個のコンテキストから矛盾をひきおこすコンテキストを削除して、無矛盾なコンテキストを管理するものである。環境(可能な仮説の組合せ)をlatticeのノードとして構成して管理することで、複数の環境を保持しながら横型探索を進

めることができる。latticeのあるノードが矛盾になったら、この矛盾のノードを含む上位のノード(Superset)を全部消すことで、仮説の矛盾する組合せの複数回の試行を回避し、高速に仮説集合を絞れる鍵となる。

例) $nogood(A, B) \Rightarrow$
 $(A, B, C), (A, B, D), \dots$
 が全部矛盾になって、削除される。

数多くの要素仮説(例えば1000個)に対処するには、各要素仮説を採用か否かをある位置のビットの1か0かで表わすなどのlatticeデータ表現が必要である。また各要素仮説の組合せを表わすノードの無矛盾性の判定にも時間を要する演算を必要とすることになる。

4. 仮説の組合わせの無矛盾性管理機構を備えた高速仮説推論システム

ATMSの仮説の組合せ管理の考え方を利用し、論理に基づく仮説推論システムを高速化するため、図1に示すようなシステムを作成している。

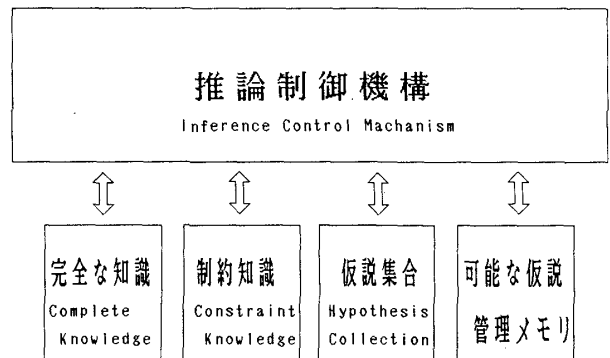


図1 システムの構成

4.1) 可能な仮説の組合せの段階的の形成

多くの問題では仮説に関連するカテゴリーがあり、推論のある局面で必要な仮説を求める場合、特定のカタ

ゴールの仮説のみを対象にすれば十分である。

例えば、部屋 (X) を満たす X を求める場合、部屋 (room_A), 部屋 (room_B) . . . の仮説カテゴリーだけを考えればよく、科目 (数学)、科目 (人工知能) . . . などは考慮の対象外となる。よって仮説をカテゴリーに分類して記述し、図 2 に示すようにカテゴリー毎に順次無矛盾な仮説の組合せを並列的に求めるようにする。これにより A T M S のような lattice を表わすデータ構造を必要せず、仮説の lattice 空間に相当する可能な仮説の組合せ (ノード) を段階的に形成することができる。

このとき、どの仮説カテゴリーを先に選ぶかが効率に大きく影響する。ここで採用しているのは、カテゴリー間の制約 (nogood として記述される) が多いものを優先する方策である。

これによって、考慮中として残す仮説の組合せが少なくなる。

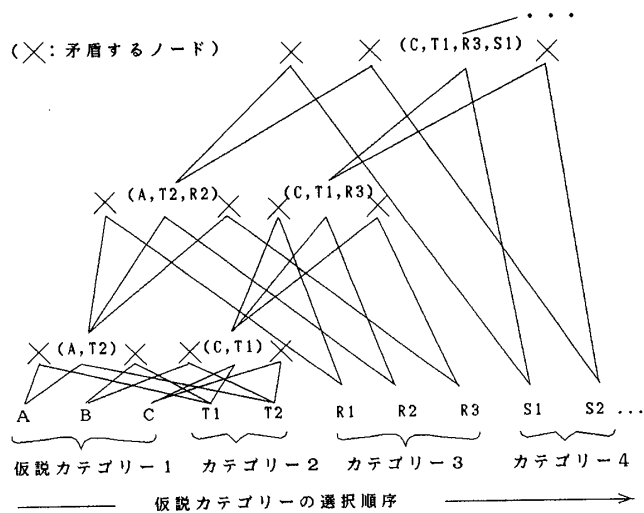


図 2 仮説の lattice 空間における可能な仮説の組合せ (ノード) の段階的形成

4.2) 知識の前処理

ここでは完全な知識 F はホーン節 (基本的に Prolog プログラム)、仮説 H はファクト型知識、制約は A T M S と同様に nogood (,) で表わしている。

仮説と制約知識を解析して上記のように仮説カテゴリーの選択順序を決定しても、F から構成される知識はルール型知識の多段の推論となるため、この選択で推論を進めることが難しい。そこでゴールを頭部にもつホーン節の展開 (unfolding) を繰り返し、展開された形式の表

現に変換する。そして、この知識表現中の仮説の generator に相当するリテラルを上記の選択順序で起動し、可能な仮説の組合せを管理メモリに保持しながら後向き横型探索に相当する推論を実現する。

このようなゴールに関するホーン節の展開は、ゴールの証明に関与する仮説に限定して仮説生成を行う機能にも利用できる。

4.3) 仮説の組合せの Forward Checking

ある仮説の選択が将来必ず矛盾を引き起こすことが予想できるには、この仮説を排除することによって探索の範囲を一層狭くできる。

例) n - q u e e n 問題:

i 行に q u e e n をおくと (i + 1) 行から N 行まで、この q u e e n と競合する場所 (すなわち将来の仮説) を削除する。すなわち、制約をできる限りに早期に考慮し、仮説の削除を行う。

このような機能は制約問題解決で Forward Checking と呼ばれている。この機能の導入による効率化も検討している。

5. あとがき

現在、このシステムは Sicstus-Prolog で S U N 4 上に作成中である。部分的に作成したシステムは 8queen 問題を 10 秒で解き、簡単な回路故障診断では 1 秒以内で結果が得られる。別途作成した設計問題に有効な制約知識のえん緩和による推論制約機構^[4]等も組み込み入れていくことを計画している。

[参考文献]

[1]石塚満:”不完全な知識の操作による次世代知識ベース・システムへのアプローチ”
人工知能学会誌, Vol13, pp. 552-562 (1988.9)
[2]石塚満:”不完全な知識の操作による高次人工知能と知識システム”「多次知識情報の知的処理統合化に関する研究」福村 (編)、オーム社, 1988, II 編, 3 章。
[3]de Kleer, J. ”An Assumption-based Truth Maintenance System” Artificial Intelligence, 28, 1986.
[4]牧野、石塚”仮説推論による回路の機能ブロック設計システム”、人工知能学会全大、No. 11-35 (1989.7)