

述語論理知識ベースに適用できる高速仮説推論システム

2K-8

近藤 朗子 石塚 満
東京大学生産技術研究所

1. はじめに

不完全な知識を取り扱うことにより高次人工知能機能を実現する仮説推論[1]は、次世代知識ベースシステムを構築する上で重要な技術であるが、そのシステムは推論速度が遅いことが最大の問題点となっている。

推論速度を向上する手段として、推論がガイドする役割を果たすヒューリスティックな知識の利用も考えられる。しかし、ヒューリスティックな知識はそれを獲得すること自体が困難である。ここでは宣言的知識表現下で仮説推論を高速化する方法を考える。

仮説推論システムにおいて、推論自体を高速化する方法として、我々は後向き推論とコンパイル法による推論パスネットワークの生成と並列前向き推論による方法[2]、ATMS[3]のlattice構造を利用して並列に解を求める方法[4]を考案、開発してきた。いずれも仮説間の矛盾によって生じるバックトラックを回避することによって推論の効率化が達成されている。しかし、これらの方法は命題論理の知識ベースの範囲にしか適用できない。今回は変数を含む述語論理にも適用できる高速化手法について検討した。

以下にその高速化システム KICK-HOPE (Knowledge-Base Handling Incomplete Knowledge - by Holding Parallel Solution on Environment Lattice)について報告する。

2. 仮説推論システム

まず最初に、仮説推論システムについて、簡単に記しておく。

知識ベースを、完全な知識(対象世界で常に成り立つ知識、事実)の集合 F と、不完全な知識(対象世界で常に成り立つとは限らない知識、仮説)の集合 H に分ける。仮説推論システムの基本動作は、ある観測 0 が与えられた時、

$$\begin{aligned} h &\subseteq H \\ F \cup h &\vdash 0 \\ F \cup h &\not\models \square \end{aligned}$$

であるような H の部分集合 h を求めることである。

このとき、求められた部分集合 h が、完全知識の集合 F と無矛盾でなければならないので、Prolog の機能を利用したインプリメントでは、仮説間の矛盾によって生ずるバックトラックにより、推論が非常に遅くなる。

3. 従来の高速化システムの考え方

推論を行う方法として、後向きトップダウン推論と前向きボトムアップ推論の2通りが考えられる。

Prolog に代表される後向き推論では、ゴールから出発するため問題に関係のないノードを解くことはないが、同じノードを何度も計算する可能性がある。特に仮説推論の場合は、矛盾する仮説のためバックトラックを頻繁に繰り返す。

一方、全探索的な前向き推論では、同じノードを繰り返し計算することはないがゴールに無関係なノードをも生成してしまう。ATMS[3]は前向き推論のプロダクションシステムと組み合わせ、ゴールに向かう推論の筋道をユーザの記述するルールによって制御し、そのもとで無矛盾な仮説の集合(環境)を見出す。

我々の仮説推論システムは純粋に宣言的記述である論理に基づいており、プロダクションシステムにおけるような制御に頼らない。その高速化手法としては、後向き推論と前向き推論の利点を生かし、仮説間の矛盾によって生ずる探索の非効率を避けることが要素となる。

伊藤らによるコンパイル化ネットワークを利用する方法[2]は、はじめにゴールから完全知識のみを用いて推論パスをネットワークとして構成し、その後仮説をこのネットワークに沿って並列に伝播させる方法である。これによって、Prolog の機能によるインプリメントと比べて、1000 倍以上の高速性を達成している。

また、近藤らによる lattice 構造を利用して並列に解を求める方法[4]は、はじめにゴールを仮説の手前のサブゴールにまで展開(unfolding)して、サブゴール毎に仮説を lattice 上にマッピングし、最後にそれらを合成するという方法である。

これらは命題論理の知識ベースに対して極めて大きな速度向上を達成する。しかし、述語論理知識ベースに適用する場合には、たいていの問題は仮説を使わないと探索木の構造が決定できないので、はじめに仮説無しでネットワークを構成したり、ゴールを仮説を使わずに展開することはできない。

特に述語論理では再帰的な述語が使われる。再帰的な述語が使われている場合に仮説を使わないでゴールから筋道をつけようと考えるといつまでも終わらないことが多い。再帰的な述語に対する処理については演繹データベースの分野で研究が盛んである[5]。

今回報告する高速仮説推論システム KICK-HOPE は再帰的な述語も取り扱うことができる。演繹データベースの問い合わせ処理との関連でいうと、QSQR 法と称される手法と関係しているが、矛盾の可能性をもつ仮説を扱う点に特徴がある。

4. KICK-HOPE の高速仮説推論法

KICK-HOPE は基本的にはトップダウンで推論の筋道をつけ、ボトムアップで並列的に仮説を合成する。しかし、それを各ノード毎に行うので述語（再帰的なものを含む）にも対応することができる。

KICK-HOPE の推論の特徴は次の2点である。

- 1) あるノードにユニファイ可能なノードはすべて or 関係として持ち、それぞれを別々に解く。
- 2) and 関係は左から解き、左のノードが確定したら右のノードにユニファイさせる。サポートする仮説はそれぞれのノードで独立に持っておき、最後に合成させる。

KICK-HOPE の and関係、or関係のアルゴリズムは下記の通りである。

<< A and B のアルゴリズム >>

- ① A を解く。（1つまたは複数の確定ノードが生成される。）
- ② A のすべてのノードを B にユニファイさせる。
- ③ ユニファイされた1つまたは複数の B のノードをそれぞれ解く。
- ④ A, B の確定ノード同士でサポートする仮説を合成する。この時無矛盾性チェックを行い、矛盾する仮説の組を除く。

<< A or B のアルゴリズム >>

- ① A, B をそれぞれ解く。（A, B ともに確定ノードが生成される。）
- ② 冗長なものを除く。

知識ベース

完全知識
 $g(X, X) :- s1(X).$
 $g(X, Y) :- s2(X, Z) \& g(Z, Y).$
 $inconsistent :- s2(X, Y) \& s2(Y, Z).$
 仮説
 $s1(a).$ $s2(a, b).$
 $s1(b).$ $s2(b, c).$
 $s1(c).$

ゴール $g(a, X)$

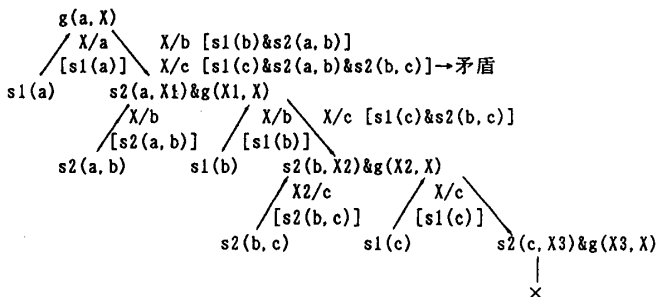


図1 KICK-HOPE の例

ここで確定ノードとは、変数を含まず、かつ

1. サポートする仮説が不要である。

→ true ノード

2. 失敗する。→ false ノード

3. サポートする仮説が決定する。

のいずれかと決定されたものである。

また、仮説の合成の際の無矛盾性チェックや冗長性チェックは、すでにノードが確定した後に行うので、命題論理の場合と同じく Environment lattice 上でのビット演算として高速計算することができる。

この方法による推論の様子を図1に示す。

5. まとめ

今回報告した高速仮説推論システム KICK-HOPE は、次の利点がある。

1. ゴール指向なので、不要なノードは生成しない。
2. 仮説の合成はボトムアップで、ビット演算などの高速計算ができる。
3. 確定したノードは再利用が可能である。

これによって、変数を含む述語論理における仮説推論システムの高速化が可能となる。

システムは SICStus-Prolog を主体とし、仮説の表現と管理を行う部分には C を用いることにより現在作成中である。

<参考文献>

- [1] 石塚：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、vol. 3, No. 5, pp. 552-562 (1988)
- [2] de Kleer, J. : An Assumption-based TMS, Artificial Intelligence, 28, pp. 127-162 (1986)
- [3] 伊藤、石塚：論理制約利用による高速仮説推論システム、情報処理学会人工知能研究会資料、70-5 (1990. 5)
- [4] 近藤、牧野、石塚：Lattice 構造を使った並列横型解法による仮説推論システムの高速化、情報処理学会第40回全国大会論文集、6C-1, pp. 182-183 (1990. 3)
- [5] 西尾、楠見：演繹データベースにおける再帰的な問い合わせの評価法、情報処理、Vol. 29, No. 3, pp. 240-255 (1988)

解 $X=a [s1(a)]$
 $X=b [s1(b) \& s2(a, b)]$

但し、[]内はサポートする仮説を表す。