

知識コンパイルの部分的適用による 高速仮説推論システム

7F-1

遠藤裕明 鶴田三郎 石塚満
東京大学

1. はじめに

われわれは、知識の幅を広げる次世代知識ベースへ向けての実用性も備えた枠組みとして、仮説推論に関する研究を進めている^[1]。そのための第一段階として、不完全な知識(常に成り立つとは限らない知識)を記述し、それに基づいて推論を行うのが仮説推論である。

仮説推論は推論速度が遅いという問題があり、その高速化のために様々な研究がなされている。中でもReiter他の提案したCMS(Clause Management System)^[3]は、論理基盤上でATMS^[2]を一般化、拡張したものと位置づけられ、知識ベースのコンパイルにより推論の高速化を図ると共に、無矛盾性の管理及び欠落した知識の発想を可能とする枠組みでもある。

ここでは、CMSのコンパイルの考え方^{[4][6]}を、仮説推論知識ベースの部分コンパイルに適用し、高速化を図る方法について、報告する。

2. CMS(Clause Management System)

2.1 CMSの概要

CMSは、一般の命題論理節を対象とし、公理節集合(Σ)と、問題解決器から送り込まれた質問節(C)とから、Minimal Support節(S) (以下、MSと書く)を生成することにより、質問節の論理的帰結性や、無矛盾での充足可能性を検討する枠組みである。

更に、効率を向上させるために、公理節集合をコンパイルすることを考える。これは、公理節集合からPrime Implicants(PI) (以下 PI と書く)を導出し、これを利用することにより、質問節に対してMSを生成する方法である。

この PI とは、公理節集合から可能な導出を全て行ったものであるため、これを求めてしまえば、もはや導出する必要はなく、MS節を PI 節と質問節との包含関係から直接高速生成することが可能になる。さらに、コンパイルは基本的に質問がなされる前に行っておけるので、多少時間がかかってもよい。従って、 PI を求めておくことは推論の高速化に大きく寄与する。

2.2 CMSの動作

次に、CMSの実際の動作を示す。

(1) コンパイル

図1はCMSにより公理系 $\Sigma = \{pvqvr, pv\sim q, p\sim r, \sim pvqvs, qvr\sim t\}$ をコンパイルして、 $PI = \{p, qvs, qvr\sim t\}$ を導出する様子である。各シンボルについて、正負のものを組み合わせて消すという作業を、全ての文字の組み合わせに対して行なうというものである。その過程で、他のものを含んでいる知識は、冗長なので消される。

(2) 証明

上の PI に対して、 $C = \{qvr\}$ という質問をする

と、 $S = PI - C$ なので、 $S = \{\sim q, \sim r, s, \sim t\}$ となる。 C を証明するのに必要な知識は $\sim S = qvr\sim t$ である。

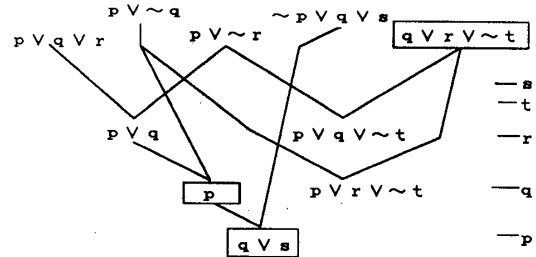


図1 CMSによる知識のコンパイル

3. 部分コンパイルを利用した高速仮説推論システム

CMSを実現するに当たって、コンパイルして PI を導出しておくことの利点はさきに述べたが、実際にはコンパイルの結果、知識の数が多くなり過ぎて計算に膨大な時間がかかる(指数オーダー)という問題点がある。この問題を解決するために、部分コンパイルすることを考える。つまり、知識をいくつかの領域に分けてコンパイルするという方法である。

これは、コンパイルを全部について行わないので、本来のCMSの様に、解を PI と質問節とから簡単に求めることはできなくなる。そのため、コンパイルした知識から、さらに推論を行うが必要になり、システムは前処理(部分コンパイル)との高速仮説推論後推論の2つの部分から構成されることになる。以下にこのアルゴリズムを利用した仮説推論システムを示す。

なお、通常CMSで扱う知識は正リテラルや負リテラルがいくつあっても構わないのであるが、ここではそれをホーン節に限定する。また、CMSは欠落した知識を生成するアブダクションの枠組みであるが、可能な仮説を予め記述しておく実用的な仮説推論を対象とすることにより、知識ベースの部分コンパイルが一層効果的に行えるようになる。

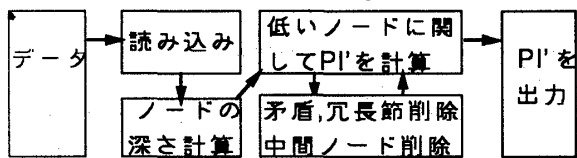
3.1 部分コンパイル

図2にシステムの概要を示す。

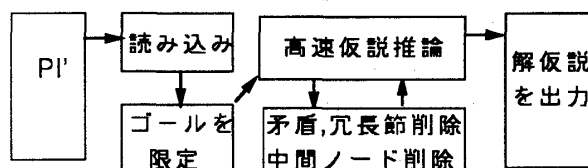
(1) 前処理(部分コンパイル)

まず素データを読み込んで、データを末端ノードから数えた深さで分類する。そして、深さの浅い順に正リテラルを持つ節を取り、その節の負リテラルに対応する正リテラルを持つ節を組み合わせる(導出する)。その際に、矛盾する節、冗長な節を取り除き、更に中間ルールを削除する。中間ルールとは、負リテラルが末端ノードでない節で、高いノード間のルールを表したものである。そこで、最終的に知りたいものは、あるノードを支持する仮説=末端ノードの集合なので、そのノードについて導出が終わったら、そのノードを正リテラルとして、末端ではないノードを負リテラルに持つ節は不要となり、削除してよい。

これを、与えられた深さのノードまで行えばよい。その結果、その深さまでのノードは負リテラルが全て末端ノードであるような節の集合となり、それ以上のノードは、初めのままのルールで存在することになる(図3)。これをPI'として出力し、後に推論を行いたいときに読み込めばよい。



前処理
部分コンパイル



高速仮説推論

図2 システムの概要

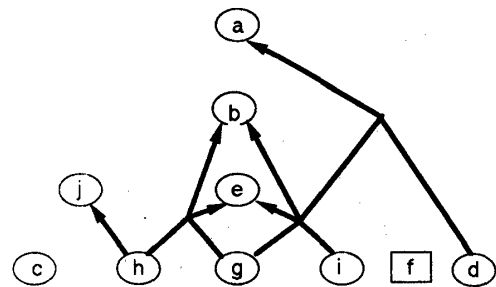
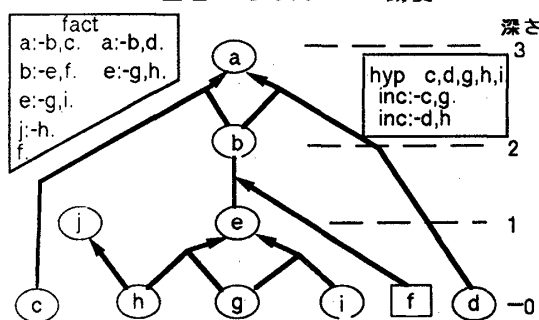


図3 部分コンパイルの例

(2) 部分コンパイル知識による高速仮説推論

その後の推論は、与えられたゴールに対する推論パスネットワークによる高速仮説推論手法^[7]に準拠する。まず、ゴールと仮説以外の末端ノードが特定されるので、無関係な節を削除する。その後、ゴールを正リテラルとして持つ節の負リテラルに対応する、節を組み合わせる。その際と同様に矛盾、冗長節、中間ルールを削除していく。これを繰り返して、ゴールを正リテラルとして持つ節の全ての負リテラルが末端ノードになったとき、計算を終了する。そして、最終的に、ゴールを支持する仮説の集合を解として出力する。

3.2 実験結果

上記のシステムを回路の故障診断に適用した結果を示す。これは、4段の全加算器に入力と出力の値の組を与えてそれを満たす故障状態を全て解として与えるものである。この例で、部分コンパイルを施したものはCPU時間で3.26s、コンパイルしなかったものは23.37sかかっている。

また、単解の探索においても部分コンパイルした場合は常に短時間で解が求められるのに対し、コンパイルしなかった場合は、解の存在場所によってはやく求まったり遅くなったりすることが分かっている。

ところが、探索木の最終段で多くの節が組み合わせられる場合(部分コンパイルで節があまり消えなかった場合)、後の推論の比重が大きく、部分コンパイルの効果は薄い。

4. 結論

3.2の結果から、部分コンパイルによって推論が高速化されることが分かったが、その有効性はデータやゴールの形に依存するようである。それは、部分コンパイルの有効性の根源がただ事前に計算しておくことにあるのではなく、制約により探索空間を減少させることにあるからである。そのため、浅いノード同士にも制約がよく効いて、節の数が減るものでは速くなるが、制約が最終段になって初めて効いてくるようなデータでは大した効果がない。

そのようなデータに対しても高速化を図るためには、制約をうまく利用できるような組合せ方を考えるとか、最終段の手前にサブゴールを作って、最終段の組合せを減らすなど、しなければならないであろう。

また、既にコンパイル済みのノードをゴールとした場合は、推論をしないで解が得られるので当然高速になる。これは、一度解を求めたら、それを蓄えておけば次回から推論せずに解が得られるようになるので学習効果もあると言える。

以上から、いくつか問題点もあるが、部分コンパイルは推論高速化のための有効なアプローチであると考えられる。

5. 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, Vol 3, No. 5, pp.22-32(1988)
- [2] J.de Kleer: "An Assumption-based TMS", Artificial Intelligence 28, pp127-162(1986)
- [3] R.Reiter, J.de Kleer: "Foundations of Assumption-based Truth Maintenance Systems:Preliminary Report", Proc. AAAI-87, pp183-188(1987)
- [4] 鶴田三郎、石塚満: 不完全な知識が関わる知識システムへのCMSの応用に関する基礎研究、人工知能学会全国大会(第3回)、pp.179-182(1989)
- [5] J.de Kleer: "A Comparison of ATMS and CSP Techniques". Proc.of Eleventh International Joint Conference on Artificial Intelligence, pp.290-296(1989)
- [6] 鶴田三郎、石塚満: 命題論理知識ベースのコンパイル法、情処学会、人工知能研資70-6(1990.5)
- [7] 伊藤史朗、石塚満: 論理的制約利用による高速仮説推論システム、情処学会、人工知能研資70-5(1990.5)