

知識と時間の論理のための推論システムの並列化手法

5 C-5

三輪 敦 何 立風 世木 博久 伊藤 英則
名古屋工業大学

1 はじめに

マルチエージェント環境において、知識と時間の論理の自動推論に関する研究は、非常に関心の高い問題である。我々はマルチエージェント系における知識と時間に関する様相論理に基づいた知識表現の体系(以下MAKTとする)をすでに提案しており[2]、MAKTの式は不等式を含んだ一階述語論理式の形に変換が可能である。

変換された論理式の充足可能性判定は、汎用の定理証明器で判定することができる。この推論過程には推論方法によってはさまざまな並列性が含まれており、並列計算機に実装することによって、効率化・高速化が期待できる。そこで本稿では、MAKT式を変換して得られた不等式を含む一階述語論理式の充足可能性判定を、並列計算機上で実現する方法について述べ、実際の並列計算機(富士通 AP1000)上に実装し、いくつかの例題について実験結果を示す。

2 推論システムの概要

様相論理式で表されたMAKTの式は、意味的に等価な一階述語論理の節の集合に変換することができる。

本システムは、そのような変換で得られた節集合の充足可能性の判定アルゴリズムを、Model Elimination procedure(以下MEとする)[3]を基に、不等式処理を加えて並列計算機上に効率よく実現するものである。本システムのアルゴリズム

本システムでは、節は以下の形で表現される。

$$L_1 \vee L_2 \vee \dots \vee L_m \vee I$$

ここで、 $L_1, L_2, \dots, L_m$ はリテラルである。 I は不等式の選言である。証明する節の集合 S 中の任意の1つの節を C (centre chain) とする。ME は、extension, reduction, contraction という以下の3つの操作から成る。

1. extension 操作

通常の導出と似ているが、 C の最も左のリテラル

L_1 と S 中の各節と導出が行う場合に、 L_1 を消去するかわりにマークし、次段の C の中に残す。

2. reduction 操作

C の左のリテラル(マークされていないもの)が、 C の中のマークされたリテラルとマッチするならば、左のリテラルを消去する。

3. contraction 操作

C の一番左のリテラルがマークされていたら、そのリテラルを消去する。マークされていないリテラルが出てくるまで繰り返し行う。

MAKT式を変換した時に現れる不等式を扱うために、次の操作を追加した。

4. 不等式推論操作

C の中には不等式が存在するので、不等式理論に基づいて不等式の項を消去する。

以上の4つの操作を C に対して逐次空節が得られるまで行う。空節が得られず、どの操作も適用できなくなった場合は、初期の C として、 S 中から別の節を選び、再びこれらの操作を行う。

例 以下のような節集合 S が与えられたとする

1. $\neg Q(t_1) \vee \neg P(t_2) \vee t_1 < 5 \vee t_2 < 2$.
2. $\neg P(4) \vee Q(6)$.
3. $P(t_3) \vee t_3 < 4$.
3. を初期の centre chain とすると、以下のよう
に推論がすすめられる。
4. $\neg Q(t_1) \vee P^*(t_2) \vee t_1 < 5 \vee t_2 < 2 \vee t_2 < 4$.
5. $\neg Q(t_1) \vee P^*(t_2) \vee t_1 < 5 \vee t_2 < 4$.
6. $\neg P(4) \vee \neg Q^*(6) \vee P^*(t_2) \vee 6 < 5 \vee t_2 < 4$.
7. $\neg Q^*(6) \vee P^*(4) \vee 6 < 5 \vee 4 < 4$.
8. $6 < 5 \vee 4 < 4$.
9. $\square$

3 並列アルゴリズム

本システムでは、初期の centre chain の選択時と、extension 操作が S 中の複数節に適用できる場合の2つの並列性に着目した。両者とも、空節を導くことができた時点で終了することができ、大きな OR 並列の要素であると考えられる。

前者は、節の集合の中に空節が得られない節が多くある場合はきわめて大きな効果が期待できるが、はじ

A Parallel Automated Reasoning System for Knowledge and Time

Atsushi Miwa, Lifeng He, Hirohisa Seki and Hidenori Itoh.
Nagoya Institute of Technology.
Gokiso-cho, Showa-ku, Nagoya 466, Japan

めに与えられた節の数だけの分岐が考えられるために、節の数が多くなった場合は各プロセッサの状況を送受するための通信メッセージを設け、その稼働状況に応じて制御することとした。

さらに後者は、1回の操作ごとに最大与えられた節の数だけの分岐を生ずる可能性があるため、扱う節が増えた場合の分岐の制御は重要である。これは各セルにキューを設け、それぞれのタスクで管理することとした。

並列計算機 AP1000 への実装

本システムには、セル(プロセッシングエレメント)32台構成の MIMD 型並列計算機(富士通 AP1000)を使用し、プログラムの作成には C 言語を使用した。各セルは、ホストプログラムによって制御され、セル間1対1、1対多の放送、及び同期用の3種類の通信ネットワークで相互に結ばれている。

プログラムは、各セルごとに実行されるセルプログラムと、各セルプログラムを制御する1つのホストプログラムの2種類に分けられる。

ホストプログラムは、節の集合が与えられると、構文解析と構築を行い、節の集合を各セルに分配する。また、推論開始後はセルから推論の結果を受信し、終了判定を行い、推論が終了した場合にセルを停止させる。

セルプログラムは、それぞれ内部にキューを持ち、セルの稼働状況を管理しながら推論を進めていく。一定の期間ごとに、自セルの稼働状況がホストに通知され、また、ホストからは他セルの稼働状況の通知を受ける。この稼働状況をもとにして、推論過程で分岐が発生した際にどのセルに分配をしたらよいかを決定することによって、効率のよい負荷分散を実現している。ただし、稼働状況があまり頻繁に通知されると、通信が多くなり、逆に効率の低下を招くため、問題の規模やセルの数によって、調整が必要になる場合がある。

その他、プログラミングの方針として、オブジェクト指向の概念をとり入れ、節やアトムをオブジェクトとして扱い、簡潔で拡張性のあるデータ構造となるよう配慮した。

4 並列効果の評価

いくつかの例題に対して実験を行った結果を図1に示す。

我々が以前行った SATCHMO/MGTP をもとにした手法 [1] では、ホーン節を多く含む節の集合に対して効果が少ないなど、問題の種類に依存するところが多かったが、今回の ME を用いた手法は、与えられた節の数が多ければ、16台のセルで10倍以上の安定した効果を出すことができた。

また、例題の1つについて、より詳細なデータを図1に示す。

5 おわりに

知識と時間の論理式を変換して得られる、不等式を含む一階述語論理式の自動推論システムを並列計算機上に実現し、その有効性を確認した。

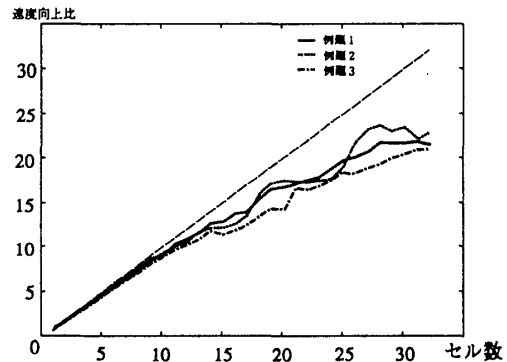


図 1: セルの数と実行時間の関係

セル	実行時間 (速度向上比)	推論 回数	通信 回数	セル 稼働率
1	14.276s (1.00)	1325	122	78%
2	7.198s (1.98)	661	145	76%
4	3.658s (3.90)	330	191	75%
8	1.910s (7.47)	159	283	73%
16	1.130s (12.63)	78	467	68%

表 1: 稼働状況 (節数:265 リテラル数:597)

- * 推論回数は1セル内で行われた操作の最大数
- * 通信回数はホストとの通信も含む総通信回数
- * セル稼働率は全セルの平均値

参考文献

- [1] 三輪 敦, 何 立風, 世木 博久, 伊藤 英則, “知識と時間の論理のための推論システムの並列処理”, 第50回情報処理学会全国大会, 1995.
- [2] L. He, H. Seki and H. Itoh, “WEIHE: An Automated Reasoning System for Multi-Agent Knowledge and Time”, *PRICAI*, 1994.
- [3] Loveland, D.W., “Mechanical theorem proving by model elimination”, *Journal of the ACM* 15, 236-251, 1968.
- [4] R. Manthey and F. Bry “SATCHMO: a theorem prover implemented in Prolog”, *9th CADE*, 1988.