

PIEにおけるゴール表現形式と
そのシミュレーション評価

1R-6

丸山 勉, 酒井 保明, 田中 英彦, 元岡 達
(東京大学 工学部)

1. はじめに

高並列推論エンジンPIEはゴール書き換えモデルに基づき論理型言語を高並列に実行する推論マシンである[1]。PIEの処理単位であるゴールフレームを構成する各データを、新たなゴールフレームが生成されるごとに全てコピーするという従来の方式では、実行される問題にもよるが、そのオーバーヘッドがシステムの処理速度を著しく低下させる危険性がある。

本稿では、新たに生成された子ゴール間で親ゴールのデータ構造を共有することにより処理速度の向上を図る方式とその簡単なシミュレーション結果について述べる。

2. ゴールフレームの表現形式

PIEに於ける処理単位はゴールフレームと呼ばれ、探索木上のある時点に於けるゴールを解くために必要なすべての環境(それ以降実行されるべきゴールリテラルとそれらの変数の束縛状況)から構成される。これらの環境は、OR並列性に基づき新たなゴールが生成されるごとに全てコピーされる。しかし、問題によっては、この環境情報の大きさが非常に大きくなり、そのコピー・転送等のオーバーヘッドがシステムの処理速度を低下させる危険性がある。従って、このような場合においても高速な処理を実現するために、ゴールフレームの表現形式を再検討する必要性がある[2]。

2.1 環境の共有

新しいゴールの生成に於いて環境を全てコピーするのではなく、生成されたゴール間でそれ以前の環境を共有することにより、コピー・転送等のオーバーヘッドを軽減することができる。但し、このとき共有されるデータを管理するメモリへの負荷の集中を避けることが重要である。共有方式はいくつか考えられるが、PIEでは、縮退操作等において定義節を単位として扱うため、定義節単位に環境を分割し、共有の対象とする方式が適当であると考えられる(図1)。共有される環境を管理するメモリへの負荷を軽減するために、共有される環境への子ゴールからの直接の書き込みは行なわず、子ゴールの実

行が終了すると共有されている環境はコピーされ、そこに子ゴールの実行結果が書き込まれるものとする。この方式では、共有によってゴール長を短くすることができるが、あるゴールの実行結果を次に実行されるべきゴールを保持する特定の推論ユニットへ転送しなければならない。

2.2 リテラル構造の共有

このように定義節単位で環境を分割し、共有を行なう場合には、ゴール間で定義節メモリ中にあるリテラルの構造等を共有することが容易にできる。このとき、ゴール中に含まれるべきデータは、変数とユニフィケーションの行なわれた構造データのみであり、それ以外の構造データは共有できるためゴール長を短縮することができる。また、定義節のコピーの手間も不要となる。

3. シミュレーション・モデル

シミュレーション・モデルを図2に示す。主要部の機能を以下に示す。

- UNI:ユニフィケーションを行なう。
- RED:ゴールのコピー及び子ゴールの実行結果の親ゴールへの書き込みを行なう。
- DMA:ゴールのコピーを行なう。
- UPC:UP部の管理を行なう。
- MMC:MM部の管理を行なう。
- IBUF, LMEM, GFM, PGFMは単なるメモリである。

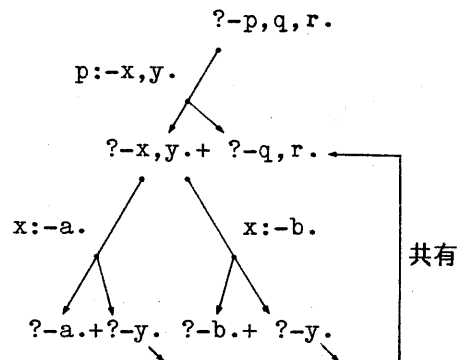


図1 環境の共有

このモデルに於ける処理の手順を以下に示す。

- ①ゴールはDMAによりGFMからIBUFへコピーされる。このとき必要に応じて、PGFM中に格納されている子ゴールの実行結果待ちのゴールもコピーされる。
- ②REDはIBUFからLMEMへゴールをコピーする。親ゴールも共にIBUFにコピーされている場合には、子ゴールの実行結果を親ゴールに書き込みつつLMEMにコピーする。この操作をユニフィケーションが行なわれる回数繰り返す。
- ③UNIがユニフィケーションを行なう。このとき一般に親と子の2つのゴールが生成される。
- ④DMAはその結果をqueueにコピーする。
- ⑤MMCはqueueからゴールを取り出し子ゴールをGFMへ、親ゴールをPGFMへコピーする。

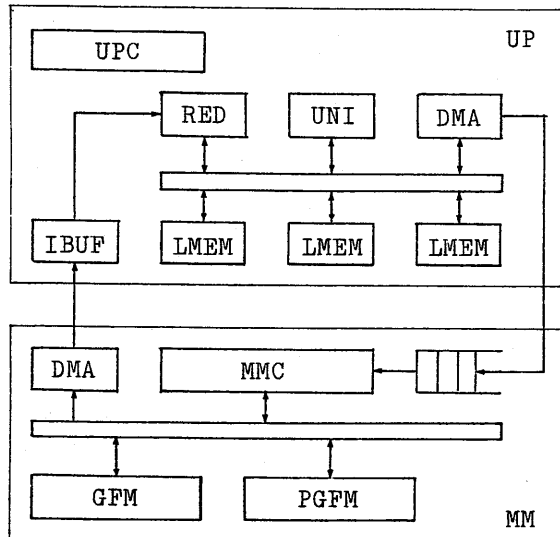


図2 シミュレーション・モデル

表1 シミュレーション結果

	6 queens	Nrev20
従来方式	1038364	275537
	44	87
共有方式	リテラル	579893
	コピー	17
	リテラル	504042
	共有	13

上段は実行ステップ数
下段は平均ゴール長

以上の①～⑤の処理は並列に行なわれる。②～④の処理は各プロセッサ間でLMEMを切り替えることによって行なわれ、ゴールのコピーは行なわれない。この切り替えは②～④の各処理が全て終了するのを待ってから行なわれる。従って最も処理に時間のかかったプロセスがUP部の処理速度を決定することになる。

4. シミュレーション結果

現在、ネットワークのシミュレータへの実装が終了していないので、図2に示すモデルについて推論ユニット1台分のシミュレーション結果を表1に示す。表中の数字は、試作単一化プロセッサ[3]程度の専用ハードウェアを仮定したときの総処理ステップ数である。従来の方式に対するシミュレーションモデルは図2に於けるものとはほぼ同様であるが、ユニフィケーションの後に縮退操作が必要であること、子ゴールの実行結果を親ゴールに書き込む必要がないこと等が異なる。

表1から明らかなように推論ユニット1台に於いては、2.1で述べたゴールの共有を行なうことにより処理速度の向上を実現することができる。また、この場合、リテラル構造の共有を行なったほうが、ゴール長を短縮することができ、より速い処理速度を実現できることがわかる。

4. おわりに

シミュレーションにより、子ゴール間で親ゴールを共有することによってゴール長を短縮し、処理速度を向上させることができることを確認した。今後、多数台の推論ユニットからなるシステムについてのより詳細なシミュレーション評価を行なう予定である。

<参考文献>

- [1] Moto-oka, T., Tanaka, H., et al, "The Architecture of a Parallel Inference Engine - PIE -", FGCS'84, ICOT
- [2] 平田, 丸山他, "高並列推論マシンにおける基本処理単位及び構造記憶に関する考察", LPC'85, ICOT
- [3] Yuhara, M., et al, "A Unify Processor Pilot Machine for PIE", LPC'84, ICOT