

高並列推論エンジンPIE

4P-9

～ゴール書き換えモデルとアーキテクチャ～

後藤 厚宏 , 相田 仁 , 田中 英彦 , 元岡 達

(東京大学 工学部)

1. はじめに

論理型プログラムは、プログラムの仕様と実行ストラテジを分離して検証できることから、知識情報処理における高度なプログラミングに対応できる能力を持つと期待されている。

しかし従来のProlog 処理系では、処理能力が不足しているだけでなく、逐次処理の制約から実行ストラテジの柔軟性が犠牲になっている。

これらの問題に対処するために、高並列推論エンジンPIE [6] の開発を進めている。

PIEでは、論理式の導出過程に忠実なゴール書き換え操作に基づくOR並列処理方式を採用している。本方式では、ゴールの導出において定義節のすべての選択肢を適用するOR並列単一化プロセスを用いる。並列処理は、プロセス間で独立性の高いゴールフレームを次々と受け渡ししながら進む。またゴールフレーム間の依存関係を関係木によって保持し、ゴールフレームの導出機構と制御を切り離している。この結果、プログラムに内在する並列性を活かした高並列処理が実現できると同時に種々の実行ストラテジに柔軟に対応できる。

本発表では、PIEの並列処理方式とアーキテクチャの位置づけについて述べる。

2. 論理型プログラムの並列処理方式

論理型プログラムの処理方式は、証明戦略と節の解釈の仕方(手続き指向、論理式指向、関係表指向)によって表1のように分類できる。

(1) 手続き指向の処理方式は本来 top-down 推論である。この方式では、単一化によって生成される

論理変数の(未定義を含む)インスタンスに着目し、節の手続き的解釈に従って計算を進める。各節は、手続き、またはそれを実行するプロセスの動作の記述であり、変数のインスタンスが手続き(プロセス)間で送受される。

(2) 論理式指向の処理方式では、論理式の導出過程に忠実に論理式自体を操作の対象とする、論理型プログラムの直接実行型であり、ゴール書き換えモデルと知識書き換えモデルの2つがある。主として前者は top-down 推論、後者はbottom-up 推論であるが、top-down 推論において知識書き換えモデルを導入することもある。PIE第一次モデルは基本的にゴール書き換えモデルに属する。

論理式(ゴール、知識)は、単一化機構によって新たな論理式に書き換えられる。証明戦略は、導出を行う節の選択手法によって数多く考えられ、また単一化機構と分けて導入できるため、証明戦略における自由度が大きい。

(3) 関係表指向の処理方式では、関係データベースの見地から、同一述語名を持つ事実節の集合を関係表としてとらえ、導出を表操作によって実現する。

3. 手続き指向方式とアーキテクチャ

手続き指向の処理方式の実現法としては、論理型プログラムを単一代入言語(またはデータフローグラフ)等に翻訳し、記号処理向きデータフローマシン上で動かすことが考えられる[5]。しかし、論理型プログラムの非決定性をどこまで決定的な手続きに変換できるかが問題となる。

[表 1] 論理型プログラムの並列処理モデル

節の解釈\証明戦略		top-down	(bi-directional)	bottom-up
逐次		Prolog	—	—
並列	手続き指向	[5], AND/ORモデル[1]	—	—
	論理式指向	<ゴール書き換えモデル> PIE [6], [3]		<知識書き換えモデル>
	関係表指向	Minker [2], [4]		

この方式の例としてはConery らのAND/ORモデル[1]がある。AND/ORモデルの場合、木状に生成されたプロセス間で照合結果の待ち合わせが必要であり、親プロセスが隘路となる危険性がある。またプロセス間のメッセージ通信の頻度が高い。一般に高並列マシンのモジュール間結合網では遅延が問題となりやすい。このため通信頻度が高いと負荷分散方式と結合網への負担が大きくなる。

4. ゴール書き換えモデルとアーキテクチャ

ゴール書き換えモデルでは、定義節(入力節)とゴール節から新たなゴール節を導出する。このとき定義節とゴール節を分離し、定義節のコピーを数多く用意することによって、処理の独立性を高めることができ、並列処理を実現する上で有利である。

(1) PIEのアーキテクチャ

PIEの主な処理モジュールは、OR並列単一化プロセスによってゴールフレームの書き換えを行うUP(Unify Processor)、実行時にゴールフレームを蓄えるMM(Memory Module)、関係木によってゴールフレームを管理し、UPの単一化の起動制御を行うAC(Activity Controller)である。モジュール間結合網には、負荷分散ネットワークとコマンドネットワークがある。(図1)

(2) 導出操作の起動

原則として、ゴールフレームはデータ駆動的にUPにおける単一化を起動する。このときACがMM内から導出を行うゴールフレームを選択することにより、探索深度やメタ知識に基づいた実行ストラテジが選択できる。また関係木とノード制御コマンドを用いてゴールフレームを活性/不活性化することにより、ゴールフレーム単位の要求駆動制御が可能である。

(3) 負荷分散

PIEでは、各UP内で実行されるOR並列単一化プロセスが独立性の高いゴールフレームを単方向的に生成/消費しながら処理を進める。このためAND/ORモデルにおけるプロセス間での相互通信が不要であり、負荷分散の自由度が大きい。

5. 知識書き換えモデルとアーキテクチャ

単位導出(bottom-up 推論)による知識書き換えモデルは、top-down 推論以上に高い並列性が得られる。しかし、その分無駄な処理も多く数の爆発を起こしやすい。また単位節は処理の進行に伴い新たに生成されるため、導出を行う2つの節をtop-down 推論のように分離することができない。並列処理に

おいては処理要素間の通信が少なくなるように、節を効率よく分割する手法を考える必要がある。

このため知識書き換えモデルは、単独ではなくゴール書き換えモデルと組み合わせた形(両方向推論)で考えるべきであろう。ただし、現段階では両者の使い分け方については不明な点も多い。

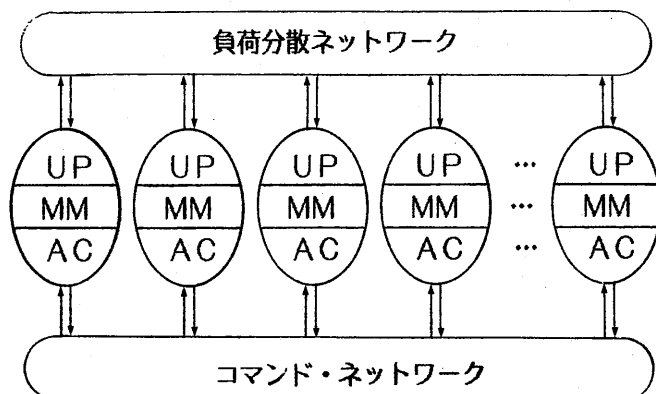
PIEでは、実行時に定義節集合(知識)を更新することによって、フィボナッチ数の計算等における計算の繰り返しを回避すること、および失敗経験を利用して処理を効率化することを検討している。

6. 関係表指向方式とアーキテクチャ

関係表指向の方式は、事実節を多く扱う応用に向いており、知識ベースマシン(関係データベースマシン)において実現される。そこでPIEでは、推論マシンの処理方式としてではなく、推論マシンと知識ベースマシンの接続において重要となる。

《 参考文献 》

- [1] Conery 他: Parallel Interpretation of Logic Programs, Proc. of the 1981 Conf. on Functional Programming Languages and Computer Architecture, pp.163-170, 1981.
- [2] Minker 他: Interfacing Predicate Logic Language and Relational Databases 1st International Logic Programming Conference, 1982.
- [3] 尾内他: リダクション機構に基づくProlog マシンの一構成法, 26回 情処全大 4N-3
- [4] 北上他: 知識ベースシステムKAISERの構想, 26回 情処全大 5C-5.
- [5] 伊藤他: データフロー方式の並列Prolog マシン, Logic Prog. Conf. 83. ICOT.
- [6] 後藤他: 高並列推論エンジンPIEについて, Logic Prog. Conf. 83. ICOT.



[図1] PIEアーキテクチャ