

1H-3

項書き換え系のための並列計算機アーキテクチャ

蒲田順 直井徹 坂部俊樹 稲垣康善

名古屋大学工学部

1. はじめに

項書き換え系(Term Rewriting Systems, TRS)は“項→項”という形の規則の有限集合であり、与えられた項(入力項という)を規則にしたがって書き換えることを計算のステップとみなすような計算モデルである。

本稿では、TRS専用の並列計算機アーキテクチャを提案し、その上での書き換えアルゴリズム、負荷分散について述べる。

2. アーキテクチャ

TRSでは入力項の任意のリデックス(書き換え可能な部分項)を並列に書き換えることが可能であり、この書き換えは局所的なデータのみで行える。このため、疎結合並列計算機アーキテクチャが適していると考えられる。

そこで、PE(Processing Element)を2次元格子状に結合した疎結合並列計算機上に項を分散配置し、並列書き換えを行うことで、TRSの高速実行をめざす。

図1はPEの内部構造である。

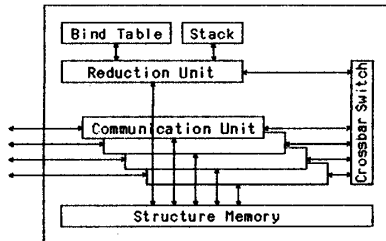


図1 PEの内部構造

SM(Structure Memory)は、入力項の複数の部分項とすべての規則をポインタを用いた木構造として格納するメモリである。

SMの1ワードはシンボル、左の子へのポインタ、右の子へのポインタ、リファレンス・カウンタの4つのフィールドからなっており、ここに木構造のノードを1つ格納する。

部分項は入力ノードのリストで管理され、規則は規則ノードのリストで管理される。また、未使用領域はフリーリストで管理される。

図2は、SM内での部分項と規則の格納法の概念図である。

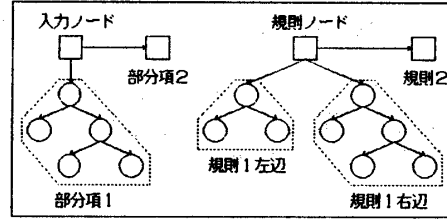


図2 SM内の項、規則

入力項の全体は、SMに分散格納された部分項をリモートポインタによってリンクして表現される。

SMは後述のRUと4つのCUから独立にアクセスされる。

RU(Reduction Unit)は、マイクロコード化されたアルゴリズムにしたがって書き換えを行う。このアルゴリズムは3節で述べる。

CU(Communication Unit)はPE間の通信をパケット交換により行う。また、他のPEからのノード読み出し、書き込み要求を受け、RUとは独立にSMにアクセスする。

ST(Stack)は、書き換えのための基本操作であるマッチング、リライト、ガーベジコレクションのために必要なスタック用メモリであり、RUからアクセスされる。

BT(Bind Table)は、変数名とその変数に代入されている部分項の対応関係を記録するためのメモリである。

3. 書き換えアルゴリズム

書き換えは、マッチング、リライト、ガーベジコレクションの3つのフェーズからなっている。

まず、規則左辺と部分項のマッチングを行う。これに成功すると、規則右辺にしたがって新しい部分項を作り、リデックスと置き換える。最後にリデックス内のノードをガーベジコレクションで回収する。

3.1 マッチング

RUは部分項からリデックスの候補を選び、それと規則左辺とを深さ優先で逐次に比較する。規則左辺に変数が現れた場合は、変数名と、その変数に代入する部分項へのポインタをセットにしてBTに格納する。

他のPEへのポインタ、すなわち、リモートポインタが現れた場合はCUを介して読み出す。

マッチングに失敗した場合、新しいリデックス候補を選び、同じことを繰り返す。このリデックス候補の選択も深さ優先で行う。

3.2 リライト

マッチした規則の右辺にしたがって、RUは新しい部分項を深さ優先で逐次に作る。規則右辺に変数が現れた場合は、BTからその変数に代入されている部分項へのポインタを取り出し接続する。最後に、リデックスへのポインタを新しく作った部分項へのポインタで置き換える。

その後、リデックスへのポインタをもとにガーベジコレクションを開始する。

3.3 ガーベジコレクション (GC)

RUは深さ優先で逐次にノードを回収しフリーリストに加える。

リモートポインタが現れた場合 (PE間にまたがるガーベジが発見された場合) はCUを介してGC要求を出す。GC要求を受けたPEのCUはRUに割り込みをかけGCを行わせる。

無駄な書き換えをしないために、SM内のノードの回収よりもGC要求の送信を優先して行う。

4. 負荷分散

リライトの際、新しい部分項の一部を隣接するPEに分割することで負荷分散を行う。

リデックスの候補をなるべく分散するために、分割すべき部分を静的に解析し、その情報を規則右辺の中に入れておく。その情報をもとに実行時に分割を行う。

分割位置の決定方法として、ユニフィケーションによる方法とトップシンボル一致による方法を考案した。

ユニフィケーションによる方法では、規則左

辺にユニファイする部分項を分割する。(図3)

トップシンボル一致による方法では、規則左辺のトップシンボルと同じトップシンボルを持つ部分項を分割する。(図4)

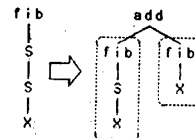


図3 ユニフィケーションによる分割

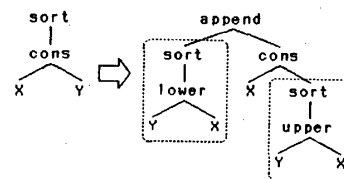


図4 トップシンボル一致による分割

5. シミュレータ

現在、このアーキテクチャのソフトウェアシミュレータを開発中である。

このシミュレータでは各種パラメータの変更はもちろん、マイクロコードをファイルで与えることで、さまざまな書き換えアルゴリズムによるシミュレーションを行うことができる。

6. まとめ

項書き換え系専用の並列計算機アーキテクチャを提案し、その上での書き換え、負荷分散の方法を示した。効率的な書き換えアルゴリズムの検討、シミュレーションによる評価は今後の課題である。

最後に、御討論頂いた平田富夫助教授、外山勝彦中京大講師、杉野花津江助手、酒井正彦助手、結縁祥治助手ならびに研究室の皆様へ感謝致します。

参考文献

- [1] S. Leinwand, et al. CELL AND ENSEMBLE ARCHITECTURE FOR THE REWRITE RULE MACHINE, Proc. Int. Conf. FGCS, pp.869-878(1988)