2C-3

PIEのハードウェアシミュレータ
——定義節メモリ——

垂井俊明、濱中直樹、田中英彦、元岡達
(東京大学　工学部)

1. はじめに

PIE[1]の推論ユニット(IU)のハードウェアシミュレータの試作を行なっている。ここではその中の定義節メモリにおける定義節絞り込みのためのアルゴリズムについて述べる。

2. DMの機能・構造

今回のハードウェアシミュレータでは、DMはメモリモジュール(MM)と同じ構成とした。[2]

PIEのハードウェアシミュレータにおいて、定義節(DT)やゴールフレーム(GF)は読み込みプログラムによって図1のような内部表現に直される。DMはDTを保持し、単一化プロセッサ(UP)の求めに応じ、GF中のひとつのリテラルについて、単一化可能な全てのDTをUPに送る。その際その述語に関する全てのDTを送っていたのでは、UPにおいて単一化に失敗するものがかなり存在し、DM→UP間のDTの転送時間や、UPにおける処理時間が無駄になる。ここで単一化の失敗のかなりの部分は、引数についての簡単な照合操作によって判定できると予想されるので、DMがUPにDTを送る際に、失敗が明白なものを取り除くことが望ましい。これを定義節の絞り込み操作と呼ぶ。具体的には、UPがDMにDTを要求する際に、単にリテラルの述語名だけではなく、引数についての情報を送り、この情報に基づいてDMがDTを検索する。

3. 定義節絞り込みのアルゴリズム

前節で述べたDMにおける絞り込みにおいて、UPからDMに転送する情報を多くすれば絞り込みの効果は上がるが、DMにおける処理が複雑化し、オーバヘッドが増大する。そこで図1の内部表現を考慮して、UPからDMに転送する情報の深さとしては、リテラル部にある情報に限定することにした(ただし構造部へのポインタは除く)。絞り込みに使う引数の数については次節で述べる。

次に検索の方法について述べる。上記の情報のひとつひとつについて検索時に比較を行なう方法もあるが、実行時の手間を最小限におさえるために、あらかじめDMにおいてDTのリテラル部の情報からハッシュ表と、単一化可能なDTの完全なリストを作り、検索時にはその表をひくだけという方法を採用する。この場合引数に未定義変数があると、未定義変数は任意の型、値の引数と単一化可能なので、ハッシュの方法に工夫が必要となる。そこで、ゴールリテラル中の引数が変数であるときは変数以外の定義も検索でき、また逆にゴールリテラル中の引数が変数でないときに、引数が変数であるような定義も検索できるようになっている必要がある。特に後者に関しては、引数が変数であるようなDTはその述語名に関する全てのリストに挿入されている必要があり、データ構造が複雑になる。また、DTがUPに送られる順序関係が、最初のプログラムへの順序関係と等しい必要がある。

第一引数のみを絞り込みに使用する場合の、今回作成したシミュレータにおけるデータ構造を図2に示す。実行時の検索の際には、UPから与えられたゴールリテラルからハッシュ値を計算してハッシュ表をひき、ポインタのリストをたぐれば必要なDT

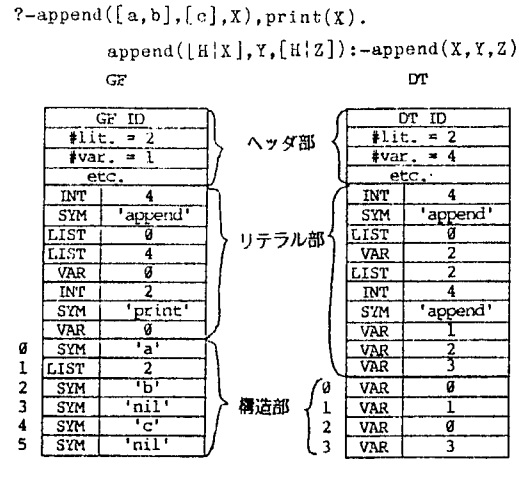

図1　GF, DTの内部表現

全てを検索できる。そこでハッシュ値の衝突が頻発しなければ検索はかなり速くできるが、その反面この方法では、DTを読み込むときの処理がやや複雑になる。しかし、定義の読み込みは一回ですむので、それほど問題にはならない。

4. 定義節絞り込みに使用する引数

ここで定義節絞り込みに使用する引数の位置および数が問題である。引数の数については、上記の検索方法では絞り込みに使用する引数の数を増加させても検索の手間はそれほど増加しない。しかし、引数の数を増加させた場合の絞り込みの効果はどこかで飽和すると予想される一方で、引数の数を増加させるとDTの読み込み時の負担やメモリ使用容量が増加するので、絞り込みに使用する引数の個数はある値に制限するのが良い。また、絞り込みに用いる引数の位置に関しては、一般に最初の方の引数に入力として定数などを書き、後ろの方の引数に出力として変数を書く傾向があるので、最初の何個かの引数を絞り込みに使用すれば良いと予想される。

そこで、絞り込みに用いる引数の数および位置を決定するため、現在ソフトウェアシミュレーションによりそれらを変えた場合の絞り込みの効果を測定しており、絞り込みの効果が飽和する付近に最終的に設定する予定である。

5. おわりに

IUのシミュレータのハードウェアはほぼ完成したので、今後はその上でシミュレーションプログラムを実際に動作させ、具体的な性能の評価や種々のデータの収集を行なう予定である。

参考文献

[1] 元岡 他,「The Architecture of A Parallel Inference Engine—PIE」, Proc. of FGCS '84

[2] 北野 他,「PIEのハードウェアシミュレータ——メモリモジュール——」, 本大会 2C-4

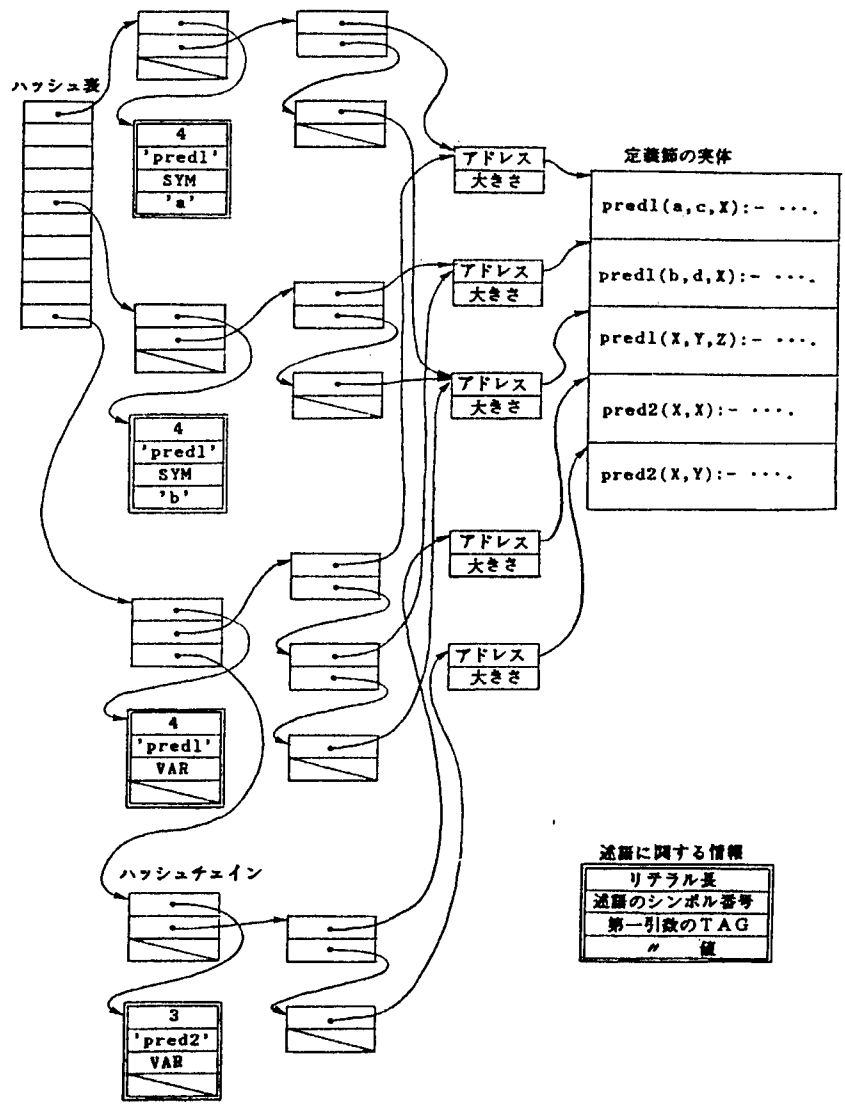

図2 DMにおけるデータ構造