制限付きPrologプログラムのコンパイル技法

3Y-7

周 能法 高木 利久 牛島 和夫
九州大学工学部

1．まえがき

本稿では，ユニフィケーションの代わりに，照合を用いたPrologの処理系を提案する．本処理系は次の三つの段階に分けてPrologプログラムを処理する．1）Prologプログラムを制限付きPrologプログラムに変換する．2）制限付きPrologプログラムを照合木にコンパイルする．3）この照合木を用いて，照合のみを用いる特殊なSLD導出を行う．

次の節では，まず，特殊なSLD導出を提案する．3節では，この特殊なSLD導出を用いて証明できるPrologプログラムの条件を与える．4節では，制限付きPrologプログラムのいくつかのコンパイル技法を提案する．5節では，いくつかの例題を評価することによって提案した技法が有効であることを示す．

2．特殊なSLD導出

Fをファクトの集合，Qをルールの集合，Gを“<-A1,...,Ak”という形のゴールとする．特殊なSLD導出では，Gが空となるまで，次の過程を繰り返す．まず，計算規則によって，Gからサブゴールを選択する．そして，次の二つの導出規則によってGから新しいゴールを導出する．

規則1：FからAiθ=Fjとなるファクトを検索して，Gから<-(A1,...,Ai-1,Ai+1,...,Ak)θを導出する．

規則2：QからAi=Hθとなるルール“H<-B1,...,Bm”を検索して，Gから“<-A1,...,Ai-1,B1θ,...,Bmθ,Ai+1,...,Ak”を導出する．

SLD導出では，ユニフィケーションを用いるのに対して，特殊なSLD導出では，照合でファクトとルールの検索を行う．従って，特殊なSLD導出に基づく処理系の実現に高速な照合アルゴリズムを用いることができる．

3．制限付きPrologプログラム

まず，述語の引数に関して，述語のモード[3]と従属関係という二つの用語を定義する．

pを引数の個数がnである述語記号，Xi(i=1,...,n)をinかoutのどちらかとすると，p(X1,...,Xn)を述語pのモードという．述語のモードはその述語の引数の方向を指定するものである．Xiがinである場合，pの第i番目の引数をin引数といい，Xiがoutである場合，pの第i番目の引数をout引数という．in引数に対しては値を与える一方であるのに対して，out引数に対しては値を与えたり，求めたりすることができる．

rをH<-B1,...,Bmという形をするルール，Wをrに現れる変数の集合とする．変数間の従属関係は2^WからWへの関係である．Biにおける変数間の従属関係Riを次のように定義する．

Ri={X->Y|XはBiのin引数に現れるすべての変数の集合で，Yはout引数に現れる一つの変数である}

ルールrにおける変数間の従属関係Rを

$R=\cup_{i=1,m}Ri$

と定義する．Vを変数の集合，Rを変数の従属関係とする．合成演算V・Rを次のように定義する．

$V\cdot R=\{Y|X\to Y\in R,\ \text{しかも},\ X\subseteq V\}$

(...((V・R)・R)・...・R)をV・R^nと表記する．nはRが式に現れる回数である．

特殊なSLD導出を用いて証明できるPrologプログラムの条件を与える．

条件1：ルールの頭部のout引数は変数でなければならない．

条件2：ルールの頭部のout引数として現われる変数と同じ変数は頭部に二回以上出現してはならない．

条件3：任意のルールrに対して，Wをrに現れるすべての変数の集合，Vをrの頭部のin引数に現れる変数の集合，Rをrにおける変数間の従属関係とすると，

$W=\cup_{i=0,\infty}V\cdot R^i$

例えば，頭部のout引数が本体に現われないようなルールはこの条件を満足しない．

条件4：選択されたサブゴールのin引数はgroundでなければならない．

制限付きPrologプログラムはファクトの集合及び条件1～条件4を満たすルールの集合からなる．与えられたゴールを特殊なSLD導出とPrologの計算規則とPrologの探索戦略とで証明する．

Compile Techniques for Restricted Prolog Programs
Nen-Fa ZHOU, Toshihisa TAKAGI, Kazuo USHIJIMA
KYUSHU UNIVERSITY

match(X,Y)をXθ=Yとなるθを求める述語とする．条件3を満たすPrologプログラムならば，match述語を用いて，そのプログラムを制限付きPrologプログラムに変換することができる．図1は変換の例を示す．

4．制限付きPrologプログラムのコンパイル

ファクトとルールの検索に必要なテスト回数を減らすために，制限付きPrologプログラムのコンパイラに以下の技法を用いた．1）ファクトとルールをルートノードと一入力ノードだけからなるReteネットワーク[1]にコンパイルする．このようなReteネットワークを照合木と呼ぶ．2）照合木の決定性性質を利用する．3）照合だけで評価できる組込み述語を照合木にコンパイルすることによって，照合木が決定的となる可能性を高くし，テスト回数を減らす．

照合木

照合木はルートノードとテストノードと終端ノードとからなる．図2はルール集合及びそれと対応する照合木を示す．“arg2:arg0=tree”というノードは，第二引数のfunctorがtreeであるというテストノードを表す．r1は終端ノードである．

選択されたサブゴールをトークンとして上から下へ，左から右へと照合木に流すことによってファクトとルールの検索を行う．テストノードにおいて，テストが成功したら子ノードを訪問する．テストが失敗したら兄弟ノードを訪問する．終端ノードrに達したならば，サブゴールに対してルールrが適用できることがわかる．

照合木の決定性

任意のトークンがルートノードから一つの終端ノードにしか達することができない場合に，その照合木を決定的という．バックトラック時に，もし，照合木が決定的であれば，最近訪問したノードの次のノードを訪問する必要はなく，直ちに失敗を返せばよい．

また，本体の照合だけで評価できる組込述語，例えば，X≠Y（XとYが頭部のin引数に現われる変数）という組込み述語≠，を照合木にコンパイルすれば，照合木が決定的となる可能性を高くすることができ，テスト回数を減らすことができる．

5．評価と今後の課題

いくつかの例題を用いてファクト及びルールの検索に必要なテスト回数の評価を行った．結果を表1に示す．

処理系に関しては，テスト回数の評価だけではなく，メモリ管理の評価などを含む全般的な評価が必要である．

参考文献

1) Forgy, C.L.: Rete:A Fast Algorithm for Many Pattern/Many Object Pattern Matching Problem, Artif. Intell., Vol.19, No.1, pp.17-37(1982).

2) Robinson, J.A.: A Machine-oriented Logic Based on Resolution Principle, J.ACM, Vol.12, No.1, pp.23-41(1965).

3) Warren, D.H.D.: Implementing PROLOG-Compiling Predicate Logic Programs Vol.1 and 2, Dept. of Artif. Intell. Research Report, No.39 and No.40, University of Edinburgh(1977).

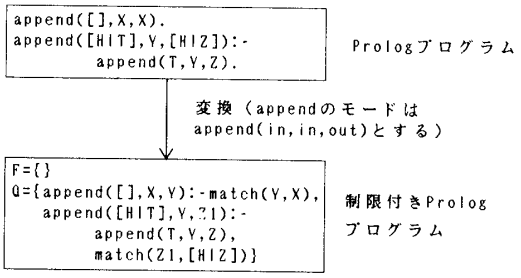

図1．Prologプログラムから制限付きPrologプログラムへの変換

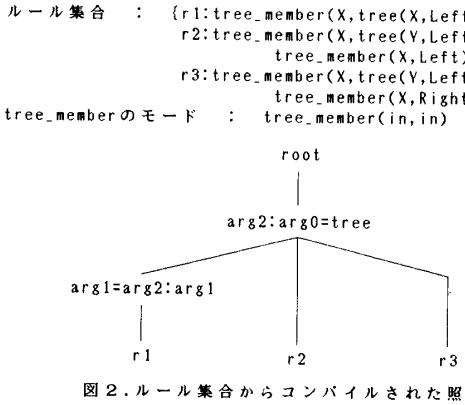

図2．ルール集合からコンパイルされた照合木

表1：評価結果
(X/Y:Xは最初の解を求める場合に必要なテスト回数，Yはすべての解を求める場合に必要なテスト回数)

テスト回数 問題 / 方法	4-queens	tree_member	reverse (10elements)
Robinsonのユニフィケーション法	1391/3149	65/475	199/207
照合木を用いない単純な照合法	154/349	8/52	177/188
本稿で提案した照合法	99/217	6/26	177/178