

Prologにおけるルール照合フィルタの改良について

7Q-7

富士通(株)国際情報社会科学研究所

新谷虎松

1. はじめに

OPS5[Forgy 81]で代表される前向き推論型プロダクションシステムの高速化技法として、RETEアルゴリズム[Forgy 82]やTREATアルゴリズム[Miranker 87]が良く知られている。これらアルゴリズムは、プロダクションシステム(PS)における照合過程を効率良く実行するためのものであり、McDermott[McDermott 78]が提案した照合フィルタの概念を用いてよく説明できる。実際のインプリメンテーションでは、一般に、照合フィルタはルールコンパイラを用いて、ルールの条件部からある種のネットワークを構成することにより実現される。さらに、RETEネットワークでは、ネットワークのノードを共有することによりネットワークのメモリスペースの削減と照合過程の高速化を図っている。

一方、Prolog上でこのようなネットワークに基づくフィルタを効率的に実現することは困難である。そこで、筆者らは、Prologの利点を生かした照合フィルタ生成のためのルールコンパイラを提案した。本コンパイラを組み込んだプロダクションシステムKORE/IEの高速性はOPS5よりすぐれた結果を示した[新谷 87, Shintani 88]。

本アプローチでは、ルールの条件要素毎にLHS節と呼ばれるホーン節をひとつ生成する。これによりPrologのユニフィケーション機能やハッシュ・インデキシング機能を十分に利用した効率的な照合フィルタ(この照合フィルタのことをLHSフィルタと呼ぶ)の生成に成功している。しかしながら、LHSフィルタは、条件要素毎にLHS節を生成する必要性からメモリスペースの点で問題があった。本講演では、特に、メモリスペースを考慮したLHSフィルタの改良について論じる。

2. LHS フィルタ

照合フィルタは、PSにおける認識一行動サイクルでの競合集合を特殊な解釈実行系を介することなく生成するためのものである[McDermott 78]。このフィルタは、①Condition-membership、②Memory-support、及び③Condition-relationshipと呼ぶ3種の知識を組み合わせ

て用いることにより構成される。①は、どの条件要素がどのルールに含まれるかについての知識である。②は、どのワーキングメモリ(WM)要素がどの条件要素を満たしているかについての知識である。③は、ひとつのルールのなかでの複数の条件要素間でどのような関係があるかについての知識である。例えば、RETEアルゴリズムは、先の②と③の知識を組み合わせたフィルタに相当する。また、TREATアルゴリズムは、先の①と②の知識及び新たに④としてConflict-set-support知識を用いたフィルタに相当する[Miranker 88]。④は、seed-orderingと呼ばれる α メモリ間のjoin演算順序を最適化するための知識である。

LHSフィルタでは、②のMemory-supportは導入していない(つまり、照合過程において中間結果を保存しない)。導入しない理由は、Prolog処理系が、一般に、このような中間結果を保存するためには多くのオーバーヘッドを必要とするからである。そこで、LHSフィルタは、①のCondition-membership、③のCondition-relationship、及び④のConflict-set-support知識を組み合わせた照合フィルタとなっている。

LHSフィルタの詳細は文献[新谷 87, Shintani 88]に詳しいが、概略を示すと図1のように示すことができる。図1は、図上で与えられたルール記述をコンパイルすることにより、図下で示すLHSフィルタを生成することを示している。LHSフィルタにおける個々の節はLHS節と呼ばれる。図で示すようにLHS節はルールの条件要素の数だけ生成される。 V_i は、スロット記述 S_i からコンパイル時に具体的に決定される複数のスロット値の並びを表す。 T_i はマッチするWM要素のタイムタグを表す(変数表現される)。LHS節は、WMの変化に伴い呼び出され、その結果(LHS節のヘッドの第一引数が具体化された結果)、第一引数からインスタンスエーションが得られる。

3. メモリスペースの削減

図1で示したように、LHSフィルタでのLHS節の数は条件要素の数だけ生成され、メモリスペースを必要とする。

