

推論システムにおけるルール照合フィルタの並列化

新谷虎松

(株)富士通研究所国際情報社会科学研究所

tora@ias.flab.fujitsu.co.jp

1 はじめに

LHSフィルタは、論理型言語の利点を生かした高速なルール照合フィルタである。本フィルタは、並列論理型言語であるKL1を用いることにより、効果的に並列化することが可能である。本稿では、推論システムにおける並列化照合過程の実現を目指して、KL1に基づくLHSフィルタの並列化について論じる。

2 LHSフィルタ

ルール照合フィルタ[4]は、照合過程を効率化するためのものであり、照合過程の結果（つまり、インスタンスエーション）を特殊な解釈実行系を介することなく生成するために用いられる。ルール照合フィルタの概念は、McDermotにより提案され、(a)Condition Membership, (b)Memory Support, 及び(c)Condition Relationshipと呼ぶ3種の高速化に寄与する知識を組み合わせて用いることにより構成される。例えば、TREATアルゴリズム[3]は、先の(a)と(b)の知識及び新たに(d)としてConflict-set Supportと呼ぶ高速化に寄与する知識を組み合わせたルール照合フィルタに相当する。LHSフィルタは、(a), (c)及び(d)の知識を組み合わせたルール照合フィルタの機能を実現する[6]。LHSフィルタでは、ルール照合フィルタに関する(b)のMemory-supportの機能は採用していない（つまり、照合過程において中間結果を保存しない）。採用しない理由は、Prolog処理系が、一般に、このような中間結果を保存・更新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。むしろ、LHSフィルタでは、Prologの節の高速な参照機能を効果的に利用することにより、照合時の再照合の効率化を図る。

LHSフィルタはルールのLHS（条件部）をコンパイルすることにより、LHS節と呼ばれるPrologのホーン節を用いて実現される[5]。照合過程（つまり、インスタンスエーション[1]の探索）は、WMの変化（つまり、WM要素の付加および削除）をLHS節へのProlog queryに変換することにより達成される。本queryは、LHS節を選択的かつ高速に呼び出すことになる。これは、普通、Prologの節がファンクタ名と第1引数を用いてハッシュインデ

キシングされるからである。

照合過程では、普通、複数のインスタンスエーションが生成される。本アプローチでは、Prologのバックトラッキング機能を素直に用いてLHS節に対するqueryの複数解を求めることにより、インスタンスエーションの全解探索を実現する。この複数のインスタンスエーションには、ひとつのルールにおける変数束縛の組み合わせにより得られる場合と、複数のルールごとに得られるものが含まれる。

3 LHSフィルタの並列化

KL1では、Prologで利用したrepeat-fail機能によるバックトラッキングやヘッドユニフィケーションに基づく論理変数束縛チェック機能はなく、KL1に基づくLHSフィルタの実装においてこれら諸機能にかわる新たなLHS節の枠組みが必要となる。KL1を用いて実現したLHS節を便宜上、p-LHS節と呼ぶ。また、p-LHS節により構成されるルール照合フィルタをP-LHSフィルタと呼ぶ。KL1では、グローバルなデータベース機能がないので、p-LHSフィルタは、PrologにおけるLHSフィルタで取り入れたルール照合フィルタと同様な機能を実現する。

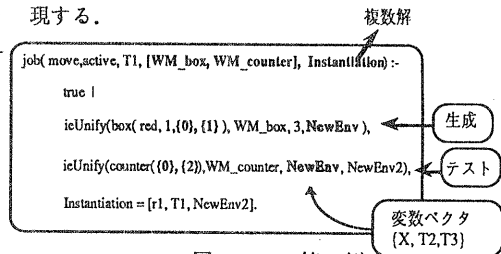


図1. p-LHS節の例

図1は、p-LHS節の概略を示している。本概略は、LHS節に対応するものである。p-LHS節は、LHS節で利用されたPrologの利点としての諸機能をKL1の機能を用いて代替する。図1のp-LHS節において、第4引き数で示した"[WM_box,WM_counter]"はWMを現す。WMは、照合過程でWM要素をp-LHS節の本体で効率的に参照するために、パターン記述のクラス名ごとに分割されている。図1のp-LHS節の本体では、boxクラスおよびcounterクラスに関連したWM要素を参照する必要

があるので、節のヘッドではこれらWM要素に関連した受け口が用意されている。これにより、p-LHS節がプロセスとして起動した際に受けわたされるWMの通信量も減らすことができ、効率化/並列化が可能になる。

図1のp-LHS節の本体における述語icUnifyは、Prologのユニフィケーションに相当する機能を提供する。本ユニファイヤは、文献[2]で実現された高速なユニフィケーションアルゴリズムを利用する。述語icUnifyの第1引数はユニフィケーションで用いられるパターン、第2引数は第1引数とユニファイするために参照するWMを現す。述語icUnifyの第3引数は、ユニフィケーションで変数束縛の制約条件として用いられる環境を現す。変数は、順に、変数ベクタの位置を表す数字を用いて{0},{1}および{3}として表される。p-LHS節本体で、ルール記述で用いられる共有変数は同じベクタの位置を表す数字が用いられる。第1のゴールである述語icUnifyの第4引数"NewEnv"はユニフィケーションにより得られた変数束縛の結果の情報を保存する変数ベクタのリストを表す。p-LHS節では、変数束縛に関連して、パイプライン並列を利用した生成-テスト方式の変数束縛チェックを実現する。これにより、一つのp-LHS節呼び出し(つまり、一つのルール呼び出し)に関連した複数のインスタンス化を得ることができる。

```

job(Name, Status, Tag, WM, Instantiations) :- true !
takeWM(box, WM, Box_WM, WM2),
takeWM(counter, WM2, Counter_WM, WM3),
job1_1(Name, Status, Tag, [Box_WM, Counter_WM], Inst1),
job1_2(Name, Status, Tag, [Box_WM], Inst2),
job1_3(Name, Status, Tag, [Counter_WM], Inst3),
...
merge([Inst1, Inst2, Inst3], Instantiations).

```

図2. インスタンス化の統合

全てのルールを対象にした複数のインスタンス化を得るためには、特別な枠組みが必要である。図2は、p-LHS節の結果を統合するためのKL1節の例である。このようなp-LHS節の結果を統合するKL1節(p-LHSフィルタ節)の集合により、ルール照合フィルタとしてのp-LHSフィルタを実現する。p-LHSフィルタ節は、図1で示されるような複数のp-LHS節を同時に呼び出すための受け口として用いられる。p-LHSフィルタ節は、ルールの条件要素で用いられるクラス名ごとに作られる。これにより、WMの変化の分に即して照合過程を実現する。

図2は、クラス名jobに関連したp-LHS節の例を示している。例えば、クラス名jobに関連したWM要素が作成されると、クラス名jobに関連したp-LHSフィルタ節のみが呼び出される。その後、クラス名jobに関連したp-LHS節が呼び出される。ゴールの実行(つまり、p-LHS節の呼び出し)結果は、KL1の組み込み述語であるmergeにより結合される。これにより、各ゴールは、通信を極力避けることにより効率的に実行される。

4 おわりに

実際のKL1プログラミングでは、KL1節の本体で使われる変数の数やゴールの数が制限されており、p-LHSフィルタ節の実現的な実現はより複雑である。ここでは、引き数をベクタ化したり、p-LHS節の呼び出しを再帰的に呼び出す工夫をすることにより、KL1プログラミングでの制限を回避している。さらに、効率的なKL1プログラミングを実現するためには、ゴールやデータのPE間の受け渡しに関連してMRB(多重参照ビット)を考慮したプログラミングが必要である。ここでの工夫は、KL1プログラミング全般において考慮すべきものであり、本稿では、p-LHSフィルタを論じる上で本質的でないので省略した。今後の課題は、複数のプロセッサを効果的に用いるための実際的な負荷分散方式を取り入れることによる、照合過程の効率化および高速化である。

尚、本研究は第5世代コンピュータプロジェクトの一環として行なわれたものである。

参考文献

- [1]Forgy, C.L.: Rete :A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artificial Intelligence, vol.19, , pp. 17-37, 1982.
- [2]越村三幸, 藤田博, 長谷川隆三: KL1上のユニフィケーションプログラムとその評価, ICOT TM-975, 1990.
- [3]Miranker, D.P.: TREAT:A Better Match Algorithm for AI Production Systems, in AAAI-87, pp.42-47, 1987.
- [4]McDermott, J., A. Newell, J. Moore: The Efficiency of Certain Production Implementations, in Pattern Directed Inference Systems, Academic Press: pp. 155-176, 1978.
- [5]Shintani, T.: A Fast Prolog-based Production System KORE/IE, in the Fifth International Conference and Symposium on Logic Programming, MIT Press, pp.26-41, 1988.
- [6]新谷虎松: prologにおけるプロダクション照合フィルタの高速化, 情報処理学会論文誌, vol.32, No.1, pp.