TM-1179

パターンの依存関係に基づく並列化照合フィルタの最適化

新谷　虎松（富士通）

May, 1992

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

パターンの依存関係に基づく並列化照合フィルタの最適化

(株)富士通研究所 国際情報社会科学研究所

新谷虎松

1 はじめに

筆者等は，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[2]を実現している[3]．本照合フィルタをLHSフィルタと呼ぶ．LHSフィルタは，LHS節と呼ばれるPrologのホーン節を用いて実現され，特別な解釈実行系を用意することなしにPrologプログラムの実行として，照合過程の機能を提供する．LHSフィルタはルールコンパイラーを用いて，ルールの条件部からLHS節を構成することにより実現される．Prologを用いたLHSフィルタは，Prologの利点である論理変数束縛機能，バックトラキング機能および節のインデキシング機能の効率性を十分に取り入れたことによりその高速性を実現した．

LHS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／高速化することが可能である．本論文では，照合過程のさらなる高速化のために並列化照合過程の必要性を背景にして，新たな視点でKL1を用いた並列化LHSフィルタの最適化手法について論じる．ここでは，ひとつのPEにおける照合過程の最適化を指向することにより，LHSフィルタを高速化する．

2 LHSフィルタ

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，生成される．LHSフィルタは，LHS節と呼ばれるPrologのホーン節を用いて実現され，推論システムにおける照合過程の機能を提供する．ルールで用いられる変数は，そのままLHSにおけ

る変数として表現される．図１は，LHS節の生成の概略を表している．LHS節の数はルールの条件要素の数だけ生成される．これはWM要素の変化に伴うLHS節の呼び出しを高速化するために，Prologのインデキシング機能を利用するためのものである．Prologのインデキシング機能は，Prologでは標準的に備わっている内部メカニズムであり，Prologにおいて節の参照を高速化するためのものである．LHS節のヘッドはWM要素の変化を利用するための受け口として利用される．図１においてP$_i$は条件要素のパターンを表している．

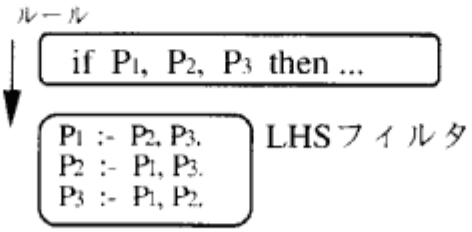

図１．LHSフィルタの概略

WMの変化は（例えば，新たなWM要素をWMに付加すること），LHS節へのProlog queryへと変換することにより，照合過程を行うきっかけとなる．queryは，LHS節のヘッドを呼び出す形式に生成され，インスタンシエーション（つまり，実行可能なルールとその変数束縛情報）を求めるために実行される．LHS節本体のゴールが成功すると，変数束縛の情報が節の本体からLHS節のヘッドへ後向きに伝播される．その結果，呼び出したqueryにインスタンシエーション情報（つまり，実行可能なルールとその変数束縛情報）が出力される（ユニファイされる）．

LHSフィルタでは，照合過程において中間結果を保存しない．中間結果を保存しない理由は，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っている．再照合は，LHS節の本体において，ハッシュインデキシング効果によりWM要素を選択的に参照することにより高速化される．

前向き推論システムでは，普通，照合過程で複数のインスタンシエーションが生成される．この複数のインスタンシエーションには，ひとつのルールから変数束縛の組み合

わせで複数のインスタンシエーションが得られる場合と，複数のルールごとに得られる複数のインスタンシエーションが含まれる．Prologを用いたインプリメンテーションでは，このどちらの場合も，単純にPrologのrepeat-fail機能によるバックトラッキングを素直に用いてLHS節に対するqueryとして複数のインスタンシエーションを求めることができる．

3 KL1に基づくLHSフィルタ

3.1 解決すべき問題点

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．特に，Prologにおけるバックトラック機能は，インスタンシエーションの複数解を求めるための重要な機能となっている．さらに，KL1では，Prologのようにオブジェクトレベルの変数とメタレベルの変数を同じKL1の変数で表現することができない．ルールで記述された変数をユニファイ（もしくは，管理）する新たに特別な枠組みを導入する必要がある．オブジェクトレベル変数管理の効率化は，照合過程の高速性を左右する本質的な課題である．本研究では，オブジェクトレベルの変数を効率的に扱うために，越村等により開発された基底項表現に基づくユニフィケーションアルゴリズム[1]を利用する．

LHS節の本体は，WM要素が前もってassertされていることが前提となり，ヘッドユニフィケーション機能やretract機能を用いてWM要素のチェックおよび更新を実現している．KL1では，このようなassert-retract機能に基づくグローバルなデータベース機能はなくWM（および競合集合）をシステムの引数として（例えば，ベクタメモリなどを利用して）持って回る必要がある．これにより，Prologで採用したようなインデキシングによる効率的な高速性は犠牲になるが，プログラムの並列性や拡張性は向上する．

3.2 実現方式

図2は，LHSフィルタ節の実現方式の概略を示している．図2において，楕円はルールの条件要素のルールパターンを表している．ここで，３種類のルールパターンが色の濃淡の違いで例示されている．図2上は，Prologに基づく実現方式であり，WMがLHSフィルタ節とは独立に参照され，変数束縛のチェックがLHS節の本体でバックトラッキングによりチェックされる．一方，図2下は，KL1に基づく実現方式を表している．ここでは，WMがLHS節の引き数として実現され，変数束縛のチェックは，Prologにおけるバックトラック機能のかわりに生成-テスト法によるパイプライン並列を用いてインスタンシエーションの検索が実現される．このインスタンシエーションは，ひとつのルールから得られる複数のインスタンシエーションに相当する．

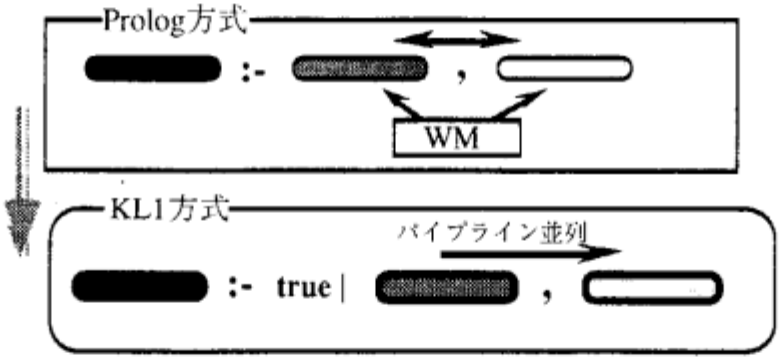

図２．LHSフィルタの実現方式

KL1に基づく実現方式において，全てのルールを対象にした複数のインスタンシエーションを得るためには，特別な枠組みが必要である．図3は，このような枠組みを示している．図３では，ルールr1およびr2により得られるLHS節が示されている．ここで，矢印で示された節は，同じパターン（つまり，黒パターン）を表している．このパターンとマッチするWM要素が生じれば，これらLHS節の両方から結果を得る必要がある（ここでは，OR並列が実行される）．ところが，KL1の実行メカニズムでは，一方のコミットされたLHS節のみが実行される．そこで，図３下で示すように（AND並列により）

各々のLHS節の実行結果を集めるマージ節を新たに導入する．本マージ節を照合マージ節と呼ぶ．照合マージ節により，全てのルールを対象にした複数のインスタンシエーションが得られる．照合マージ節の特長は，本体のゴール（つまり，LHS節の呼び出し）を独立的に行なうことができ(PE間の通信を極力避ける），効果的にKL1の並列性を引き出すことができることである．

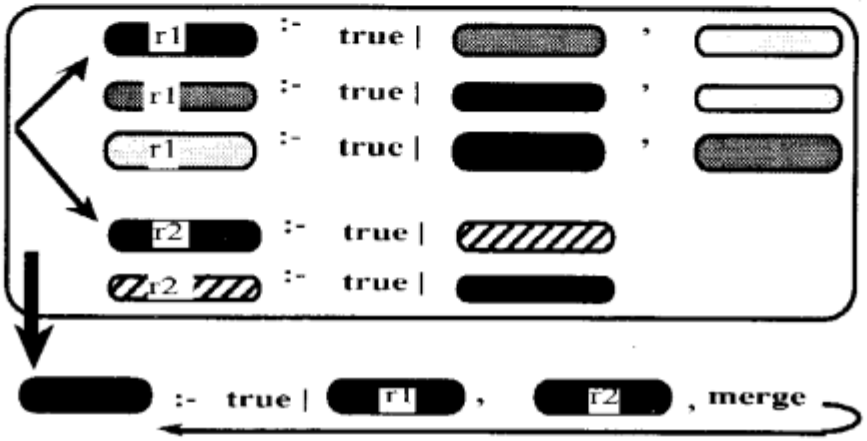

図３．照合マージ節の概略

4 最適化方式

照合マージにおいて，LHS節がインスタンシエーションの生成に関連して完全に独立的（つまり，LHS節のヘッドパターンが表すWM要素がユニークに決定されること）であれば，排他的にLHS節を実行でき，また実行結果をマージすることなしに独立的にインスタンシエーションを求めることができる．

LHS節呼び出しの排他性は，LHS節のヘッドのパターン（LHS節パターンと呼ぶ）に着目してそれらパターン間の包含関係を求めることにより，チェックできる．パターンの包含関係は，ユニフィケーションを用いて得られる．例えば，ユニフィケーションunify(X, Y, Z)により得られるZがXと等しくなれば，パターンXはYに含まれる($X \subseteq Y$)と定義する．もし，XとYがユニファイできなければXとYは独立している（排他的）とす

る．排他的なLHS節パターンを持ったLHS節は，呼び出しが排他的に実行可能である．

具体的には，パターンの包含関係は，ユニフィケーションを用いて次に示すアルゴリズムＰで定義される．

```
if   unify(X, Y, Z)
     if  X = Z  then  X ⊆ Y  ..... Case 1
     if  Y = Z  then  X ⊇ Y        Case 2
     else
        X or Y                     Case 3
else
     X  exclusive or Y             Case 4
```

アルゴリズムＰにおいて，unify(X, Y, Z)により，ＺはパターンＸとパターンＹをユニファイして得られることを示している．パターンの包含関係は，アルゴリズムＰにおけるCase1からCase4までの４つに場合わけされる．LHS節がインスタンシエーションの生成に関連して完全に独立的である場合は，Case4の時である．

P1: job(name=X, status=active)
P2: job(name=put, status=active)
P3: job(name=put, status=Y)
P4: job(name=get, status=sleep)

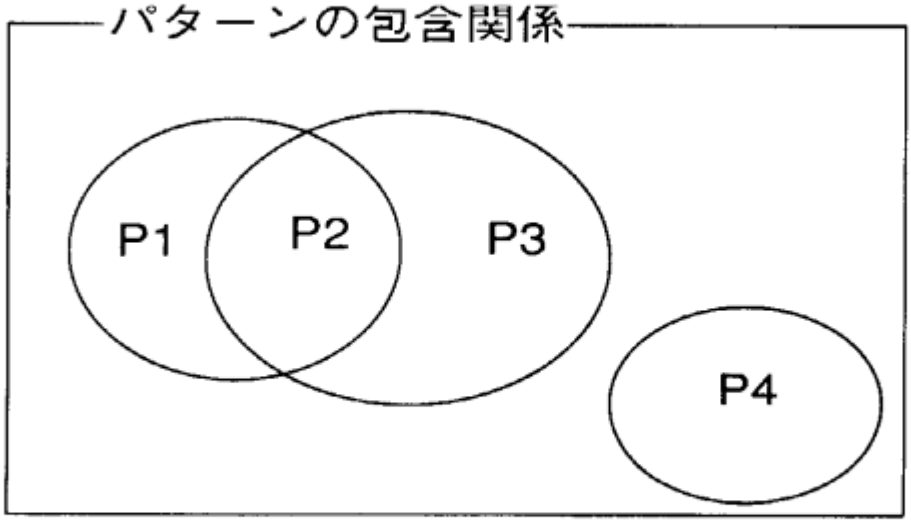

図４．パターンの包含関係の例

ここで，図４を用いてパターンの包含関係の例を示す．ここで，p1,p2,p3,およびp4は図で示すようなパターンを表す．例えば，P1とP2の関係は，それぞれをXおよびYに対応させると，Case2になる．P1とP3はCase3であり，P1とP4はCase4の例であるパターン（= {P1, P2, P3,P4}）の包含関係を図示すると図４下のようになる．

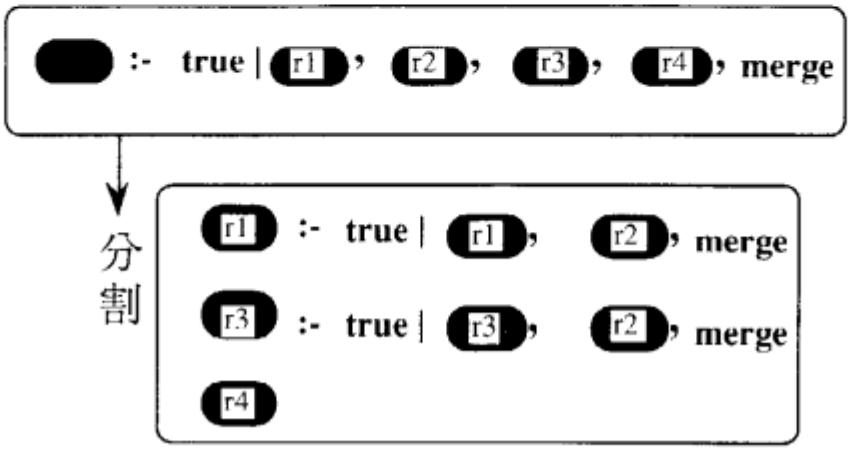

図5．照合マージ節の分割

もし，照合マージ節におけるLHS節パターンが排他的である場合は，インスタンシエーションの探索に関連して無駄なLHS節の呼び出しが行なわれることになる．このような無駄なLHS節の呼び出しを回避するために，照合マージ節を効果的に分割可能である．図5は，LHS節パターンr1,r2,r3,r4の包含関係（ここでは，r1,r3,r4は排他的であり，r2⊆r1, r2 ⊆ r3）に着目して，図5上で示された照合マージ節が図5下で示される照合マージ節に分割された例を示している．分割により，無駄なLHS節の呼び出しを回避できる．最良の場合（つまり，全てのLHS節のヘッドパターンが排他的な場合），照合マージ節が不必要になる．

さらに，LHSフィルタ節の分割に関連して，Case1およびCase2の場合も特別に考慮することにより，効率的なLHSフィルタ節を構成できる．例えば，図4の例において，パターンP1およびパターンP2はP2⊆P1の包含関係がある．この時，パターンP2に関連し

たパターンを先にチェックすることにより，パターンP1のチェックを回避することができる．つまり，もし，P2の呼び出しが失敗すれば，P1も必ず失敗するのでP1のチェックを回避することができる．

5　評価

図6のグラフにおいて，新LHSフィルタは，旧LHSフィルタの照合マージ節を分割したものである．グラフにおいて，縦軸はルールひとつあたりの平均実行時間(CPU時間(msec))である．横軸はルール数を示している．本性能評価で用いたテストルールは，WMの要素を次々に付加して行くものである．本テストルールは，図7で示すように，その実行においてjoin演算も含まれ，通常のルールプログラミングで良く用いられる形態のサブセットとなっている．図6で示す例では，新LHSフィルタは，ルール数にほとんど依存しない理想的な実行速度を達成している．

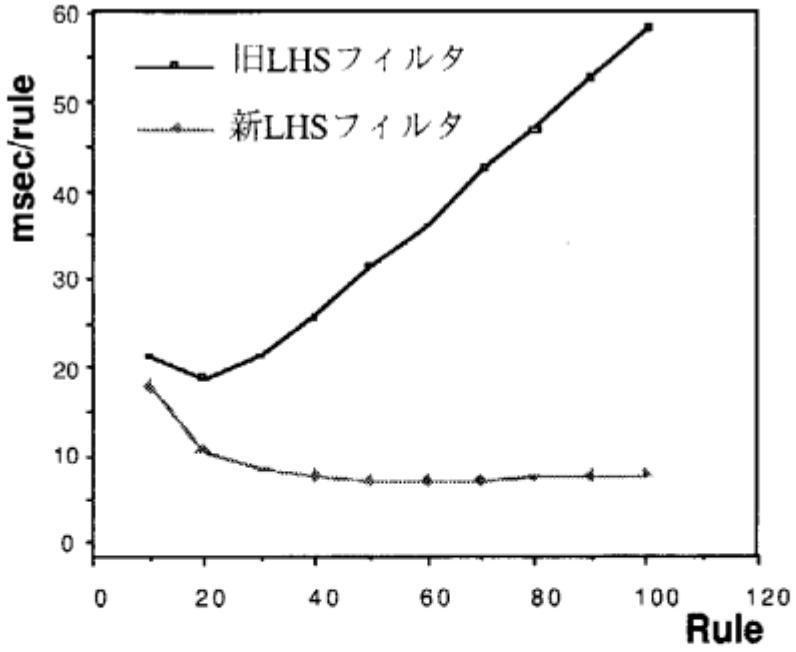

図6．LHSフィルタの評価

```
r1 : if f(s1=1, s2=X) & g(s1=1, s2=X)
     then
       make(f(s1=2, s2=2)) & make(g(s1=2, s2=2)).

r2 : if f(s1=2, s2=X) & g(s1=2, s2=X)
     then
       make(f(s1=3, s2=3)) & make(g(s1=3, s2=3)).
     :
     :
rn: if f(s1=n, s2=X) & g(s1=n, s2=X)
    then
       make(f(s1=n+1, s2=X+1)) & make(g(s1=n+1, s2=X+1)).
```

図7. テストルール

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，LHS節や照合マージ節の実現は，複雑である．ここでは，引き数をベクタ化したり，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，本論文では，LHSフィルタを論じる上で本質的でないので省略した．

本アプローチは，並列度を考慮した実現手法を導入したことにより，LHSフィルタの枠組みは，台数効果を期待できる．図8のグラフは，図７で示した１００個のテストルールを用いて，旧LHSフィルタにおける台数効果を計測したものである．ここで，旧LHSフィルタを用いたのは，旧LHSフィルタは新LHSフィルタに比べ照合過程において実行プロセスの数がが多くなるので，あえて台数効果を計測するために選んだ．負荷分散は，LHS節の呼び出し時にプロセッサ番号をランダムに割り振ることにより実現した．グラフの縦軸は，実行時間(mscc)を表している．横軸は，PEの台数を表している．図８で示すように，PEを７台までは台数効果を得ている．ここででは，ルール数が１００個

であったので7台までの台数効果であったが，ルール数をさらに増やすことにより，PEの数に比例した台数効果を得ることができる．

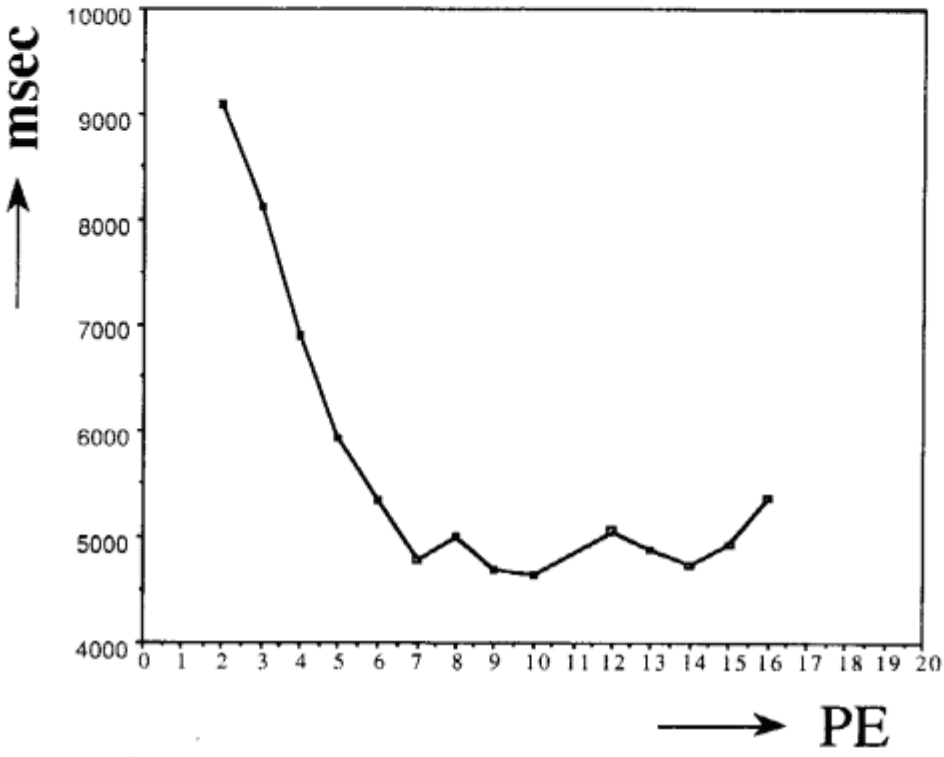

図8．LHSフィルタにおけるPE台数効果

6 おわりに

本研究では，照合過程のさらなる高速化のために並列化照合過程の必要性を背景にして，新たな視点でKL1を用いたLHSフィルタの並列化およびその最適化について論じた．パターンの包含関係を考慮したLHSフィルタは，最良の場合，KL1の節インデキシング機能を利用することにより，ルール数にほとんど依存しない理想的な実行速度を達成した．LHSフィルタを用いた照合過程は，KL1の並列実行形式を考慮したことにより（つまり，プロセッサ間の通信を極力回避したことにより）十分な並列性を得ることができた．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／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]越村, 藤田, 長谷川: 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.

[2]McDermott, J., A. Newell, J. Moore: The Efficiency of Certain Production Implementations, in Pattern Directed Inference Systems, Academic Press: pp. 155-176, 1978.

[3]新谷: 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. 20-31, 1991.