

# 知識ベースマシンMu-X (2)

6Q-2

— キュエリ解析方式 —

仲瀬 明彦 酒井 浩 物井 秀俊 伊藤 文英 三友 雄司  
 (東芝総合研究所) (ICOT) (日本システム)

## 1. はじめに

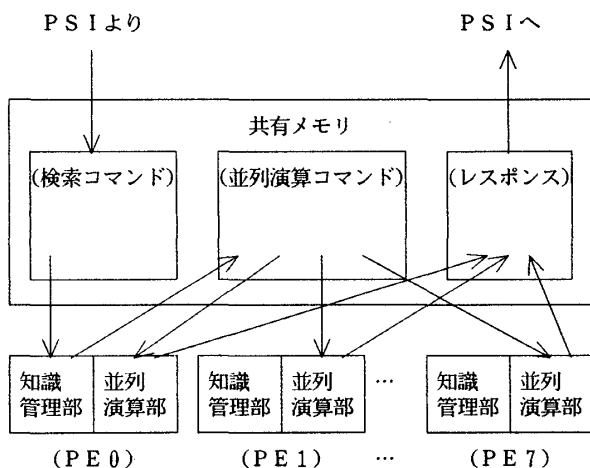
これまでに筆者らは、知識ベースマシンMu-Xのハードウェアの概要、ソフトウェアの概要、また推論マシンPSIのバックエンドプロセッサとしてMu-Xを使用する場合の、PSIからの知識ベース検索用言語について述べた。[1]

Mu-X上のソフトウェアは、Mu-X内のリレーションを管理しPSIからの検索コマンドを解析し検索の戦略を立てる知識管理部と、知識管理部の立てた戦略を元に実際の検索演算を行う並列演算部に分けられる。両ソフトウェアとも、Mu-Xの各PE上で稼動している。

以下にMu-X上のソフトウェアの動作の概要についてべる。

PSIからの知識ベース検索コマンドは、Mu-XのFEPを通じて共有メモリ上に書き込まれるが、Mu-Xの1つのPEの知識管理部がこの検索コマンドを解析し、並列演算のコマンド列を共有メモリ上に書き込む。次に全PEの並列演算部が、共有メモリ上の並列演算のコマンド列を実行し、結果を共有メモリに書き込む。最後にPSIからのコマンドを解析したPEの知識管理部が、FEP経由で検索結果をPSIに返す。(第1図)

本稿では、知識管理部における知識ベース検索コマンドの解析方式について述べる。



第1図 Mu-X上のソフトウェアの動作の概要

## 2. PSIからの検索コマンドと並列演算部の概要

PSIからMu-Xに送られる検索コマンドはretrieveコマンドと呼ばれ、4引数の項の形をしており、その関数子及び引数の意味は以下の通りである。

- 関数子: retrieve
  - 第1引数: 結果リレーションの指定
  - 第2引数: 検索対象リレーションと検索条件の論理式
  - 第3引数: 終了ステータス
  - 第4引数: 検索に要したコスト
- 以下にretrieveコマンドの一例を示す

retrieve(a(X.Y),((b(X.Z).c(f(Y).Z));d(X.Y.a)),S.C)

これは、リレーションcを第1属性でselectionしたものの第2属性とリレーションbの第2属性との間でjoinを行い、その結果生じたリレーションと、リレーションdを第3属性でセレクションしたリレーションをappendしたものを、リレーションaとするものである。

また並列演算部に発行されるコマンドは、以下の3つに分類される。

- ①関係のいくつかの属性に対するselection
- ②2つの関係のjoin
- ③2つの関係のappend

知識管理部では、PSIからの項形式の検索コマンドを、並列演算部での実行効率を考慮しながら並列演算部へのコマンド列に変換する。

## 3. 検索コマンドの解析, 変換方式

PSIからの検索コマンドは以下に述べる3つのステップに従って並列演算コマンドに変換される。

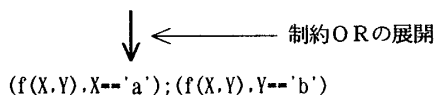
### 3.1 キュエリ解析木の作成

キュエリは、以下の形式を持つキュエリ解析木に変換される。

- ① 解析木のルートノードは、キュエリの結果リレーションを指定しているノードである。
- ② ルートノードの下ノードには、検索の対象となるリレーションを指定したノードもしくは、検索の際の条件を指定したノードが繋がっている関係がAND関係であるかOR関係であるかにより、AND/ORノードとなる。

③ 制約を指定したノードのOR結合は展開され、制約はリレーションのノードとのAND関係としてのみ現われるようにする。

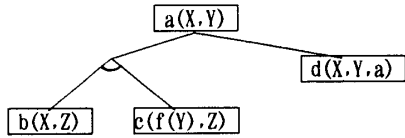
例: (f(X.Y), (X=='a';Y=='b'))



Knowledgebase machine Mu-X(2) -Query analysing methods-

Akihiko Nakase Hiroshi Sakai (Toshiba R&D Center)  
 Hidetoshi Monoi Fumihide Itoh (ICOT)  
 Yuji Mitomo (Japan Systems Co.)

本稿に示したretrieveコマンドの例をクエリ解析木に変換した様子を第2図に示す。



第2図 クエリ解析木(1)

3. 2 クエリ解析木の変換

以上の操作により作成されたクエリ解析木を等価変換し、単純な変数のみが関数子の中に現われる検索対象リレーションと、その制約条件のAND, ORの形式にする。この変換は以下の3ステップに従って行われる。

(1) 検索対象リレーションの中に単純な変数のみが現われる様にする。

検索対象リレーションの引数としては、①定数、②変数、③変数を含まない項、④変数を含む項、の4種類が考えられ得る。①、③については、selectionを行ない、④については、ユニフィケーションを含むselectionを行なうことにより単純な変数のみを含む検索対象リレーションを作成する。

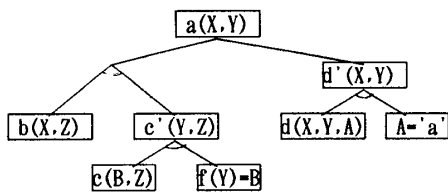
例えば、検索コマンド例内の“c(f(Y),Z)”、“d(X,Y,a)”は、“c'(Y,Z) ← (c(A,Z),f(Y)=A)”

(selection)

“d'(X,Y) ← (d(X,Y,B),B=a)”

(selection)

で示されるc',d'に変換される。クエリ解析木は第3図の様になる。



第3図 クエリ解析木(2)

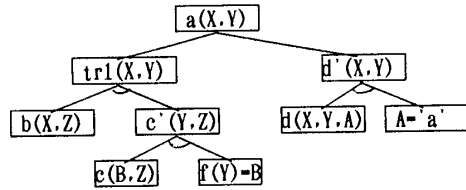
(2) 中間リレーションのノードを作る

以上の様に作成されたクエリ解析木において、AND, ORに分岐している部分で、ノード内にリレーション指定の無いノードに中間リレーションを割りつける。この場合中間リレーションのタプル構成は、各AND/ORノードにおいて以下の規則に従う。

(ANDノード) : あるANDノードのタプル構成は1つ上位のノードのリレーションのタプル構成と一致している。

(ORノード) : あるORノードでは、そのノードの1つ下のノードで出現し、かつ1つ上のノードでも出現する変数からなるタプル構成をとる。

本稿中の例に対しては第4図の形式のクエリ解析木に変換される。



第4図 クエリ解析木(3)

(3) 一時結合リレーションを作成する

ANDノードに3つ以上のリレーションノードが接続されている場合、Mu-Xの並列演算処理では、1回のjoin演算で2つのリレーションしか扱わないので、2回以上のjoin演算に分解する。そこで、3つ以上のリレーションノードを持つANDノードから2つのリレーションを取り出しANDノードと取り出されたリレーションの間にjoinの中間結果を作る。これを一時結合リレーションと呼ぶ。

一時結合リレーションのタプルは、2つのリレーション内に現われる変数で、1つ上のノードリレーションに現われるか兄弟リレーションに現われる変数からなる。

3. 3 並列処理コマンドの作成

最終的に作成されたクエリ解析木より並列処理コマンド列を作成する。並列処理コマンドは、クエリ解析木の1つの1レベルの部分木(1つの親ノードとそれに繋がっている子ノードの組)に対応するので、クエリ解析木をbottom upに走査してコマンドとして生成されたものから順に書き出して行くと並列処理コマンド列が生成される。本稿におけるクエリ解析木の例より発行されるコマンド列は以下の通りである。

```
[c'(Y,Z) ← c(B,Z),f(Y)=B] selection
[tr1(X,Y) ← b(X,Z),c'(Y,Z)] join
[d'(X,Y) ← d(X,Y,A),A='a'] selection
[a(X,Y) ← tr1(X,Y);d'(X,Y)] append
```

第5図 生成された並列コマンド列

4. おわりに

以上、Mu-X上でPSIからの検索コマンドを解析して並列処理コマンドに変換する方法について述べた。クエリ解析木を変換する際に対象とするリレーションの大きさなどを考慮に入れるとより最適な並列コマンドが発行されると思われる。クエリ解析の最適化も含めた性能評価が今後の課題である。

[参考文献]

[1] 物井, 他, 「知識ベースマシンMu-X(1)~(4)」, 第36回情報処理学会全国大会予稿, 5E-4~5E-7, 1988