知識ベースマシンＭｕ－Ｘ（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

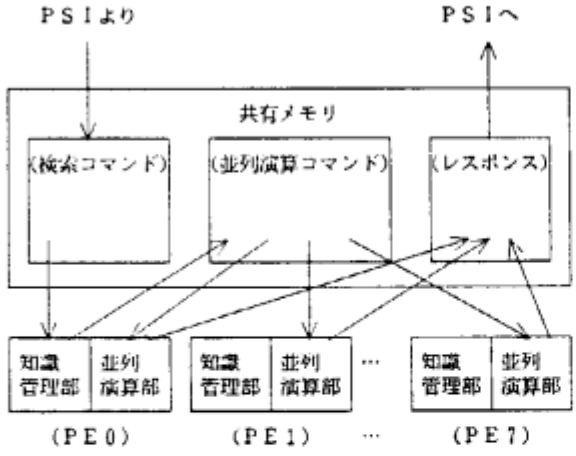
これまでに筆者らは，知識ベースマシンＭｕ－Ｘのハードウェアの概要，ソフトウェアの概要，また推論マシンＰＳＩのバックエンドプロセッサとしてＭｕ－Ｘを使用する場合の，ＰＳＩからの知識ベース検索用言語について述べた．［１］

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　ルートノードの下のノードには，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')

↓ ←――― 制約ＯＲの展開

(f(X,Y),X=='a');(f(X,Y),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コンマンドの例をキュエリ解析木に変換した様子を第２図に示す．

第2図　キュエリ解析木（１）

３．２　キュエリ解析木の変換

以上の操作により作成されたキュ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．キュエリ解析木は第３図の様になる．

第3図　キュエリ解析木（２）

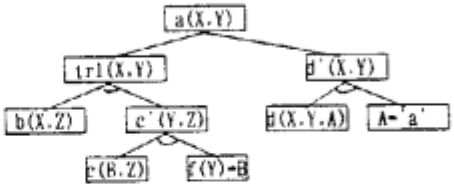
(2) 中間リレーションのノードを作る

以上の様に作成されたキュ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図の形式のキュエリ解析木に変換される．



第4図　キュエリ解析木（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キュ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，キュエリ解析木の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，キュエリ解析木を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．本稿におけるキュ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．キュ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い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．キュ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(I) ～(4)」，第３６回情報処理学会全国大会予稿，５Ｅ－４～５Ｅ－７，１９８８