

知識ベースマシン Mu-X (3)

6Q-3

～表型言語から項型言語への変換～

武脇敏晃, 酒井浩, 柴山茂樹, 物井秀俊, 森田幸伯
 榎東芝 I C O T

1. はじめに

知識ベースマシンMu-X[1] と推論マシンPSI とのインタフェースとして項を基本とした問合わせ言語[2] を提案した。この問合わせ言語は関係スキーマに格納された項集合に対する操作言語といえる。

データベース操作の言語としてSQL がJIS で規格化され利用されつつあるが、使いこなすには、データベース機能に対する専門知識も必要となる。ここでは、初心者にも容易に問合わせが行なえるように表型のユーザインタフェースを採用した。表型インタフェースの代表格としてQBE [3] がある。ここでの言語は、処理対象のデータとして項を許せるようにQBE を拡張したものになっている。

この表型の問合わせを、先に提案した項形式の問合わせに変換して知識ベースマシンへ問合わせを行う。

2. 推論マシンとのインタフェース (項型言語)

知識ベースマシンと推論マシンのインタフェースは、推論マシン上に関係型知識ベースの検索言語を関係論理に基づいた関係データベースの検索言語の拡張として設計した。関係型知識ベースモデルは、データベースにおける関係モデルを拡張したものであり、操作対象を単純なデータの集合から項の集合へと拡張したものである。

ここでの項型検索言語は、関係スキーマに格納された項関係に対する操作言語と位置付ける。この言語の記述能力として要求されることには、

- ①論理型言語から利用されるため、論理型言語と構文的に差がないこと。
- ②検索言語に含まれる基本演算の結果が論理型言語と等しいこと。
- ③単一化による項の操作と定義が記述できること。

などがある。

これらの要件を満たすインタフェースとしてretrieveという述語を考えた。retrieveは更新を含まない述語であり、一般的なシンタックスは

```
retrieve(Relation,Query,Status,Cost)
である。
```

report	No	Author	Title	Date
	"A1234"	member("A", ==)		Date
	P.No	member("B", ==)	substr("Prolog")	Date1>Date

図1 表型言語による問合わせの例

Relationには、結果となる一時関係を指定する。Queryには、検索を行う複数の項関係と条件を指定する。Statusには終了状態、Costには、処理に要した資源の情報が返される。以下に出現する例では、簡単のため、RelationとQuery のみを示すこととする。例には、次の関係スキーマを使う。

report(No, Author, Title, Date)

4属性の項関係reportは文献情報が格納されている。Noには文献番号が文字列で、Authorには著者名が文字列のリストで、Titleには題目が文字列で、Dateには論文の発行日がdate(年, 月, 日)の項の形(各引数は整数)で格納されている。

3. 表型のユーザインタフェース

表型言語の特徴は、

- ①使用される関係を表を用いて表現するため、視覚的に理解することができる。
 - ②関係の属性間の結合(関連付け)は、同じ変数を記入することによって行なえる。
 - ③表の欄を埋める(問合わせを作成)する順序に制限がないため、利用者の思考を妨げない。
- などである。

例を図1に示す。これは、『"A"さんの論文"A1234"よりも新しい"B"さんの論文で題目に"Prolog"というワードを含む論文の文献番号を求めよ』という問合わせである。

memberという関数はその属性に第1引数の要素が第2引数の条件で成立つかを調べるものである。substrという関

数はその属性に第1引数の文字列が出現するかを調べるものである。P.という関数(演算子)は、その引数を出力することを指定するものである。比較演算などの2項演算子では、左辺がその属性を示す。他と結合関係が無いときは省略可能である。また、部分項(属性の一部)に対する条件の設定を行う場合、必要な部分項までの枠を記述しその中に条件を設定を行う。例えば、『87年以前に発行された論文』を求めるためには、発行日の属性をdate(Year<87, __, __)と設定すればよい。

4. 表型言語から項型言語への変換

表型言語は、条件設定の順序に制限を与えていないため、より効率的な検索が行なえるように条件を配置していく必要がある。推論マシン側で行う最適化は静的に解析可能な範囲にとどめる。

作成された表から項型言語に変換するため、表の各行を次のような中間表現(副コマンド)にする。

```
subcommand( 項関係, 選択条件, 結合条件, メタ演算,
            結合変数, 出力変数)

図1の表は、次のように表現される。
subcommand( report("A1234",A1,__,Date),
            [member("A",A1,==)],[,],[Date],[,]),
subcommand( report(No,A2,T1,Date1),
            [member("B",A2,==),substr("Prolog",T1)],
            [Date1>Date],[,],[Date],[No]).
```

これらの中間表現からand結合、or結合を生成し、retrieveコマンドに変換する。and.or結合を区別するのに用いられる情報は変数だけである。これには、次のような規則を用いる。

- ①異なる副コマンドに同一名の変数が含まれているときは、and結合である。
- ②出力関係の指定がなく、異なった副コマンドに異なった出力変数が出現するときは、or結合である。
- ③出力関係の指定があるときは、その出力変数を含む関係はand結合である。
- ④各種の条件はand結合である。
- ⑤出力変数および結合変数を含まない関係は検索には無意味な指定であるため除去する。

つぎに項型言語への変換アルゴリズムを示す。

- ①出力変数を含む副コマンドを選択してくる。出力関係が指定されているときは、それらの出力変数を含むすべての副コマンドを選択してくる。
- ②それら副コマンドと同じ結合変数を含んでいる副コマンドを選択してくる。選択してきた副コマンドに対して新たに選択すべき副コマンドがなくなるまで②を繰り返す。(and結合の処理)

問合わせ:『論文数を第一著者ごとに求めよ』

report	No	Author	Title	Date
	P.C.Num	[P.g.F __]		

図2 集約演算を含んだ問合わせの例

- ③各副コマンドに含まれる選択条件は、その条件を適用する項関係の直後に位置させる。
- ④結合条件は、それらの変数を含むすべての関係が出現した後に位置させる。
- ⑤集約演算などのメタ演算を含む場合、出力側に指定を行う。
- ⑥他に出力変数を含む副コマンドが存在すれば、①から繰返す。(or結合の処理)

先の中間表現に上記の変換アルゴリズムを適用することによって、次のようなretrieveコマンドが生成される。

```
retrieve(output(No),
         (report("A1234",A1,__,Date), member("A",A1,==),
          report(No,A2,T1,Date1), member("B",A2,==),
          substr("Prolog",T1), Date1>Date) )
```

また、図2に示すように集約演算を含む場合には、次のようなコマンドが生成される。図中のg.はその属性によるグルーピングを指定しており、C.はその属性の集計を指定している。

```
retrieve(group(output(count(Num),F),[F],[==]),
         report(Num,[F|__],__,__))
```

5. おわりに

知識ベースマシンのエンドユーザインタフェースとしてQBEを拡張した表型の言語を設定した。そして、この言語から推論マシンと知識ベースマシンのインタフェースである項型言語に変換できることを示した。このことは、項型言語が強力な表現力を持ち、異種のユーザインタフェースに柔軟に対応できることを示している。また、現在は、表型言語のプロトタイプを完成した段階である。今後は、PS1の持っているマルチウインドウシステムを十分に生かし、よりユーザフレンドリーなインタフェースにして行きたいと考えている。

参考文献

- [1] 柴山、他、"知識ベースマシンMu-X", 信学技報 Vol.88, No.56, DE88-6. 電子情報通信学会, 1988
- [2] 物井、他、"知識ベースマシンMu-X (1)~項集合に対する検索言語", 第36回情報処理学会大会, 5E-4, 1988
- [3] Zloof, M.M., "Query-by-Example", IBM Research Report, RC4917, 1974