

知的データベース型エキスパートシェル

6J-2

－関係データベースとの融合－

本山 貴巳 岩崎 修 上田 謙一

松下電器産業株式会社 東京研究所

1. はじめに

関係データベース(RDB)に演繹機能を付加する方法としては、PROLOG や Lisp などのAI用のプログラミング言語とRDBを融合する方法[1][2]やエキスパートシステムと融合する方法[3][4]などが報告されている。

我々は、知識処理ツールIKE[5][6]とRDBを融合し、IKEからRDB上のデータを用いて、演繹を行えるようにした。

本稿では、IKEでの、RDB上のデータの知識表現、及び、RDB上のデータを用いての推論方式について述べる。

2. システム構成

システムの基本構成は、第1図の通りである。

IKEとRDBは、LANによって接続する。この方式は、データ転送などの通信に時間が掛かるが、以下の利点があるので採用した。

- ・RDB上のデータを変更せずに使用できる。
- ・RDBの検索能力を利用できる。
- ・分散処理の可能性がある。

また、RDBに対する問い合わせ言語としては、標準に成りつつある、SQLを用いている。

尚、IKEの基本構成については、参考文献[5]を参照されたい。

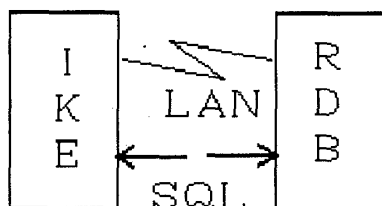


図1 基本構成

3. 知識表現

IKEでは、一階述語論理形式の知識表現を使用できる[5]。そこで、一階述語論理の述語とRDB上の関係表(仮想表も含め)との対応を取る方式を用いた[1][3]。

構文は、以下の通りである。

```
(DEF-MAPPING '述語名 '表名)
```

例)

表1の社員表と表2の部署表がRDB上にあり、IKEで、

```
(DEF-MAPPING '社員 '社員表)
```

```
(DEF-MAPPING '部署 '部署表)
```

を行うと、以降、社員と部署という述語で、RDB上の社員表と部署表を検索できる。

すなわち、後ろ向き推論で、

```
社員(101 田中 100)
```

を問い合わせると、真となる。

社員番号	氏名	部署番号
101	田中	100
...	...	...

表1 社員表

部署番号	部署名	住所
100	東研	川崎
...	...	...

表2 部署表

4. 推論方式

1) 前向き推論

前向き推論では、reteアルゴリズムを用いた。

RDBとreteアルゴリズムの融合方式は、参考文献[4]等に、報告されているが、我々は、以下の方式を用いた。

IKEのreteネットのnodeは述語となっている。例えば、

```
(def-rule (2 (if (and 社員(?x ?y ?z)
                     部署(?z ?w ?v))
                then 勤務地(?x ?y ?v)
            )))
```

というルールの前提部は図2のようなネットとなる。

そして、`root`からは述語を流す。

例えば、図2の`rete`ネットに

と

```

社員(101 田中 100)
    部署(100 東研 川崎)
  
```

の2つの述語を流すと以下のような動きをする。

`root`は、述語名を調べて対応する`node`に述語の引数を渡す。この場合、`node1`に

(101 田中 100)

`node2`に

(100 東研 川崎)

を渡す。`node1`では、変数 `?_1 ?_2 ?_3` にそれぞれ、101、田中、100を束縛した束縛リスト(`b-list`)を`node3`に渡す。`node2`では、変数 `?_4 ?_5 ?_6` にそれぞれ、100、東研、川崎を束縛した`b-list`を渡す。

`node3`では、`node1`、`node2`それぞれから渡された`b-list`の中から、`?_3`と`?_4`が同じものに束縛されている`b-list`の対を選び、その対から新しい`b-list`を作り`conflict set (c.s.)`に追加する。

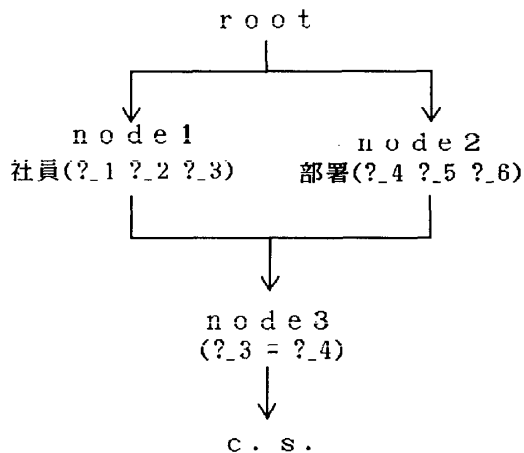


図2 `rete`ネット

この`rete`ネットにテーブル述語を流すために、`def-mapping`を行ったときに次のような述語を生成してそれを`rete`ネットに流す。

例えば、前ページの例の2つの`def-mapping`を行うと、

と

```

社員(@_1 @_2 @_3)
    部署(@_4 @_5 @_6)
  
```

という述語を作りこれらを`rete`ネットに流す。

但し、`@_1 @_2 @_3`は社員表に、`@_4 @_5 @_6`は部署表に対応付けられている特別な変数である。以後、このような変数をテーブル変数と呼ぶ。

`rete`ネットでは、テーブル変数は全てのものに

束縛できるものとみなして処理を行う。そして、`c.s.`に追加する前に、`b-list`から条件を生成し、その条件に合うタプルがあるかを調べ、対応するタプルがあれば、`c.s.`に追加する。

この方式の利点は、以下の通りである。

- (1) 一個の`instantiation`で複数の`fact`を扱える。
- (2) 一個の`instantiation`に対して一回の問い合わせですむ。

欠点は以下の通りである。

- (1) 表の変化を捕らえにくい。
- (2) 表のタプル1つ1つに対する操作がやりにくい。

しかし、これらの欠点に対しては、表操作の関数を用意することで容易に解決できると思われる。

2) 後ろ向き推論

後ろ向き推論では、非評価法を用いた[1]。

後ろ向き推論は`prolog`の様にプログラム言語として用いられるため、普通の述語と述語の間に、組み込み述語を記述できるようにしてある。このため、後ろ向き推論では、前向き推論と異なって、ルールの前提部の検索順序が重要になるので、非評価法によって、述語の検索のたびに条件に合うタプルがあるかどうかを調べる。

但し、バック・トラックがかかるごとに1つ1つデータを渡すのではなく、テーブル変数としてデータをまとめて取り扱う。

5. おわりに

テーブル変数を用いて関係表と述語を対応付ける方法はデータベースの検索回数を減らし前向き推論を高速に行える可能性がある。しかし、個々のデータに対する細かな操作がやりにくい。

今後、個々のデータを扱えるような関数を用意する必要がある。

6. 参考文献

- 1) 今中ほか: `Prolog`と関係データベースとの結合システム`DB-Prolog`, 情報処理学会論文誌, Vol. 28, No. 4, pp. 349-358 (1987).
- 2) 細井, 金崎: データベース管理システム`G-BESE`における`LISP`インタフェースの構成, 情報処理学会第35回全国大会予稿集, 7C-7
- 3) 松尾 正信: 知識ベースとデータベースの結合化システム, 昭和61年電気・情報関連学会連合大会講演文集, 32-2
- 4) 安次富, 黒瀬: `DM-Lisp`を用いた推論システム—アルゴリズム—, 情報処理学会第36回全国大会予稿集, 3R-10
- 5) 上田ほか: 知識処理ツール“`IKE`”(1)—システム概要—, 情報処理学会第36回全国大会予稿集, 3R-3
- 6) 間藤ほか: 知識処理ツール“`IKE`”(2)—知識表現の融合方式—, 情報処理学会第36回全国大会予稿集, 3R-4