

演繹データベースシステム

EX_DDBの応用支援環境

3K-2

嶺野 和夫** 中川 俊夫* 土井 晃一* 吉田 敦* 田中 英彦*
 (**富士通株式会社 *東京大学工学部)

1. はじめに

エキスパートシステムに代表される知識工学応用において、本格的な実用システムを構築する上での重要な課題の1つに大規模知識への対応が有る。我々が開発している演繹データベースシステムEX_DDBはその様な大規模知識を想定した知識ベースマシンを応用、言語、実行方式、マシンアーキテクチャの各面からめざすものであり、現在C言語で記述された研究用システムが稼働している[1]。本稿では、応用面として、演繹データベースにおけるエキスパートシステムの構築方法を検討し、これを支援する為の応用支援環境としてエディタ(EX_DDB/KED)とデバッガ(EX_DDB/DBG)を紹介する。

2. 応用面の基本方針

EX_DDBは関係データベース(以下RDB)の自然な拡張として開発されており、その応用においてもRDBの利点を最大限活用する事を考える。具体的には既存のRDBからのファクト型知識コンバートや、関係、正規性、キー、従属性等を利用したファクト型知識の管理とそれにより獲得されたファクト型知識を基にしたルール記述等である。最終的にはこれらに基づいて、演繹データベース(以下DDB)の特徴を活かし、現存システムと協調でき、汎用性/拡張性の有る応用システム構築方法を確立する事が目標である。

3. エキスパートシステム構築方法

DDB上にエキスパートシステムを構築する場合の形態として、①構築ツールと考えて直接知識を記述する方法と、②AI指向言語と考えてエキスパートシステムシェルを記述しその上に知識を記述する方法が有る。

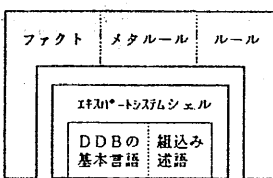
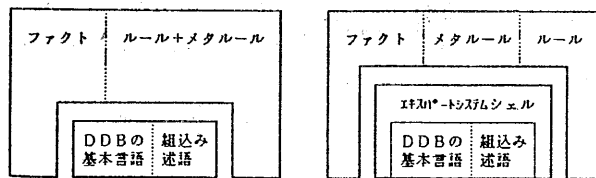


図1 ①ツールとしての応用 図2 ②言語としての応用

前者はDDBの特徴が表面に出ており、その利点の活用が容易である反面、推論制御等の応用記述面における自由度が小さい性質を持つ。後者はそれと反対の性質を持つ。我々はDDBの性質の活用を重視して前者を採用した。

現在想定しているエキスパートシステム構築手順を次に示す。

- ①関係表現が可能なファクトについて関係スキーマを設計↓し、従属関係、制約条件、同値性と共に宣言する。
 ↓(可能な限り関係は正規性を取る様にする:この時点では関係スキーマに対応する関係の枠組みが生成される。)
 - ②関係表現が不可能なファクトについて述語論理型言語で↓個別に記述する。
 - ③宣言されたファクトのスキーマを参照しながらルールを↓記述する。
 - ④用意されている関係にファクトを入力する。この時、既↓に他のRDBに情報が存在するならば、コンバート機能↓によってファクト型知識を獲得する。
 - ⑤運用・デバッグ
- 以上①~⑤を繰り返す事によって段階的に知識ベースを成長させる。

この方法を採用する事により期待される効果を次に述べる。
 ●ファクトのスキーマを参照しながらルールを記述する事で記述された知識の意味を理解する事が容易になり、述語名や引数等のアトム単位での矛盾を防止する。
 ●ルールを記述する時点では、スキーマのみが必要で、実際のファクトを必ずしも必要としない。この為本質的にファクトが後から生成される設計型等のシステム構築が容易になる。
 ●スキーマを既存RDBの関係と同様な設計にする事により、既存のRDBとの対応が更に容易になる。
 ●正規性の導入により、知識の汎用性が向上する。また、不必要な概念へのアクセスを削減する。
 以下にこの構築方法をサポートする主な機能を説明する。

4. 知識コンバート機能

RDBとDDBの親和性が良い事は自明な事項であり、RDBはDDBに理論上、包含される。具体的にはRDBのタプルとDDBのファクト型リテラルが1対1で対応している。この性質により、RDBからDDBにファクト型知識としてデータをコンバートし、膨大な知識獲得を効率良く行なうという事はごく自然な発想である。EX_DDBはファクトの他に、その管理情報もコンバートする機能を持つ。その意味でEX_DDBはRDBを完全に包含していると言える。EX_DDBは内部に定義管理テーブルと呼ぶ一種の知識辞書を持ち、推論に使用可能な知識は全てこの知識辞書で管理する。関係スキーマや知識の種類等

Application Supporting Environment of Deductive Database System EX_DDB

Kazuo MINENO**, Toshio NAKAGAWA*, Kouichi DOI*, Atsushi YOSHIDA*, Hidehiko TANAKA*

(**FUJITSU LIMITED, *Faculty of Engineering, The University of Tokyo)

の情報もこの辞書に格納される。知識コンバータは図3の様に、RDBファイルとDDBファイル双方へのI/O関数を持ち、RDBファイルから読んだ情報は知識辞書に登録した後でDDBファイルに書き出される。また、DDBファイルはファクト型知識のファイルに限って、そのままRDBファイルにコンバートされる。RDBからDDBへのコンバートの際、既に同スキーマの知識ファイルがDDBに存在する場合には、知識ファイルの更新が行なわれる。この事によってRDBからDDBのファクトを更新する事が可能になり、運用面においても両者の協調が実現される。EX_DDBは内部関数を利用してRDB機能を実現した関係データベースシステム (EX_DDB/RDB) を持つ。

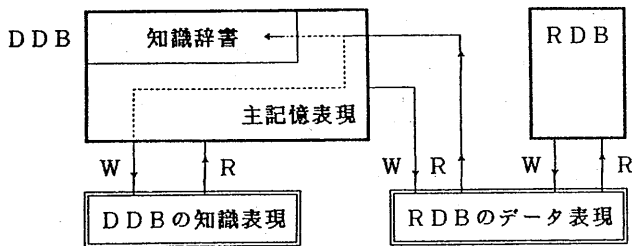


図3. DDB⇔RDB双方向コンバータ

5. 知識管理機能

EX_DDBは自らの推論によって知識を管理する機能を知識の形で実現する。ファクト型知識に対する従属関係、制約条件、同等性、キー、スキーマ等をファクトの形で持ち、それを利用した知識管理機能をメタルールとして用意する。利用者はメタルールに対して質問を行なう事により各管理機能を利用でき、機能の拡張も容易である。現在は正規性の検定用ルールが実装されている。管理用知識のスキーマとルールの一部を図4に示す。

```

◎管理用ファクト (スキーマ)
スキーマ (関係名、属性数、(属性名))。
関数従属 (関係名、属性数、(従属先の属性名)、(従属元の属性名))。
多値従属 (関係名、属性数、(従属先の属性名)、(従属元の属性名))。
キー (関係名、属性数、(キー属性名))。
制約 (関係名、属性数、属性名、制約値)。
同等 (関係名1、属性数1、属性名1、関係名2、属性数2、属性名2)。

◎第3正規性の検定用ルール
nf3_check (R, N) :- 関数従属 (R, N, __決定項, __従属元),
                    キー (R, N, __キー属性),
                    multi_member (__決定項, キー属性),
                    multi_member (( ), __)。
multi_member ( ( H | T ), Y ) :- member (H, Y),
                                multi_member (T, Y)。
member (H, ( H | T ))。
member (X, ( Y | T )) :- member (X, T)。

```

図4. 管理用知識

6. 知識エディタ (EX_DDB/KED)

知識エディタの主な機能を次に示す。[3]

- ①RDBとのファクト型データ相互コンバート
- ②定義ファイル操作
- ③定義編集/参照
- ④関係スキーマ参照

7. デバッガ (EX_DDB/DBG)

デバッガの主な機能を次に示す。[3]

- ①推論経過のログ情報蓄積と解析
 - ②ブレークポイントの設定/解除
 - ③中間解、束縛情報等の参照
 - ④ステップ動作
- デバッガは言語 (ROTHORN) の組み込み述語として実行させる。また、推論経過のログ情報はエキスパートシステムにおける説明機能にも利用される。

8. 考察と今後の課題

以上、RDBとDDBの親和性を積極的に利用したDDB上での応用記述方法を検討し、それを支援する環境について紹介した。現在この環境の元で、評価用の例題としてエキスパートシステム (パーソナルコンピュータ構成支援システム) の試作を行っている。現時点ではファクト型知識の取り扱いを規定した段階であり、現在、例題の構築を通じて、制御戦略、ルール管理等のルール型知識について検討中である。それに伴って、管理用知識や言語機能[2]の充実等も含めた環境の整備が望まれる。またEX_DDBは実行方式上[4]、全解探索を実行する為に、応用によっては導出された全解の中から1個の解を選択する必要が生じる。この解決策として、CF値の様な不確実性を導入して最良の解を選択する方法がある。具体的には、各リテラルの引数の1つに値を割り当て、処理系によってこの値を自動的に更新する方法が考えられる。

```

cxddb
===== EX_DDB メインメニュー =====
1.....推論/言語 (ROTHORN)
2.....関係データベース (EX_DDB/RDB)
3.....知識エディタ (EX_DDB/KED)
4.....
5.....
6.....
7.....
8.....
9.....システムコール
10.....ファイルコンバータ DDB to RDB
11.....ファイルコンバータ RDB to DDB
12.....
13.....
14.....
15.....
16.....
17.....
18.....
19.....試験モード ON(1)/OFF(0) --> now[0]
20.....関数テスト
99.....作業終了
=====
番号で選択して下さい。 -->3

```

図5. EX_DDBのメニュー画面

```

type 'h' for help
edit command -->h
===== EX_DDB/KED 知識エディタメニュー =====
at [tnum].....タプル追加          atz.....先頭にタプル追加
cpt [tnum].....タプル複製          atz.....最後にタプル追加
cv [tnum] [vnum].....タプル変更    cds [status].....定義変更
delt [tnum].....タプル削除          deld [filename].....定義ファイル削除
h.....ヘルプ表示
j.....関係の属性表示
ld.....全定義ファイルリスト表示  全定義数 [ 3 ]
lfd.....全定義中の定義ファイル表示  作業定義数 [ 1 ]
lwd.....作業中の定義ファイル表示
ld.....作業定義スキーマ表示
lt [tnum].....作業対象タプル表示  ltz.....最終タプル表示
lv.....作業対象属性表示
new.....新関係生成
q.....終了
rd.....ファイルからの定義読み込み
seld [defNo].....作業定義選択: 現在=[h4_要求]
scld [tnum].....作業タプル選択: 現在=[1]
selv [vnum].....作業属性値選択: 現在=[head]
sl.....定義名検索
ss [f/w].....文字列検索 [file / memory]
ud.....ファイルへの定義書き込み
===== 関係名 [ h4_要求 ] ===== 属性数 [ 2 ] ; タプル数 [ 1 ] =====
edit command -->new
新定義名は? (取消->'')-->c

```

図6. 知識エディタのメニュー画面

参考文献

- [1]吉田、中川、土井、嶺野、田中:「演繹データベースシステムEX_DDBの概要」第34回情処全国大会, 3K-1, 1987.
- [2]吉田、中川、土井、嶺野、田中:「演繹データベースEX_DDB上の言語ROTHORN」第34回情処全国大会, 3K-3, 1987.
- [3]吉田、嶺野、田中:「大規模知識処理支援環境EX_DDB」信学技報 Vol.86 No.264, A186-33, 1986.
- [4]吉田、土井、嶺野、田中:「演繹データベースにおける推論実行方式の評価」信学技報Vol.86 No.204, A186-24, 1986