

拡張項と階層構造に基づいた演繹データベースの試作

2C-7

高橋千恵 横田一正[†] 横塚実 梶山拓哉^{††}(財)日本情報処理開発協会 (財)新世代コンピュータ技術開発機構[†](株)アーティフィシヤル・インテリジェンス^{††}

1 はじめに

知識ベースへの有力なアプローチの1つとして、演繹データベース (DDB) がある。しかし、1階項/確定節で表わされるルールは、さまざまな応用で使用する実体を表現するにはレベルが低い。それがDDBは応用がないと言われる原因の一つにもなっている。DDBの拡張としては以下の2つを考えている。1つは項関係を拡張することにより、複合オブジェクト、継承関係、オブジェクト指向概念等を組み込むことであり、もう1つは、DDBを他のパラダイムに組み込むことによって複数のDDBを扱うことである。この2つは直交している概念で組み合わせ可能である。

我々は、項表現の拡張 (CRL^[1]) によるDDBと、実体の(振る舞いを含む)素直なモデリングが行えるオブジェクト(指向)概念にDDBを組み込む事を考えている。このアプローチに沿って、ICOTで開発した逐次型推論マシンPSI上でDDBの拡張をおこなっている。

まず項表現の拡張として、非正規関係に対応したDDBを開発した。外延データベース (EDB) は、非正規関係DBMSのKappaに格納され、項表現/ルールはCRLで記述される。

次に複数DDBを扱うために、階層概念を導入し、それにより知識をモジュールとして扱うDDBの拡張を考えている。以下に、その概要を述べる。

2 項表現の拡張による演繹データベース

2.1 CRL項の構文と制限

我々は、非正規関係を意識したCRLという論理型言語を考えた。CRL項は以下のように定義される。

<項> ::= <定数>
 | <変数>
 | <proper>
<定数> ::= 整数 | 文字列 | アトム
<proper> ::= <属性名> / <定数>
 | <属性名> / <変数>
 | <属性名> / {<定数>, ..., <定数>}
 | <属性名> / [<proper>, ..., <proper>]
<属性名> ::= アトム

<proper> はそれぞれ上から定数項、変数項、集合項、組項と呼ぶ。

A Prototype System of Deductive Databases Based on Extended Terms and Hierarchical Structure

C. Takahashi K. Yokota[†] M. Yokotuka T. Kajiyama^{††}
JIPDEC ICOT[†] AIC^{††}

組項には以下の制限を置く。項 t に含まれる属性名の集合を $attr(t)$ とすると、組項が $a/[t_1, \dots, t_n]$ の時、

$$\begin{aligned} \forall i, j, i \neq j \cap attr(t_i) \cap attr(t_j) &= \phi \\ \forall i, a \in attr(t_i) \end{aligned}$$

を満足しなければならない。

CRLでは、<proper>は次の様に書かれる。

定数項: $system/“KAPPA”$

変数項: $name/X$

集合項: $sports/\{ski, tennis\}$

組項: $person/[name/[first/evaniste, last/galois], born/1811]$

また、ルールについても、

$$\begin{aligned} anc/[des/X, anc/Y] : -parent/[chi/X, par/Y]. \\ anc/[des/X, anc/Y] : -parent/[chi/X, par/Z], \\ \quad \quad \quad anc/[des/Z, anc/Y]. \end{aligned}$$

の様に書くことができる。

2.2 CRL項の単一化

CRLでは、独自の単一化アルゴリズムを用いている。CRL項には内部表現としてタグが付けられる。項とタグの対応を以下に示す。

<定数>、定数項、集合項	$\Leftrightarrow$	set
<変数>、変数項	$\Leftrightarrow$	var
組項	$\Leftrightarrow$	tup

内部表現として付けられているタグによって単一化の処理は下記のように分類される。

入力タグ	出力タグ	その他制約
(var, var) :	var	制約 ($var = var$)
$(var, set), (set, var)$:	set	制約 ($var = set$)
$(var, tup), (tup, var)$:	tup	制約 ($var = tup$)
(set, set) :	set	$set \cap set (\neq \phi)$
$(set, tup), (tup, set)$:	—	—
(tup, tup) :	tup	下記参照

集合項どうしの単一化では、2つの集合のintersectionを取り、その結果が空集合でないものが単一化の結果となる。例えば、

$$T_1 = sports/\{ski, baseball\}$$

$$T_2 = sports/\{tennis, riding, baseball\}$$

の時、 $T_1 = T_2$ の単一化の結果は、 $sports/\{baseball\}$ となる。

組項どうしの単一化では、まずトップの属性名が等しいかどうか調べる。次に、組要素の項で属性名の等しいものどうしが単一化され、等しい属性名がないものは結果に出力される。例えば、組項の単一化は次のようになる。

```
T1 = hobby/[sports/{tennis,cycling,basketball},
             books/{novel,mystery,poem}]
T2 = hobby/[books/{commic,fiction,novel},
             alcohol/{wiskey,wine,sake}]
```

の時、この単一化の結果は、

```
hobby/[alcohol/{wiskey,wine,sake},
        books/{novel},
        sports/{tennis,cycling,basketball}]
```

単一化の結果、 $T1$ に対しては *alcohol* 属性が、 $T2$ に対しては *sports* 属性がそれぞれ付加されている。

2.3 CRL 演繹データベース

Kappa の非正規関係は表現志向の意味論を持っており、CRL-DDB の意味もこれに対応している。つまり、集合項は効率を良くするための表現なので、CRL-DDB は確定節 DDB に対応させることが出来る。したがって、CRL-DDB の質問処理にも確定節 DDB での処理アルゴリズムが容易に適用可能である。

簡単にその処理概要を説明する。問い合わせの入力に対して、内包データベース (IDB) から問い合わせを処理するのに必要なルールだけを取りだし、clique を作成する。これ以降の処理は、この clique に対して実行される。問い合わせの束縛情報を使い、横方情報伝達 (sideways information passing)^[2] により、ルールの本体の順序を決定する。次に、再帰問い合わせを効率良く処理するために問い合わせのあるルールを一般マジック集合 (generalized magic sets)^[3] でプログラム変換する。

例えば、問い合わせ $anc/[des/taro, anc/X]$ が与えられると、上記のルールは以下のようにプログラム変換される。

```
magica[mag/taro].
magica[mag/Y] : -magica[mag/X], parent/[chi/X, par/Y].
anc/[des/X, anc/Y] : -magica[mag/X], parent/[chi/X, par/Y].
anc/[des/X, anc/Y] : -magica[mag/X], parent/[chi/X, par/Z],
                    anc/[des/Z, anc/Y].
```

又、導出述語間の評価順序に従って、同値関係の概念を使い、ルール群を評価単位 (component) からなる依存グラフに展開している。評価は、依存グラフから極小元の評価単位を順次取り出し、セミナイーブ評価法 (semi-naive evaluation)^[4] でボトムアップ評価を行っている。

EDB 項の評価は、CRL 項から拡張関係代数 (射影、結合、選択) 演算を作成し、非正規形データベースをアクセスすることによっておこなっている。確定節 DDB と異なり、EDB のタプル数を減少させることが出来、効率的である。

3 階層構造による演繹データベースの拡張

演繹 + オブジェクト指向データベースへのアプローチとして、階層概念に DDB を組み込む方法について検討している。その中の 2 つについて概略を述べる。

3.1 階層構造の導入

実世界の知識を見ると、知識はバラバラなものではなく、分類、モジュール化されている。そのモジュール化された知識が互いの関係を意識しながら問題解決や質問処理にあたる。これらの特徴を使って DDB を拡張する^[5]。

モジュール化された知識の単位を「世界」=DDB と考え、半順序関係を持った DDB の集合を「宇宙」とする。この関係に、知識の単調増加と洗練化を考えている。これらによって、知識の効率的表現 / 効率的管理が可能となる。また、特定世界を意識した質問や特定の知識を持った世界を求める質問、さらに、もしこんな継承があったら、こんなルールを追加したら等という仮説的質問を行うことができる。これが階層構造を持った DDB である。この DDB の意味論としては、各「世界」に独立した DDB を対応させ、「宇宙」自体にも 1 つの DDB を対応させている。

3.2 分散化 / 並列化

上記の処理系として各「世界」に自律的な処理機能を持たせ、メッセージ通信による質問処理を考えている。つまり各「世界」が独立に働き、メッセージ通信によって他「世界」と協調して結果を求める。これは、各「世界」を動的オブジェクトとも考えることに対応している。また、ICOT で研究開発している並列推論マシン / 並列知識処理とも対応している。さらに、分散演繹データベースにも適用できよう。

実際の実装は 2 段階の開発ステップを考えている。まず試作システム (CRL による DDB) を拡張して階層構造を導入する。その次に、分散 / 並列処理を実現した DDB の開発を行うことを検討している。

4 今後

応用としては、証明支援システム (「世界」に分割した DDB が使える) や専門家システムを考えている。さらに、ICOT の他プロジェクトで研究開発されている構造データサポート用の論理型言語との統合も考えている。

参考文献

- [1] K.Yokota, "Deductive Approach for Nested Relations", in *Programming of Future Generation Computers II*, K.Fuchi and L.Kott, eds., North-Holland, 1988.
- [2] J.D.Ullman, "Implementation of Logic Query Languages for Databases", *ACM TODS*, vol.10, no.3, 1985
- [3] C.Beeri and R.Ramakrishnan, "On the Power of Magic", *ACM PODS*, 1987
- [4] F.Bancilhon, "Naive Evaluation of Recursively Defined Relations", in *On Knowledge Base Management Systems*, M.L.Brodie, et al, eds., Springer, 1986
- [5] 横田一正, "階層構造をもった演繹データベース", ICOT 内部メモ, 1989