

ICOT Technical Memorandum: TM-0764他

TM-0764他

第39回情報処理学会全国大会論文集
(4研関連)

July, 1989

©1989, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191-5
Telex ICOT J32964

Institute for New Generation Computer Technology

- TM-0764 拡張項と階層構造に基づいた演繹データベースの試作
高橋千恵, 横塚 実, 梶山拓哉, 横田一正
- TM-0766 並列推論マシンPIM/k ハードウェアアーキテクチャ
浅野滋博
- TM-0767 Multi-PSI/V2における一括GCおよびメモリ消費特性の評価
宮崎信仁(著), 川田易治(SEI), 中島克人
- TM-0768 分散環境における構造体管理
六沢一昭, 近山 隆, 中島 浩, 瀧 和男,
市吉伸行, 中島克人, 稲村 雄
- TM-0769 Kappaデータベース機能の検査システム
吉武 淳, 金枝上敦史, 佐藤靖彦, 平岡公一,
河村元夫
- TM-0770 MRB-GCの評価—アクティブセル量とヒープ消費—
武井則雄, 半野喜芳, 西崎慎一郎, 森田京子,
木村康則, 近山 隆
- TM-0775 PIMOSの評価
佐藤裕幸, 星田昌紀, 近山 隆, 藤瀬哲郎
- TM-0779 並列推論マシンPIM/pの要素プロセッサにおける分岐機能の高速化のためのアーキテクチャ
篠木 剛, 久門耕一, 安里 彰, 服部 彰,
近山 隆, 後藤厚宏
- TM-0798 共有メモリ結合マルチプロセッサにおけるKL1向き
ガーベジコレクション
今井 明, 川合英夫, 中川貴之, 中島克人,
後藤厚宏
- TM-0799 PIMOSのセルフ・コンパイラ
佐藤令子, 堀 敦史, 関田大吾, 近山 隆

拡張項と階層構造に基づいた演繹データベースの試作

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)アーティフィシャル・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一つとして、演繹データベースがある。しかし、1階項／確定節で表わされるルールは、知識表現としては低レベルであり、それが演繹データベースは応用がないと言われる原因の一つにもなっている。考えられる知識表現の拡張としては、項表現の拡張、継承関係の導入等がある。又、より柔軟な知識の構成として、知識のモジュール化／体系化が考えられ、複数の知識が協調して問題解決を行うとも考えられる。演繹データベースをより強力にするために、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)概念と演繹データベースの統合(融合)も考えられている。1つは演繹データベースにオブジェクト(指向)概念を組み込む方法がある。もう1つは、演繹データベースを(オブジェクト指向概念を含む)他のパラダイムに組み込む事である。これによって、DDBの拡張をおこなっている。

我々はまず項表現を拡張した演繹データベースをICOTで開発した逐次型推論マシン PSI 上で開発した。外延データベースは、繰り返しの構造を持つ可変長レコードをデータ形式の基本とした非正規形データベース管理システム Kappa に格納している。

次に、上記のシステムに階層概念を導入し、知識をモジュール化する事によって演繹データベース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、非正規形を意識した CRL[1] 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。
```

C. Takahashi (K. Yokota M. Yokotuka T. Kajiyama
 JIPDEC ICOT AIC

項に含まれる属性名の集合を attr(項) とすると、属性名 / [$<\text{proper}>_1, \dots, <\text{proper}>_n$] の時、 $i \neq j$ ならば、
 $\text{attr}(<\text{proper}>_i) \cap \text{attr}(<\text{proper}>_j) = \emptyset$
 任意の i に対して、属性名 $\in \text{attr}(<\text{proper}>_i)$ の制限を置く。

CRL では、 $<\text{proper}>$ は次の様に書かれる。
 定数項: system/"KAPPA" 变数項: name/X
 集合項: sports/{ski, tennis}
 組項:
 person/[name/[first/evarie, last/galois], born/1811]

また、ルールについても、
 $\text{anc}[\text{child}/X, \text{parent}/Y] : -\text{father}[\text{child}/X, \text{papa}/Y]$.
 $\text{anc}[\text{child}/X, \text{parent}/Y] : -\text{mother}[\text{child}/X, \text{mama}/Y]$.
 $\text{anc}[\text{child}/X, \text{parent}/Y] : -\text{anc}[\text{child}/X, \text{parent}/Z]$,
 $\text{anc}[\text{child}/Z, \text{parent}/Y]$.
 の様に書くことができる。

2.2 CRL 項の单一化

CRL では、独自の单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	↔	set
<変数>、変数項	↔	var
組項	↔	tup

内部表現として付けられているタグによって单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 \wedge set ($\neq \emptyset$)
(set, tup), (tup, set):	-	-
(tup, tup):	tup	下記参照

集合項どうしの单一化では、2つの集合の intersection を取り、その結果が空集合でないものが单一化され、intersection が導出される。例えは、

T1 = sports/{ski, baseball}
 T2 = sports/{tennis, riding, baseball} の時、T1 = T2 とすると、
 $T1 = T2 = \text{sports}/\{\text{baseball}\}$ となる。

組項どうしの单一化では、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そのまま導出される。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/{whiskey,wine,sake}] の時、  
  
T1 = T2 とすると、  
T1 = T2 = hobby/[alcohol/{whiskey,wine,sake},  
                  books/{novel},  
                  sports/{tennis,cycling,basketball}]
```

単一化の結果、 T_1 に対しては $alcohol$ 属性が、 T_2 に対しては $sports$ 属性がそれぞれ導出されている。

2.3 CRL 演繹データベース

簡単に CRL 演繹データベースの処理概要を説明する。問い合わせの入力に対して、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用い、横方情報伝達 (side-ways 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に関係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。

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、セミナイン評価法 (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。

3 階層演繹データベース

階層概念を導入し、知識をモジュール化する事によって演繹データベースを拡張する [5]。以下に、その概要を述べる。

3.1 「世界」と「宇宙」

演繹データベースの集合を階層構造に埋め込み、個々の演繹データベースを「世界」、その全体を「宇宙」とする。

演繹データベースの集合を考える。それぞれの要素は互いの識別のため、識別子を持っている。この各要素を「世界」と呼ぶ。そして、互いに同一の識別子をもたない「世界」の集合に半順序関係を導入したものを「宇宙」と呼ぶ。

「世界」の意味は、「宇宙」における半順序関係を外延階層 (extent hierarchy) とし、上位の「世界」の全てのルールを継承した演繹データベースに対応していると考える。

「宇宙」の意味は、その「宇宙」に含まれている各「世界」に対し、上位の「世界」の全てのルールを継承した演繹データベースを考え、その和集合とする。「宇宙」の中では、半順序関係にない別の「世界」どうしには、

相反したルールが干渉し合うことなく存在する事を許される。したがって、「宇宙」はさまざまな異種の「知識」が共存することが出来る。これが階層を持った演繹データベースである。

3.2 「宇宙」に対する質問

「宇宙」に対する質問は、「宇宙」の動的な再構成機能と、複数の「世界」に対する問い合わせ機能を含む。質問は以下の 3 つの情報と回答のための引数からなっている。

- 質問対象とする「世界」の識別子の集合、あるいは変数
- 質問対象とする「世界」に対するゴール (述語の集合)
- 「宇宙」に対する更新情報 (「世界」ごとの半順序と規則の追加情報)

質問によって与えられる「更新情報」は、データベースに対する更新と異なり、通常の演繹データベースと同様に問い合わせの拡張として位置づけることができる。

質問の種類は次の 4 つになる。

- 特定の「世界」に対する質問
- 「世界」を求める質問
- 他「世界」の継承の場合の質問
- 「世界」に知識を追加した場合の質問

4 分散 / 並列演繹データベース

知識ベースはデータベースと異なり、単一化など本質的に計算量の大きな操作を基にしているので、量の問題ははるかに深刻である。分散 / 並列処理は考えなければならない 1 つの課題である。我々はそれらを意識し、「宇宙」と各「世界」をそれぞれ独立したプロセス (自律的な管理機能を持つ) と考える。

5 今後

階層構造を持った演繹データベースの実現に対して、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 による演繹データベース) を拡張して階層構造を導入する。その後に、分散 / 並列処理を実現した演繹データベースの開発を行うことを検討している。

参考文献

- [1] K.Yokota,"Deductive Approach for Nested Relations", in *Programming of Future Generation Computers II*, eds. by K.Fuchi and L.Kott, North-Holland, 1988.
- [2] J.D.Ullman, "Implementation of Logic Query languages for Databases", *ACM TODS*, vol.10, no.3, 1985
- [3] C.Berri and R.Ramakrishnam, "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, "階層構造をもつた演繹データベース", ICOT 内部メモ, 1989