7F−5

関係代数による高次論理型データベース ALDB の問合せ手続き

大原 剛三　井戸 譲治　馬場口 登　北橋 忠宏

大阪大学産業科学研究所

1 はじめに

演繹データベース DDB(Deductive DB)[1] は, 常に正しい完全な知識を対象にしている. ところが, 現実世界では完全な知識よりも例外に代表されるような常には正しくない不完全な知識の方が多い. 筆者らは,DDB を不完全な知識をも取り扱えるように拡張した高次論理型データベース ALDB(Advanced Logical DB) の開発を進めている [2]. 本稿では, 従来の反駁証明による問合せ手続き [3] における, データベースとしての効率性と操作性という問題点を解決すべく, 集合演算である関係代数を用いた問合せ手続きを提案する.

2 ALDB の枠組

ALDB では, 完全な知識, 不完全な知識, 事実, 制約を次のように論理式の形式で表現している. ここで制約とは, 述語間の矛盾を定義するものであり, ⊥は恒偽命題を表す記号である. また, 不完全ルール中の ⇐は例外許容型のオペレータである.

完全ルール　$A \leftarrow B_1, \cdots, B_n$

「$B_1, \cdots, B_n$ ならば必ず A である」

不完全ルール　$A \Leftarrow B_1, \cdots, B_n$

「$B_1, \cdots, B_n$ ならば通常 A である」

事実　$p(t_1, \cdots, t_n)$

「$t_1, \cdots, t_n$ は p である」

制約　$\perp \leftarrow C_1, \cdots, C_n$

「$C_1, \cdots, C_n$ が同時に成り立つことは矛盾する」

A, B_i, C_i は正リテラル, p は述語, t_i は定数である. 尚, 不完全ルールの意味論は現在, 自己認識論理 [4] の意味論に従っている. また, 完全ルールの関係代数による問合せは,DDB の分野で明確化されているので, ここでは不完全ルールに議論を絞る.

Query Procedure of Advanced Logical Database by Relational Algebra,
Kouzou OHARA,Jouji IDO,Noboru BABAGUCHI,Tadahiro KITAHASHI,
The Institute of Scientific and Industrial Research,Osaka University

3 問合せ手続き

3.1 否定を表す述語と例外を表す述語

ALDB では述語 A に対して A の否定を表すために $notA$ という述語を用いている. ヘッドに $notA$ をもつルールは, 通常, 述語 A の否定, つまり「A でないもの」を定義するが,A をヘッドとするルールの例外, すなわち「A の例外」を定義している場合もある. このようなルールにおける例外と否定を明確に区別するために, ここでは exc という記号を導入し, 述語 A の例外を $exc\text{-}A$ と表記する. また,$exc\text{-}A$ の例外 (A の例外の例外) は $exc^2\text{-}A$ と表記し, 一般に $exc^n\text{-}A$ の例外を $exc^{n+1}\text{-}A$ と表し,n を次数と呼ぶことにする. よって, この表記における exc の次数は, ルールの階層を表し,exc の次数が高いほど特殊なルールとなるわけである. 以下に exc の表記規則をまとめておく. 但し,$exc^n\text{-}A$ $(n = 1, 2, \cdots)$ は1つの述語として扱う.

$$exc^0\text{-}A \triangleq A$$
$$exc^1\text{-}A \triangleq exc\text{-}A$$
$$exc^n\text{-}A \triangleq exc(exc^{n-1}\text{-}A) \qquad (n = 1, 2, \cdots)$$

3.2 不完全ルールの関係代数への変換法

不完全ルールについて, ここでは, 自己認識論理の意味論から離れもっと直観的な意味を考えることにする. 不完全ルールの直観的な意味とは,「そのルールを完全ルールとみなした時の結果から例外となるものを除いたもの」といえる. 次に挙げる変換方法は, この解釈に沿うように構成されている. また, 以下では規則中には再帰は現れないものとし,exc^n は既にルールに付加されているものとする.

定義３．１（不完全ルールの変換）

不完全ルール

$A(\alpha) \Leftarrow B_1, \cdots, B_n$　　(α : 変数の並び)

を関係代数演算に変換すると,

$$A(\alpha) = \pi_\alpha(\sigma_\theta(B_1 \bowtie \cdots \bowtie B_n)) - exc\text{-}A(\alpha)$$

となる. ここで,$\pi, \bowtie, \sigma$ はそれぞれ関係代数演算の射影 (projection), 自然結合 (natural join), 選択 (selection)

を表す演算子であり,θは選択の条件を表す式である.

ここではまず,$B_1 \bowtie \cdots \bowtie B_n$により関係$B_1, \cdots, B_n$の自然結合をとり,これらの関係すべてを満たす組を計算する.そして,これらの限定した組から条件θによる選択σ_θにより,リテラル中に現れる定数を含む組を選び出し,αへの射影π_αによりαに現れない変数を取り除いている.つまり,$\pi_\alpha(\sigma_\theta(B_1 \bowtie \cdots \bowtie B_n))$ の部分がこのルールを完全ルールとみなした時の結果を表し,これと例外を表す$exc\text{-}A(\alpha)$ の集合差をとっていることになる.さらに,例外を定義する例外ルールは以下のようになる.

定義3.2 (例外ルールの変換)

不完全な例外ルール

$exc^n\text{-}A(\alpha) \Leftarrow B_1, \cdots, B_n$

は次のように関係代数演算に変換される.

$exc^n\text{-}A(\alpha) = \pi_\alpha(\sigma_\theta(B_1 \bowtie \cdots \bowtie B_n)) - exc^{n+1}\text{-}A(\alpha)$

また,$exc^n\text{-}A$ をヘッドとするルールについては,ルール中に既に存在するものだけでなく,次の変換により制約の式から得られたものも考慮しなければならない.

定義3.3 (制約の変換)

制約

$\perp \leftarrow exc^{n-1}\text{-}A(\alpha), C_1, \cdots, C_m$

は次のように関係代数演算に変換される.

$exc^n\text{-}A(\alpha) = \pi_\alpha(\sigma_\theta(C_1 \bowtie \cdots \bowtie C_m))$

3.3 ルールの評価法

ルールの変換後,関係Aの集合を求める場合,ルールのヘッドにおいて最も次数の高い$exc^n\text{-}A$を持つルール,すなわち最も特殊なルールから評価していく.この理由は,Aが不完全ルールのヘッドであるとき,関係Aを求めるには$exc\text{-}A$が必要であり,同様に関係$exc^n\text{-}A$を求めるには関係$exc^{n+1}\text{-}A$が必要であるからである.

いまnをルールのヘッドにおけるexcの最高次数とすると定義3.2より,次の変換式が存在することがある.

$exc^n\text{-}A(\alpha) = \pi_\alpha(\sigma_\theta(B_1 \bowtie \cdots \bowtie B_n)) - exc^{n+1}\text{-}A(\alpha)$

しかし,nが最高次数なので$exc^{n+1}\text{-}A(\alpha)$をヘッドとするルールは存在せず,この時点では,このルールは完全ルールと同様に扱える.以後,$exc^{n-1}\text{-}A, exc^{n-2}\text{-}A, \cdots, A$の順にボトムアップに評価していく.$A$をヘッドとするルールが複数ある場合は,それらの評価結果の集合和をとる.

4 具体例

次のような規則,事実を考える.

$notfly(x) \Leftarrow animal(x)$
$animal(x) \leftarrow bird(x)$
$exc\text{-}notfly(x) \Leftarrow bird(x)$
$bird(x) \leftarrow penguin(x)$
$exc^2\text{-}notfly(x) \Leftarrow penguin(x)$
$penguin(Tweety)$

これらを前述の定義に従い変換すると次のようになる.

$$NOTFLY(x) = ANIMAL(x) - exc\text{-}NOTFLY(x) \quad (1)$$
$$ANIMAL(x) = BIRD(x) \quad (2)$$
$$exc\text{-}NOTFLY(x) = BIRD(x) - exc^2\text{-}NOTFLY(x) \quad (3)$$
$$BIRD(x) = PENGUIN(x) \quad (4)$$
$$exc^2\text{-}NOTFLY(x) = PENGUIN(x) - exc^3\text{-}NOTFLY(x) \quad (5)$$

ここで,$ANIMAL(x)$と$BIRD(x)$についてはそれらをヘッドとする完全ルールが上記以外にも存在するものとし,それらから得られる集合をそれぞれR(動物の集合),S(鳥の集合)とする.

このとき,$NOTFLY(x)$を求めるには,一番次数の高い式(5)のルールから評価する.いま,$exc^3\text{-}NOTFLY(x)$をヘッドとするルールがないので,このルールは完全ルールと同様に評価され,$exc^2\text{-}NOTFLY = \{Tweety\}$という集合を得る.次に,式(3)のルールを評価する.ここで鳥の集合は式(4)のルールと集合Sにより$BIRD = S \cup \{Tweety\}$となる.よって$exc\text{-}NOTFLY = S$という集合を得る.最後に式(1)のルールを評価する.動物の集合は鳥の場合と同様にして$ANIMAL = R \cup S \cup \{Tweety\}$のようになる.よって,$NOTFLY = R \cup \{Tweety\}$が得られる.

5 おわりに

不完全な知識を伴う論理型データベースの関係代数による問合せ手続きを提案した.今回の提案では,再帰を伴わない規則について効率的に妥当な結果が得られることを実験的に確認した.今後は,さらに再帰を伴う式にも適用範囲を広げ,また不完全ルールによる多重拡張にも適切に対処できるよう検討していく必要がある.

尚,本研究の一部は文部省科学研究費(重点領域研究(知識科学)No.04229211)の補助による.

参考文献

[1] J.Minker: "Foundations of Deductive Database and Logic Programming", Morgan Kaufmann(1988).

[2] 馬場口,大川:"完全/不完全知識を扱う高次論理型データベースにおける知識獲得",文部省科学研究費重点領域(知識科学)成果報告論文集(1994.3).

[3] 井戸,馬場口:"完全/不完全なルールに対する推論手続き",情報処理学会研究報告,93-AI-86-3(1993).

[4] R.C.Moore: " Semantical Considerations on Non-Monotonic Logic",*Artif.Intell.*,25,1,pp75-94(1985).