DE86-22

手続知識を管理する演繹データベースへの代数的アプローチ

An Algebraic Approach to Deductive Database for managing procedural knowledge

大森 匡　　田中 英彦

Tadashi OHMORI　　Hidehiko TANAKA

東京大学工学部

The University of Tokyo

1. 研究の目的

CAD ・マルチメディア等の分野では，従来の宣言的な知識に加え，『手続知識』をも管理するデータベースシステムが必要とされている。そのために，論理型言語による演繹データベースシステム（以下DDB ），オブジェクト指向に基づくオブジェクト指向データベースシステム（以下ODMS）が提唱されてきた。これら従来の研究は，『手続』に対応するデータ（i.e. DDB におけるルール節，ODMSにおけるメソッド）を主記憶にのみあると仮定して実装方式を検討している。

しかし，より実用的なシステムでは『手続』自体も大規模化し，二次記憶に置く必要があると我々は考える。我々は，DDB を用いてこのデータベース管理システム（以下DBMS）を開発する。

本論文では，DDBにおいて大量のルール節が二次記憶にある環境下のデータベース管理システム（以下ルールDBMS）を提案する。 そのため，大量にあるルール節（以下ルール）を『メタ関係』と呼ぶ集合データ構造で管理する。 ルールDBMSへの質問は『単一化を伴う関係代数（Relational Algebra extended with Unification；RAU ）』と呼ぶ集合演算による質問木形式で記述，実行する。

単一化を関係代数に組み込む概念は，ICOT-VLKBMによってRBU 演算として初めて提案された〔Mori86〕。RBU は関係データベースマシン上でProlog処理系を記述しているが，我々は大規模ルールDBMSの効率良い管理と高速質問処理を目的とする。 以下，ルールDBMSに必要な諸機能を述べ，我々のRAU によるDBMSがこの機能を満たすことを述べる。

2. 大規模ルールDBMS

2.1 ルールDBMSの定義

本論文ではファンクタを許したDDB を扱う。これは，〔Lloyd84〕に言うgeneral clauseの集合である。general clauseはルール節（以下ルール）とファクト節（以下ファクト）の集合である。さらに，次の制約を加える。

① ルールは関係データベース（以下RDB ）のビュー定義。これによって定義された関係を導出関係と呼ぶ。

② ファンクタは許す。再帰述語は許さない。

故にルールのBody部は全て関係代数プログラム（以下RAP ）にコンパイル可能である。ファンクタを許しても適当な演算子を加えればやはりRAP にコンパイルできる。

③ ファクトは変数を含まない。故にその述語に対応する関係のタプルとして表現できる。ファンクタは許す。この関係を基底関係と呼ぶ。

本論文では，この仮定の下で次のような環境を想定する。

・ルールが大規模化して二次記憶にある。

かつ，

・ファクトも二次記憶にあり，RDB として管理されている。

2.2 ルールデータベースと質問例

ルールDBMSの管理対象となるルールデータベース（以下ルールDB）の例を図1に示す。

今，各ルール節のHead部の述語をそのルール節の『種類』と呼ぶ。図1では二種類のルールec,tg が管理されている。ルールがディスクに置く程大規模になる原因は

① 一種類のルールが大量 （一種類による大規模）

② ルールの種類が大量 （多種類による大規模）

の二つである。

①の例

図1の種類”ec”のルールは，ネットワークと端末との接続条件を記述している。ネットワークのどの地域（AREA）にどの型（TERMTYPE）の端末をつなぐかに応じて，Body部で参照される関係表やそれらへの操作の仕方が全く異なるとすると，AREAが100 個，TERMTYPEが100 個なら，種類”ec”に属するルール節は10000 個となり大規模化する。

②の例

知識ベースkb1 が10000 種類のルールを知っている場合。

今，このルールDBへの質問Q1,Q2 を次のように与える。

〔管理されているルール節〕

・ec (TERMTYPE, GID, AREA, UID) :— COND.
地域 (AREA) のネットワークにグループID (GID), ユーザID (UID), 端末型 (TERMTYPE) の端末が接続する条件 (COND)。

・tg (TERMTYPE, GID, AREA) :— COND.
地域 (AREA), グループID (GID), 端末型 (TERMTYPE) の端末がグラフィック能力を持つ条件 (COND)。

〔操作用のメタ述語〕

・ruledb (KB, HEAD, BODY).
知識ベース (KB) がルール節 (HEAD) :— (BODY). を知っている。

・demo (KB, GOAL).
理論 (KB) において, (GOAL) が定理。 特に, (KB) =rdb はファクト集合を管理している関係データベース。

図1. ルールデータベースの例

Q1「グループID"kelly"の端末がネットワークに接続しており, かつグラフィック能力をもっているための条件を求めよ。但し, 接続条件は"kb1"が, グラフィック能力は"kb2"が知っているルールを使うこと。」

Q2「Q1の条件を満たす端末のユーザIDを求めよ。」

これを, 図1のメタ言語による操作述語ruledb,demo を用いて記述すると下の様に書ける。ルールDBMSはこうした質問を高速に処理しなければならない。

Q1

```
q1 (TERMTYPE, AREA, UID, COND) :—
  ruledb (kb1,ec (TERMTYPE, kelly, AREA, UID) ,COND1) ,
  ruledb (kb2,tg (TERMTYPE, kelly, AREA) ,COND2) ,
  COND= (COND1,COND2) .
```

Q2

```
q2 (UID) :— q1 (TERMTYPE, AREA, UID, COND) ,
            demo (rdb , COND) .
```

2.3 ルールDBMSの必要機能

2.2節の質問Q1, Q2が表すように, 大規模ルールDBMSは, ルールが二次記憶にあるとき, 次の機能が必要である。

〔1〕ルールの高速検索 — Q1

〔2〕検索したルールを実行してファクトを高速に検索する — Q2

〔2〕はさらに次の四つの必要機能に分類される。

① ルールの実行は別のルールを呼び出し, 二次記憶上のルールにランダムアクセスを起こす。これを回避する機能。

② 評価戦略の変更

例. 今, A, B :条件A, Bを表すルールの集合

I〔ルール集合〕:ルールを満たすファクト集合

として, 条件A&Bを満たすファクトI (A&B) は, ルールをどこで実行するかで三通りの評価戦略が可能。

i) AかつBのルールを求めて実行しファクトを求める。

ii) ファクトIAを求めてから, これに対応するBのルール ((IA) &B) を出して実行する。

iii) ファクトIA, IBを求めてその共通部分を求める。

状況に応じてこれら評価戦略を変更できることが必要。

③ ルールを実行すると, ファクトデータベースへのアクセスが発生する。これをファクトDBアクセスプラン (以下プラン) と呼ぶ。この最適化。

例. p (X,Y) :— r1 (X,Y), r2 (Y). に対しゴール節 { :— p (1,Y). } が与えられたとする。r1,r2はファクト節集合でRDB 上にある。このとき, r1はX=1 でselection を始めからすべきで { (X,Y) | p (X,Y) is true } を求めてからX=1でselection とすべきでない。

④ ルール実行時の共通項共有

例. ルール p (X) :— RAP1,RAP0.
p (X) :— RAP2,RAP0. があって, ゴール節 { :— p (X) } が与えられると, RAP0を共有すれば一回の実行ですむ。

2.4 従来のDDB 実装法のルールDBMSとしての欠点

従来のDDB 実装法としてTopdown 法とBottom up 法とを取る。Topdown 法は, 質問をProlog処理系でRAP に対応するゴール節に変換後, 関係DBで実行する。 Bottom up 法はルール節集合を予め関係代数のネットワークにコンパイルして全てRAP で実行する。

2.3 節の必要機能の内, Bottom up 法は

〔2〕—① Body部をRAP ネットワークに予めコンパイルすることで満たしている。

〔2〕—④ コンパイル時に予め大きな単位で共通項を共有しており満たす。

〔2〕—③ 共有の単位が大きすぎ, RAP ネットワークの枝刈り等の最適化は殆ど不可能である (図2)。そのため, 各ルールから参照されるファクト関係へのアクセスが充分に最適化できない。結果として, ファクトのデータベースへの大量のディスクアクセスが発生する。故に, 満たさない。

〔1〕, 〔2〕—② ルールの検索はできず, 固定評価戦略のため満たさない。

対してTopdown 法は

〔1〕 ルールの検索は可能だが一回にルール一個を処理するので2.2 節の質問Q1ではO (n^2) (nはec, tg各種類のルール節の数) となり満たさない。

〔2〕—③ 実行すべき必要最小限のプランのみを生成し実行するので満たす。

〔2〕—④ 個々のプランを独立して実行するので満たさない。

〔2〕—①, ②は共に満たさない。

i）ゴール節 {:- p(1,Y).}, {:- p(2,Y).} がルール p(X,Y) :- r1(X,Y). に与えられたとする。(r1は関係代数プログラム(RAP))。 このとき, $\sigma_{X=1}(r1)$ と $\sigma_{X=2}(r1)$ をすべきである。このルールに対応するRAP (p(X,Y)にあたる) を共有すると, {(X,Y) | r1(X,Y) is true.} を求めてからX=1 or2 でselection をするので大量のディスクアクセスが発生。

ii）ルール節

p(g(3,X)) :- RAP1. q(g(X,Y)) :- RAP3.
p(g(X,5)) :- RAP2.

にゴール節 {:- p(X), q(X).} を与える。このとき, RAP1 $\bowtie \sigma_{X=3}$(RAP3) と RAP2 $\bowtie \sigma_{Y=5}$(RAP3) を実行する方が, {X | q(X) is true.} を共有してRAP3を直接実行するよりディスクアクセスが少ない。

図2．共通項共有によるファクトDBアクセスへの悪影響

3．代数的アプローチ

3.1 基本的な概念

以下，ルールのBody部は予めRAP にコンパイルされているとする。こうするとルールDBへのランダムアクセスは回避でき，上の必要機能〔2〕—①が満たされる。このとき2.2 節に示したルールDBの大規模化の要因の内，多種類のルールによる大規模性が一種類のルールによる大規模性に変換される。

例 p :- q,r. で，q,r が共にn個のRAP にコンパイルされると，p はn^2個のRAP にコンパイルされる。

故に，一種類の条件を表すルール節が大量に二次記憶にある環境化で高速な質問処理を実現しなければならない。2.4 節に示した従来の方式は一度に一個のルール節を処理しているので不適当である。我々は，集合と集合演算によるルールDBMSを提案する。基本的なアプローチは以下のように極めて単純である。

「今，一種類の条件A,B を表すルールの集合を各々A，Bとし，次の代数演算を用意する。

A uJOIN B： 条件Aかつ Bを表すルール集合
σ_F A ： Aの内，制約Fを満たすルール集合
π_C A ： 属性Cに関するAのルール集合
IA ： ルールAを満たすファクト集合

これら集合データ構造と集合演算を用いて，ルールDBMSへの質問をRDB 同様，質問木（以下，RAU 質問木）で記述する。この質問木を，その中間ノードである各演算子間の可換則に基づいて最適化した後，高速集合演算アルゴリズムで実行する。」

例えば，質問Q3 「条件Aにさらに制約F1を加えたルールを満たし，かつ，条件Bに制約F2を加えたルールを満たすファクトを求めよ。」は，上の集合演算によって次のように記述できる。

Q3 = π I〔(σ_{F1} A) uJOIN (σ_{F2} B)〕

この一種類のルール節からなる集合をメタ関係，各集合演算を 「単一化を伴う関係代数 (Relational Algebra extended with Unification, RAU)」と呼ぶ。

3.2 RAUのルールDBMSとしての利点

RAUによるルールDBMSは2.3 節の各必要機能を満たす。

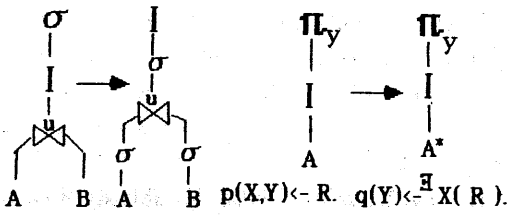
〔1〕RAU の各集合演算アルゴリズムを，集合演算アルゴリズム，専用ハードウェアによって高速に実行すれば可能。

〔2〕—① 各ルールのBody部を予めRAP コンパイルしたのでBottom up 法同様満たす。

〔2〕—② I演算子を含む一つの質問木は一つの評価戦略を示している。RAU には演算子間の可換則 I・I = I, I (A uJOIN B) = (IA) uJOIN (IB) = I (IA uJOIN B) が成立し，質問木を変換できる。故に，オペラントであるメタ関係間の統計情報に基づいて評価戦略を決定できる。

〔2〕—③ Topdown 法同様，実行すべき必要最小限のルール集合を求めることが可能（A uJOIN B）なので，各ファクトDBアクセスプランは最適化される。さらにRAU 質問木は σ, πを葉ノードに落とすことで，実行するルールに制約を加えることが可能。

例.



〔2〕—④ ファクトDBアクセスプランは集合単位で実行される（ルール集合Aに対してIAを実行する場合）ので，その中で共通項を検出し共有することは可能。

4．メタ関係と単一化を伴う関係代数（RAU）

4.1 メタ関係の定義

オブジェクト理論のルール節はメタ理論の複合項である。メタ理論の述語を「一種類のルール」に応じて与えれば，述語が一定なファクト節の集合は関係表で表現できる。

定義1．〔メタ関係〕

メタ理論Tm の述語 p(arg1,...,argN) に応じてメタ関係 rel(p) が次のように定義される。

・スキーマ＝〔arg1,...,argN〕 (argiは属性)
各属性の定義域はTm の複合項
・スキーマを満たすタプル＝そのスキーマの各属性から定義域への関数
・メタ関係＝そのスキーマを満たすタプルの集合
(但し，変数の有効範囲は，その変数が現れる各タプル内)

□

各メタ関係R のタプルはメタ理論の整式に対応する。この写像Lr：タプル → 整式 は各スキーマによって決定する。

R[A, B, Cond1] : a set of rule r(A,B) :- Cond1.
T[A, Cond2] : a set of rule t(A) :- Cond2.

R

A	B	Cond1
f(X)	a	rel1(X)

this tuple means r(f(X),a) :- rel1(X).".

T

A	Cond2
X	rel2(X)

this means "t(X):- rel2(X)."

図3．メタ関係の例

4.2 RAU演算子の定義

以下では，

M, N, R － メタ関係．

θ--- substitution とする。

定義2．〔$\in$, $\in_w$〕

タプルt,メタ関係Rに対し，

・『$t \in R$』$= \exists t1, \exists\theta$: renaming substitution, t1がRに属し，かつ， $t = t1\theta$．

・『$t \in_w R$』$= \exists t1, \exists\theta$: $t1 \in R$ かつ $t = t1\theta$

・タプルtに対し，t〔i〕はtのi番目の要素。

□

例 図3で

t1=〔f(Y),a, rel1(Y)〕 $\in$ R.

t2=〔f(1),a, rel1(1)〕 $\in_w$ R. not $\in$ R.

t1〔3〕=rel1(Y)．

定義3．〔$\stackrel{u}{=}$,=〕

・A $\stackrel{u}{=}$ B ：AとBが単一化可能。単一化は実行する。

・A＝B ：等式論理における等号。

□

定義4．〔単一化を伴う関係代数演算〕

次の演算子を総称して『単一化を伴う関係代数（Relational Algebra extended with Unification, RAU』）と呼ぶ。

〔和〕A ∪ B —— 関係代数と同じ。

〔直積〕A × B —— 関係代数と同じ。但し，メタ関係の定義により，A,B 間に共有変数はない。

〔選択〕$\sigma_{F(t)}A = \{t \mid t \in_w A$ かつ F(t)$\}$

但し，F(t)は $\bigwedge_{i=j}(t〔i〕=Ci)$ の形。Ciは t〔j〕を変数として許した複合項。Ciの t〔j〕形以外の変数の有効範囲は F(t)。 e.g. t〔1〕= f(t〔2〕,X)．

例. 図3で『属性Aが f(X) 形になっているときの条件tを表すルール集合』=

$\sigma_{1 \stackrel{u}{=} f(X)} T[1,2] =$

A	Cond3
f(X)	rel2(f(X))

This is a set of rule q(A) :- Cond3.
q(A) is defined as q(A) :- t(A), A=f(X).

〔合成〕R〔B1,...,Bm〕$= S_{scheme}$ (M〔A1,...,An〕) $=$
$\{t \mid \exists t1,\ t1 \in M$ かつ $t = scheme(t1)\ .\}$.

但し，scheme＝〔B1,...,Bn〕。各BiはA1,...,Anのみを変数とする複合項であり，scheme（t）は各Aiに t〔i〕を代入して得られる複合項。

例. scheme＝〔f(A1,A2),A3〕，

〔展開〕M〔A1,...,An〕$= P_{scheme}$ (R〔B1,...,Bn〕) $=$
$\{t \mid \exists t1,\ t1 \in R$ かつ $t1 =$scheme$(t)\}$.
但し，schemeの定義は上と同じ。S の逆演算子。

例.

$S_{[f(A_1, A_2), A_3]}$

A1	A2	A3
g(X)	a	X

=

B1	B2
f(g(X),a)	X

$P_{[f(A_1, A_2), A_3]}$

B1	B2
f(g(X),a)	X

=

A1	A2	A3
g(X)	a	X

〔代入〕R〔A1,...,An〕= I cond (M〔A1,...,An〕) $=$
$\{t \mid \exists t1, \exists\theta$; $t1 \in M$ かつ (cond(t1))θ が定理.
かつ $t = t1\theta$.$\}$.
但し，condはA1,...,Anのみを命題変数とした命題論理式。
cond(t1) は，各Aiにt1〔i〕を代入した論理式。

例. 図3で『条件tを満たすファクト集合』

$= I_2 T[1,2] =$

1	2
a	rel2(a)

where rel2

1
a

This is a set of fact satisfying rule t.

〔条件付差〕

R〔A1,...,An〕=M〔A1,...,An〕 $\overline{}_{Alist}$ N〔A1,...,Am〕 $=$
$\{t \mid t \in M$ かつ
$\forall s\ (s \in N \rightarrow$ not $(t〔Alist〕 \stackrel{u}{=} s〔Alist〕))\}$
但し，Alist は {A1,...,An} の部分集合。

例.

A1	A2
X	f(X)
h(X)	a

$\overline{}_{A1}$

A1
g(X)

=

A1	A2
h(X)	a

unifiable

さらに，略記号を定義する。

〔射影〕π_{Alist}M〔A1,...,An〕$= S_{Alist}$ M.
但し，Alist は {A1,...,An} の部分集合。

例. 図3で『属性Aに関する条件rの集合』=

$\Pi_{[1,3]}$ R[1,2,3] $= S_{[1,3]}$ R[1,2,3]

=

A	Cond4
f(X)	rel1(X)

This is a set of rule q(A) :- Cond4. q(A) is defined as q(A) :- r(A,B).

〔単一化を伴う自然結合 --- uJOIN or $\stackrel{u}{\bowtie}$〕

R〔A,B,C〕=M〔A,C〕uJOIN N〔B,C〕 $=$
$\pi_{〔A,B,C〕}\ \sigma_{M.C \stackrel{u}{=} N.C}$ (M×N).
但し，C はMとNの共通属性のリスト。

例．図３で『条件ｒかつｔを表すルールの集合』＝

R

A	B	Cond1
f(X)	a	rel1(X)

$\overset{u}{\bowtie}$ T

A	Cond2
X	rel2(X)

＝

A	B	Cond1	Cond2
f(X)	a	rel1(X)	rel2(f(X))

this set is q(A,B) :- Cond1,Cond2.
q is defined as q(A,B) :- r(A,B),t(A).

〔ρ演算子〕
$\rho_{TERM}\ M\ (A1,\ldots,An) =$
$\pi_{Vlist}\ P_{TERM}\ \sigma_{(A1,..,An)\ \overset{u}{=}TERM}\ M\ (A1,\ldots,An)$
但しTERMは複合項のリスト。Vlist はTERM中の変数リスト。

$\rho_{[f(X),\ h(X)]}$ [| A1 | A2 |
|---|---|
| f(X) | h(a) |

] ＝

X
a

$= \pi_X P_{[f(X),\ h(X)]}\ \sigma_{[A_1,\ A2]\ \overset{u}{=}\ [f(X),\ h(X)]}$ [

A1	A2
f(X)	h(a)

]

□

上で定義された演算子σ，π，uJOINは，ホーン節に限れば3.1 節の演算子要求をみたしている。例えば，2.2節の質問Q1,Q2 はRAU 演算子による質問木として，図４のように記述できる。

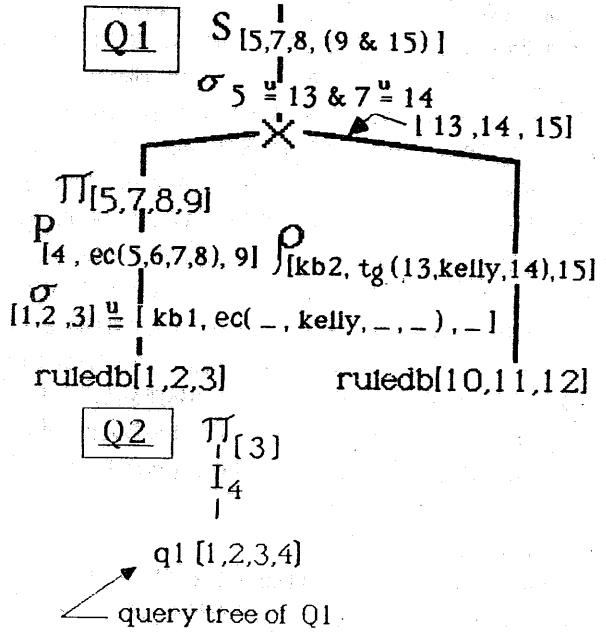


図４．RAU 質問木の例。

5．RAU の記述能力

ＲＡＵが理論Ｔにおける定理証明システムであるためには，『Ｔの任意の定理はRAU で計算可能であり，かつ，RAU で計算可能なＴの整式は定理である。』ことが必要である。これを代数的等価性と呼ぶ。　本節では，一階述語論理の部分クラスとして『単一化を伴う関係論理（Relational Calculus extended with Unification,RCU ）』を定義し，その中で代数的等価性を持つ質問クラスを与える。

定義５．〔単一化を伴うドメイン関係論理RCU 〕
式｛(d1,...,dn) | W (d1,...,dn)｝（Wは整式）は，メタ理論Tm に関して次の条件を満たすときRCU 式である。
・アトム＝Ri (TERMi) or demo (KBi,ri (TERMi))　or
”＜variable＞＝Tm の複合項”　として，
Wがアトムの {AND,NOT,∃} から成る整式。
・d1,...,dnのみがWの自由変数（ドメイン変数）。
但し，
・Riは任意のメタ関係名であり，Tm の述語。
・riは変数を含まないファクト節集合を表す関係名。即ち，演繹データベースの基底関係か，または導出関係名。Tm のファンクタ。
・TERMi は変数を許したTm の複合項のリスト。
e.g. TERM1 ＝〔f (X) ,a,X〕のとき，r1 (TERM1) は r1 (f (X) ,a,X) を意味する。

□

RCU に対して”safety”を定義する。
定義６．〔ドメインＤに関するRCU 式のsafety〕
RCU 式｛(d1,..,dn) | W｝は，Wの関数Ｄ＝ dom (W) に対して以下の条件を満たすとき”safe”である。
・Ｄは複合項の集合。
・W中の全てのメタ関係と，（ファクト節集合である）基底関係とに任意の値を代入したとき，次の条件を満たす。
i) ∀(d1,..,dn) 〔W (d1,..,dn) → each di $\in_W$ D) 〕
ii) ∃u (W1 (u)) をWの任意の部分式として，
For all variable 〔W1 (u) → u $\in_W$ dom (W1) 〕.

□

safeなRCU の部分クラスは代数的等価性を持つ。
定義７．〔質問クラスRCU-C1〕
質問Ｑ｛(d1,...,dn) | W (d1,...,dn)｝は，次の条件を満たすときRCU-C1式である。
・W (d1,...,dn) がdemo (KBi , ri (TERMi)) の
の {AND,NOT,∃} から成る整式。
・Ｄ＝dom-C1 (W) ＝$\bigcup_{Vi}\pi_{各属性}\ \rho_{TERM_i}$ (Ri)
に対してＱがsafe．但し，Riは
｛(x1,...,xj) | demo (KBi,ri (x1,...,xj))｝を表すメタ関係。riはｊ引数の基底関係か導出関係なので，Ri，Ｄは変数を含まない。明らかに，Ri ＝π I 〔riを定義するルール節又はファクト節集合たるメタ関係〕の形で記述される。□

定義８．〔質問クラスRCU-C2〕
質問Ｑ｛(d1,...,dn) | W (d1,...,dn)｝は，次の条件を満たすときRCU-C2式である。
・Wは∃(x1,...,xj) 〔(d1,...,dn) ＝scheme (x1,...,xj) &
V (x1,...,xj) 〕の形。

・Vは Ri (TERMi) の {AND,OR,∃} よりなる整式。但し，Riは 一般のメタ関係。Vの自由変数はx1,...,xjのみ。

・$D = \mathrm{dom\text{-}C2}(V) = \bigcup_{V_i} \pi_{\text{各属性}} \cdot \rho_{TERM_i}(Ri)$
に対してVがsafe.

□

クラスC1,C2 はそれぞれ『ルールを検索して実行し，ファクトを求める。』，『ルールを検索する。』ための質問である。C1,C2 に対応するRAU の部分クラスを定義する。

定義９．〔RAU-C1,RAU-C2〕

・〔RAU-C1式〕

RAU 演算子の集合 {Pσ (Ri)，×，σ，U，－，π} (但し,Ri は変数を含まないメタ関係) から有限回で生成可能な演算子列をRAU-C1式と呼ぶ。

・〔RAU-C2式〕

RAU 演算子の集合 {Pσ (Ri)，×，σ，U} (ただし，Riは任意のメタ関係) から有限回で生成可能な演算子列Eに対して，Sscheme (E) をRAU-C2式と呼ぶ。

□

RAU-C1のメタ関係Riは，変数を含まない式ＩＲを含んでいる。明らかに，次の定理が証明される〔大森87〕。

定理〔クラスC1,C2 の代数的等価性〕

- クラスRCU-C1,RCU-C2 はそれぞれRAU-C1，RAU-C2と代数的に等価
- RCU-C1,-C2 はそれぞれ，〔Lloyd84〕にいう general program GP，program Pに変換できる。
- RCU-C1はcomp (GP) 上のSLDNF 導出において健全かつ完全。RCU-C2はP上のSLD 導出において健全かつ完全
- クラスRAU-C1,-C2は理論RCU-C1,-C2においてそれぞれ健全かつ完全。

〔証明〕クラスC1は，本質的に〔Ullman82,Lloyd86〕と同じ。C2は，safetyによって q (X,Y) :- r1 (X) OR r2 (Y). を排除している。詳細は〔大森87〕参照。 □

6．RAU 質問木の最適化戦略

RAU 演算子には以下の可換則が成立する。

① $\sigma_{1\&2}(A \times B) = \sigma_{1\&2}(\sigma_1 A \times \sigma_2 B)$.
(1,2 は各々A,B の属性のみへの選択述語)

② $S_{\{p(1),p(2)\}}(A \times B) = S_{p(1)}A \times S_{p(2)}B$.
(p (1) ,p (2) は各々A,B 各属性のみから成るscheme)

③ (A uJOIN B) uJOIN C = A uJOIN (B uJOIN C).

④ $\sigma I A = I \sigma A$.

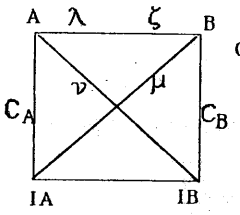
⑤ $\pi_1 I_3 A\ (1,2,3) = \pi_1 I_3 \pi_{1,3} A\ (1,2,3)$.

⑥ $I_{A\&B}(A \text{ uJOIN } B) = I_B((I_A A) \text{ uJOIN } B) = (I_A A) \text{ uJOIN } (I_B B)$.

⑦ $IR \overline{\cap}_A IT = IR \overline{\cap}_A I\ ((\pi_A IR) \text{ uJOIN } T)$.
(但し，ＩＲ，ＩＴ共に変数を含まない。)

①～③によって質問クラスＣ２は関係データベースのＰＳＪ質問クラスと同じ最適化技法が適用可能である。④，⑤はRAU 質問木の実行途中にユーザから制約条件を与えられて探索空間を削減する際に用いられる。

質問クラスＣ１は⑥によって評価戦略変更を行う。ここでは，質問木中のuJOIN 順とＩ演算子の位置について，図５のような「質問グラフ」を生成し，最適化を行なっている。図５は質問Ｉ（A uJOIN B）を実行する際の質問グラフである。(Ri,IRi) 形以外の辺 (n1,n2) のパラメタは，n1 uJOIN n2 における単一化成功率を示す統計情報である。(Ri,IRi) のパラメタはRiの実行コスト。一つの辺に印を付けると，その辺に対応するRAU 演算子が実行される。



C_A C_B : execution-cost for A, B.

λ = (A uJOIN B中のタプルを生成するAのタプル数) ／|A|

ζ, ν, μも同様。

図５．質問グラフ Ｉ（A uJOIN B）．

C1の最適化は，この質問グラフ上でコスト和最小の印付け系列を求める問題である。これはＮＰ完全であり，Ｐ≠ＮＰならε近似算法も存在しないことが証明される。我々は，最も単純な最近隣法を修正して使用し，良好な結果を得た。

7．考察

RAU 演算子の内，σ，πはＯ（ｎ）のフィルタプロセッサで処理可能である。最も負荷の重い uJOINの実行アルゴリズムは既に発表している〔大森86，87〕。RAU-DBMSは，２章に述べた一種類のルールによる大規模性を本質的に解決している。多種類のルールによる大規模性は，RAP コンパイルによって回避しており，根本的にはページングアルゴリズムが必要である。また，共通項の共有は今回は行っておらず，RAU への実装は今後の課題である。再帰述語に関しては，現在の研究は文法上の制約をつけてRAP にコンパイルする方式が中心であり，本論文の方式と矛盾しない。ICOTのVLKBM との違いはその計算モデルにある。即ち，本論文のRAU-DBMSは大規模なルールの処理と大規模なファクトの処理を共に，『集合と集合演算による質問木表現，この質問木の最適化，集合演算の高速実行』という統一された計算モデルによって行う。

8．結論

本論文では，大規模ルールDBMSとして『集合と集合演算』による代数的方式を提案し，その有効性を示した。

〔参考文献〕

〔Lloyd84〕J.W.Lloyd, "Foundation of Logic Programming", 1984,Springer Verlag.
〔Lloyd86〕J.W.Lloyd,R.Topor, "A BASIS FOR DEDUCTIVE DATA-BASE SYSTEM II",Journal of Logic Programming,vol.3,No.1, 1986,pp.55-67.
〔Mori86〕Y.Morita, et.al, Proc. of VLDB86,1986,pp.52-59.
〔Ullman82〕J.D.Ullman, "Principles of Data Base Systems", 1982,Computer Science Press.
〔大森86〕大森，他．情報処理学会全国大会第33回5L-6．1986年
〔大森87〕大森．東京大学情報工学専門課程修士論文1987年２月