

関係データベースにおける複数質問処理の効率化について

京大 工学部 吉川正俊 (Masatoshi Yoshikawa)

上林弥彦 (Yahiko Kambayashi)

矢島脩三 (Shuzo Yajima)

1. まえがき

関係データベースシステムでは、高水準の質問言語を支援しているため、他のモデルに基づくデータベースに比べて、より使い易い利用者インタフェースを実現可能である。しかし、このことは同時にシステムによる質問最適化が全体の効率向上に大きな影響を与えることを意味する。データベースは本来、複数の利用者によって共有されるものであるにもかかわらず、従来の質問処理アルゴリズムはほとんどすべてが単一の質問のみを対象としてきた。本稿では複数の質問の処理について考察する。ここで言う“複数質問”とは、複数の利用者の質問のみならず、単一の利用者の質問を複数個に分解したもの（たとえば、論理和記号によって結ばれている複数の自然結合質問）をも含む。本稿では、環境として、分散型データベースを考え、処理が容易な複数質問の性質付けに

関する基礎的な考察を行なう。

2. 基本的事項

関係とは組の有限集合である。属性集合 $\{A_1, A_2, \dots, A_n\}$ から成る関係スキーマ R を $R(A_1, A_2, \dots, A_n)$ で表わす。関係の集まりをデータベースと言ひ、それを D で表わす。また、対応する関係スキーマの集まりをデータベーススキーマと言ふ。データベーススキーマ D' 内の関係スキーマがすべてデータベーススキーマ D にも存在するとき、 D' は D の部分スキーマであると言ふ。任意のデータベーススキーマは以下の判定法によって巡回型スキーマと α -非巡回型スキーマ^[1] に分類される。

[巡回型 / α -非巡回型スキーマの判定法]^{[2][3][4]}

- 1° 一つの関係スキーマのみに含まれる属性を除去する。
- 2° 他の関係スキーマを構成する属性集合の部分集合から成る関係スキーマを除去する。
- 3° 上記1°及び2°を適用不可になるまで繰り返した結果が空集合のとき、もとのデータベーススキーマは α -非巡回型スキーマであり、それ以外の場合は巡回型スキーマである。

また、任意の部分スキーマが α -非巡回型スキーマとなるようなデータベーススキーマを β -非巡回型スキーマ^[1] と呼

ぶ。

ある組 t のうち、属性集合 X に対応する値だけを取り出したものを、 t の (X 上への) 射影と言ひ、それを $t[X]$ で表わす。関係代数操作に関しては以下の表記法を用いる。

射影: $R[X] = \{t[X] \mid t \in R\}$

自然結合: $R_i \bowtie R_j = \{t \mid t \in R, t[R_i] \in R_i, t[R_j] \in R_j, \\ R = R_i \cup R_j\}$

あるデータベーススキーマ D の部分スキーマ $D'(R_1, R_2, \dots, R_m)$ 及び、ある関係スキーマ (目的関係と呼ぶ) $R_i (1 \leq i \leq m)$ に対し、

$$(R_1 \bowtie R_2 \bowtie \dots \bowtie R_m)[R_i]$$

を求める質問を自然結合質問と言ふ。また、同一の目的関係を有する複数の自然結合質問の共通集合、和集合、差集合 (或いはそれらの組合せ) を求める質問を自然結合集合質問と言ふ。データベース D に対する質問 Q の答を $Q(D)$ で表わす。任意の D に対し $Q_1(D) = Q_2(D)$ が成立するとき、2つの質問 Q_1 と Q_2 は等価であると言ふ。自然結合質問は各関係を節点とし、結合操作を施す関係に対応する節点どうしを無向枝で結んで得られる質問グラフによって表現できる。ただし、各枝には対応する結合属性をラベルとして付けるものとする。ある質問と等価な質問の質問グラフをその質問の被覆と言ひ、

被覆のうち木のものを被覆木と言う。自然結合質問のうち被覆木を持つものを木型質問, それ以外のものを巡回型質問と言う。木型質問は準結合操作を用いることにより, 効率的に質問処理を行なえることが知られている^[5]。

[例1] 5つの関係スキーマ $R_1(E, A)$, $R_2(A, B)$, $R_3(B, C)$, $R_4(C, D)$, $R_5(D, E)$ から成るデータベーススキーマ D の部分スキーマ $D'(R_1, R_2, R_3, R_4)$ 及び目的関係 R_1 から構成される自然結合質問

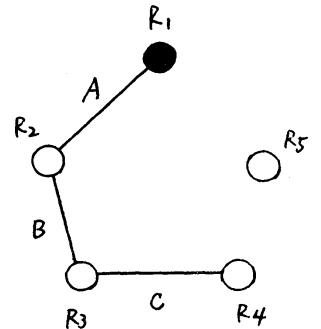


図1. 被覆木

(Q_1 とする) の質問グラフを図1に示す。図中, 黒丸の節点は目的関係を表わす。

本稿では, 各関係が相異なる別の地点に存在するような分散型データベースの環境のもとで質問処理を考える。

3. 複数質問の同時処理

一般に, 複数の利用者によって与えられた質問を処理する方法としては以下の2通りが考えられる。

- (1) 一つの質問を処理する際の間中間結果を蓄えておき, それを他の質問の処理時に利用する。
- (2) 複数の質問をまとめて同時に処理する。

上記(1)の方法では, どのようなデータと中間結果として保存しておくのが最も効率良いかを決定する必要がある。本稿

では、複数の利用者によって与えられた自然結合質問の同時処理について方法(2)の立場で考察するが、この立場に基づけば、単一の利用者によって与えられた自然結合集合質問に関しても、それを構成する各自然結合質問を別々の質問と考えることにより同様の議論を適用することができる。

分散型データベースにおける質問処理は、各地点における局所処理と地点間のデータ転送から成るが、一般には後者のコストの方が前者のそれよりもはるかに大きい。したがって複数の質問を同時に処理するためには、各質問を処理する際の転送データのうち起点と終点の地点が同一のものに関しては共通部分を同時に転送することにより、全体の質問処理コストを低減することが重要となる。

木型質問の処理は、被覆木において目的関係に対応する節点を根とし、葉から根に向かって順に準結合操作を実行することによって実現される^[5]。このような準結合操作列を表わす有向木を処理順序木と呼ぶことにすると、複数の木型質問の同時処理は、複数の処理順序木の重ね合わせの問題であると見なすことができる。

[例2] 例1の質問 Q_1 の処理順序木を図2(a)に示す。また、 D の部分スキーマ $D''(R_1, R_2, R_3, R_4)$ と目的関係 R_5 から成る自然結合質問(Q_2 とする)の処理順序木を図2(b)に

示す。 Q_1 と Q_2 はそれぞれ別の利用者によって与えられた質問と考えることもできるし、単一の利用者が与えた自然結合集

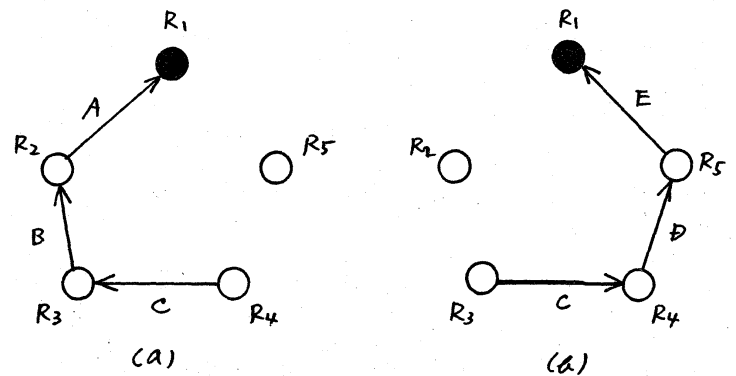


図 2. 処理順序木

合質問 ($Q_1 \cap Q_2$, $Q_1 \cup Q_2$, $Q_1 - Q_2$ など) の構成要素と見なすこともできる。

図 2 (a) の有向枝 $\langle R_4, R_3 \rangle$ と、同図 (a') の有向枝 $\langle R_3, R_4 \rangle$ は互いに向きが逆になっている。このように、異なる処理順序木の 2 つの枝で向きが逆のものは互いに逆枝であるという。2 つの質問をそれぞれの処理順序木に基づいて同時に実行する場合、逆枝に相当する準結合操作に対応するデータ転送は重ね合わせることが困難であるため、与えられた 2 つの質問に対し、互いに逆枝を持たないような処理順序木を選択することが重要となる。一般に 2 つの質問が与えられたとき、これらの処理順序木のいかなる組み合わせを考えたも互いに逆枝を持つとき、もとの 2 つの質問は逆枝を生じると言う。

[例 3] 図 2 (a) (図 2 (a')) の処理順序木は、質問 Q_1 (Q_2) の唯一の処理順序木である。したがって、 Q_1 と Q_2 は逆枝を生じる。

[例 4] 5つの関係スキーマ $R_6(A, D)$, $R_7(A, B, C)$, $R_8(A, B)$, $R_9(C, E)$, $R_{10}(B, F)$ から成るデータベーススキーマ上の2つの質問

$$Q_3 = (R_6 \bowtie R_7 \bowtie R_8 \bowtie R_9) [R_6] \quad \text{及び}$$

$$Q_4 = (R_6 \bowtie R_7 \bowtie R_8 \bowtie R_{10}) [R_6]$$

を考える。図3(a)の実線(破線)で示した枝は $Q_3(Q_4)$ のある処理順序木を示す。これら2つの質問は一見逆枝を生じるように見えるが、 Q_4 の処理順序木として図3(b)の点線の枝によって表わされるもの考えることにより、逆枝を解消することができる。

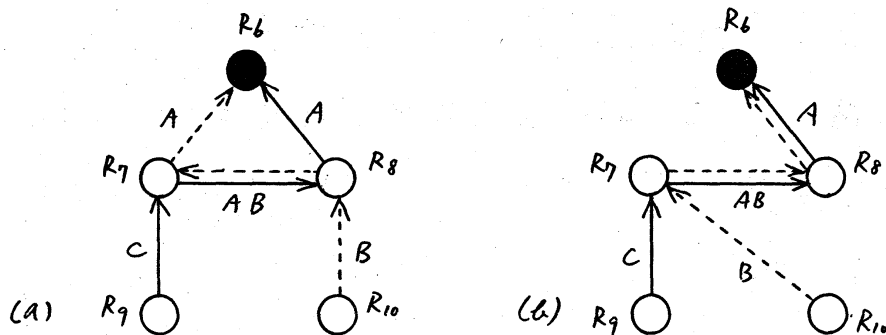


図3 処理順序木の選択による逆枝の解消

4. 処理が容易な複数質問

本節では、2つの質問が逆枝を生じないための一つの十分条件を与える。

[定理1] β -非巡回型スキーマ上で同一の目的関係を有する2つの自然結合質問は逆枝を生じない。

(証明) β -非巡回型スキーマ上の
 2つの自然結合質問を Q_1 と Q_2 とし、
 目的関係を R_T とする。 β -非巡回型
 スキーマの定義より Q_1 と Q_2 はいずれ
 も木型質問となる。 Q_1, Q_2 に対応す
 る処理順序木 T_1, T_2 が逆枝を持つと
 仮定する (図4参照)。図中、実線と
 点線はそれぞれ T_1, T_2 の一部を表わすものとする。 T_1 の有向
 枝 $\langle R_i, R_j \rangle$ と T_2 の有向枝 $\langle R_j, R_i \rangle$ が互いに逆枝となっている。
 T_1 における R_i から目的関係に至る有向経路と T_2 における R_i から
 目的関係に至る有向経路の最初の共通節点を R_c とし、 T_1 にお
 ける有向経路 $\langle R_j, R_c \rangle$ と T_2 における有向経路 $\langle R_i, R_c \rangle$ をそ
 れぞれ p_1, p_2 とする。質問グラフにおける枝 $\langle R_i, R_j \rangle$ のラベ
 ルを $X (= R_i \cap R_j)$ とする。

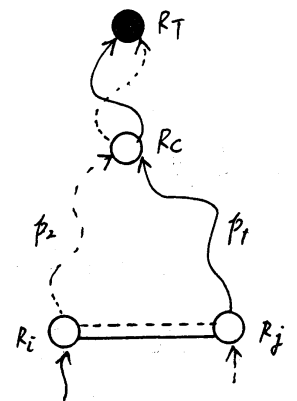


図4. 逆枝を持つ
 2つの処理順序木

ii) p_1 (または p_2) 上に $X' \subseteq X$ (または $X' \supseteq X$) なるラベル X' を持
 つ枝が存在する場合。

Q_1 (または Q_2) の被覆木として他のものを選択することによ
 り逆枝を除去することができる。例えば、 p_1 上に $X' \subseteq X$
 なるラベル X' を持つ枝 $\langle R_k, R_l \rangle$ が存在する場合には、経
 路 $\langle R_j, R_k \rangle$ 上の枝はすべてラベルとして X' を含む ($\because Q_1$ は
 木型質問) ため、その被覆として枝 $\langle R_k, R_l \rangle$ の替わりに

枝 $\langle R_i, R_c \rangle$ を選択することにより, 図 5 に示すように逆枝を除去できる。

(ii) (i) 以外の場合

P_1 上の節点の並びを $\langle R_j, R_1, \dots, R_{m-1}, R_c \rangle$,

P_2 上の節点の並びを $\langle R_i, R_{m+n}, \dots, R_{m+1}, R_c \rangle$

とし, 枝 $\langle R_i, R_{m+n} \rangle$, $\langle R_j, R_1 \rangle$ のラベル

をそれぞれ Y , Z とする。(ただし,

$R_m \equiv R_c$ とする。) (図 6 参照) ここでは

X, Y, Z が互いに素の場合だけを考える。

ただし他の場合も同様に証明できる。

以下の手続きを実行する。

0° $D \leftarrow \{R_i, R_j\}$

1° k を $R_i \cap R_k \neq \phi$ かつ $R_i \cap R_{k+1}$

$= \phi$ なる最小の整数とする。(ただし, $1 \leq k \leq m+n$,

$R_{m+n+1} \equiv \phi$)

2° $D \leftarrow D \cup \{R_k\}$

3° $R_c \leftarrow R_k$

4° h を $R_c \cap R_h \neq \phi$ かつ $R_c \cap R_{h+1} = \phi$ なる最小の整数とする。(ただし, $0 \leq h \leq c-1$, $R_0 \equiv R_j$)

5° $D \leftarrow D \cup \{R_h\}$

6° もし, $h \geq 2$ ならば $R_c \leftarrow R_h$ とし 4°へ

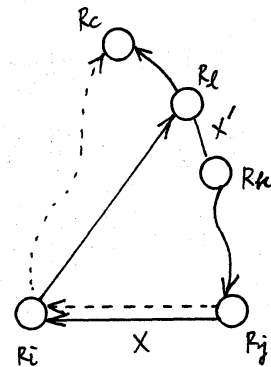


図5 枝の付け替え
による逆枝の消去

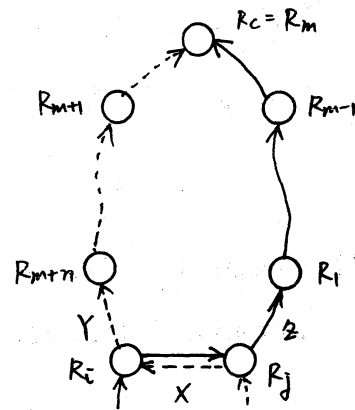


図6. 逆枝を含む閉路

以上の手続きの結果得られる関係スキーマの集合 $\mathcal{E}$ も質問と見なして質問グラフを構成すると単一の閉路ができ、すべての枝の属性ラベルは互いに素となる。このことは、もとのデータベーススキーマの部分スキーマに巡回型のものが存在することを意味し、初めの仮定に矛盾する。(証明終)

5. あとがき

異なる目的関係を有する複数の質問あるいは複数の目的関係を有する質問に対する考察が課題として残されている。

- [1] Fagin, R., "Types of Acyclicity for Hypergraphs and Relational Database Schemes", IBM Research Report, No. RJ3300, Nov. 1981.
- [2] Graham, M.H., "On the Universal Relation", Technical Report, Univ. of Toronto, Sept. 1979.
- [3] Yu, C.T. and Ozsoyoglu, M.Z., "An Algorithm for Tree-Query Membership of a Distributed Query", Proc. of IEEE COMPSAC, pp.306-312, Nov. 1979.
- [4] Goodman, N., Shmueli, O. and Tay, Y.C., "GYO Reductions, Canonical Connections, Tree and Cyclic Schemas and Tree Projections", Proc. of 2nd ACM SIGACT-SIGMOD Symposium on PODS, pp.267-278, Mar. 1983.
- [5] Bernstein, P.A. and Goodman, N., "The Theory of Semi-Joins", SIAM J. of Computing, Vol.10, No.4, pp.751-771, Nov. 1981.