数理解析研究所講究録
461巻 1982年 206-225



# 分散データベースの質問処理における データ転送量の削減について

京大 工学部 吉川 正俊
上林 弥彦
矢島 脩三

## 1. まえがき

近年の計算機ネットワーク技術の進歩に伴い，分散型データベースはその重要性を増しつつあり，同種及び異種モデルに基づく実験システムが構築されている[ROTHB8003][SMITB8105]. 分散型データベースにおける質問処理では各地点間のデータ転送に伴う通信コストを考慮に入れる必要があり，集中型の場合とは異なった処理アルゴリズムの開発が必要である。関係モデルに基づく同種型の分散型データベースシステムにおいて結合操作を効率的に実行するために準結合操作が提案されている[BERNC8101]. 準結合操作のみを用いて解くことができる質問は木型質問と呼ばれ，その処理の最適化に関する議論がいくつか行なわれている[CHIU8005][CHIU81]。ところが，木型質問以外の質問である巡回型質問に関しては，効率の良い処理方式はほとんど知られていない。我々は既に一般化準結

合操作を提案し，この操作を用いることにより任意の巡回型質問を解くことができることを示した[KAMBY8110]。本稿では巡回型質問の処理の効率化並びに仮性関数従属性（関係中に一時的に成立する従属性）を用いた処理の効率化について議論を行なう。

本稿の構成は以下のとおりである。第2章では本稿で用いる用語及び記号の定義を行なう。第3章においては，文献[KAMBY8110]で提案した巡回型質問の処理方法とデータ圧縮に関する基本的な説明を行ない処理の一例を示す。第4章においては，単純巡回型質問をもとにすべての関係を1つの地点に集めて処理を行なう方法と一般化準結合操作を用い，質問全域木に沿って順次関係を転送して行く方法の通信コストの比較を行ない，ほとんどの場合前者に比べ後者の方がコストが小さくなることを示す。第5章では，一般化準結合操作に基づいた処理の際に用いる質問グラフの全域木として，質問グラフの推移的閉包の極大全域木を考え，一般的にはこの全域木に沿った処理がコストの点で有利となることを示す。また第6章では目的属性を含む関係が複数個存在する場合に対する考察を行なう。第7章では第3章で述べたデータ圧縮による前処理の一般化を行ない，第8章では仮性関数従属性を用いた(i)巡回型質問の木型質問への変換，及び(ii)転送デー

タ量の圧縮について述べる。

## 2. 諸定義

属性$A_1, \ldots, A_n$からなる関係Rとは，各属性の定義域の直積の部分集合である。Rの要素を組と言う。関係Rの属性集合を関係スキーマと言い$R(A_1, \cdots, A_n)$あるいはRでそれを表わす。

$A_{ik}, A_{jk}$をそれぞれ$R_i$, $R_j$の属性とするとき$R_i.A_{ik} = R_j.A_{jk}$の形の条件節の論理積結合を等結合条件式と言う。本稿では以後，結合する属性が等しいとき，すなわち$A_{ik} = A_{jk}$の場合のみを考察の対象とする。データベースに対する質問は条件式（qualification）と目的属性（target attribute）から構成されるが，条件式が等結合条件式の場合はその質問を等結合質問と言う。以後，等結合質問のことを単に質問と言う。また，目的属性を得るために必要な関係を目的関係と言い，その集合を$R_T$で表わす。

等結合質問Qに対する質問グラフ$G_Q(V_Q, E_Q, L_Q)$はラベル付きの無向グラフである。ここで$V_Q$は節点集合であり，質問に現われる各関係に対して1つの節点が対応する。また質問Qの条件式中に条件節$R_i.A = R_j.A$が存在するときまたそのときに限り節点$R_i$と$R_j$の間を枝で結び，その枝にはこのよう

な属性Aのすべての和集合をラベルとして付ける。等価な質問に対応する質問グラフは互いに等価であると言う。$G_Q$と等価な質問グラフのうち枝数及びラベル数が最大のものを$G_Q$の推移的閉包と言い，それを$G_Q^+$で表わす。質問グラフが木で表わされる質問あるいはそれと等価な質問を木型質問と言い，木型質問以外の質問を巡回型質問と言う。また，単一の閉路のみから成る質問グラフで表わされる巡回型質問を単純巡回型質問と言う。

関係スキーマ$\underline{R}$に対し，$X$を$X \subseteq \underline{R}$なる属性集合とするとき，関係$R$の$X$上への射影を$R[X]$で表わす。2つの関係$R_1$, $R_2$に対し，次式で定義される属性集合$\underline{R_1} \cup \underline{R_2}$上の関係$R$を$R_1$と$R_2$の自然結合と言い，それを$R_1 \bowtie R_2$で表わす。

$$R = \{t \mid \exists t_1 \in R_1,\ \exists t_2 \in R_2 \ \text{s.t.}\ t[R_1] = t_1 \text{ かつ } t[R_2] = t_2\}$$

関係$R_2$による$R_1$の準結合は$R_1 \ltimes R_2$で表わされ，次式で定義される。

$$R_1 \ltimes R_2 \triangleq (R_1 \bowtie R_2)[\underline{R_1}]$$

$X$を$\underline{R_1} \cap \underline{R_2} \subseteq X \subseteq \underline{R_2}$を満足する属性集合とするとき，$X$における$R_2$による$R_1$の一般化準結合は$R_1 \overset{X}{\ltimes} R_2$で表わされ，次式で定義される。

$$R_1 \overset{X}{\ltimes} R_2 \triangleq R_1 \bowtie R_2[X]$$

関係表において一時的に成立する関数従属性を仮性関数従

属性（以後SFと）と呼ぶ。

本稿では，分散型データベースを考え，この分野の他の大部分の研究にならい各関係は互いに異なる一つの地点に存在するものと仮定する。関係Rの存在する地点をS(R)で表わす。また質問に対する答を最終的に求めている地点を$S_T$で表わす。

## 3. 巡回型質問の処理方式とデータ圧縮

本章では，一般化準結合操作を用いた巡回型質問の処理方式及び通信コストの低減を目的としたデータ圧縮手法について述べる[KAMBY8110]。

巡回型質問の処理方式の概要を以下に示す。ただし，ここでは$R_T=\{R_T\}$とし，目的関係$R_T$において答の部分を同定するまでを考察の対象とし，答を$S_T$へ転送することは考えない。

(1) 質問グラフにおいて$R_T$を根とする全域木を求める。

(2) 全域木に属さない枝をすべて全域木に埋め込む。

(3) 全域木の枝とそれに埋め込まれた枝のラベルの集合和を求め，それを全域木の枝の新しいラベルとする。

(4) 全域木において$R_i$を葉，$R_j$をその親とし，$R_i$と$R_j$を結ぶ枝のラベルをXとするとき，一般化準結合操作 $R_j = R_j \underset{X}{\boxtimes} R_i$を実行し，全域木から節点$R_i$を除去する。

(5) 全域木が節点 $R_T$ のみになるまで上記(4)を繰り返す。

次に例を示す。

[例1]　図1(a)に示す質問グラフを持つ巡回型質問について考える。全域木として図1(b)の実線部で示された枝を選んだとすると，全域木に属さない枝（図1(b)では点線で示されている）を全域木に埋め込み（図1(c)参照），ラベルの集合和を求めることにより，図1(d)に示す全域木を得る。以後，この全域木に沿って一般化準結合 $R_2 \underset{CD}{\boxtimes} R_3$, $R_1 \underset{BD}{\boxtimes} R_2$, $R_T \underset{AD}{\boxtimes} R_1$ を順に実行することにより，$R_T$ の答の部分を求めることができる。□

一般化準結合では通常の準結合操作に比べ，余分の属性の値を転送する必要が生じる。上記の例では，通常の準結合操作に比べ属性Dの値を全域木に沿って関係 $R_3$ から $R_T$ まで転送する必要があり，その分だけ転送データ量が増加する。したがって

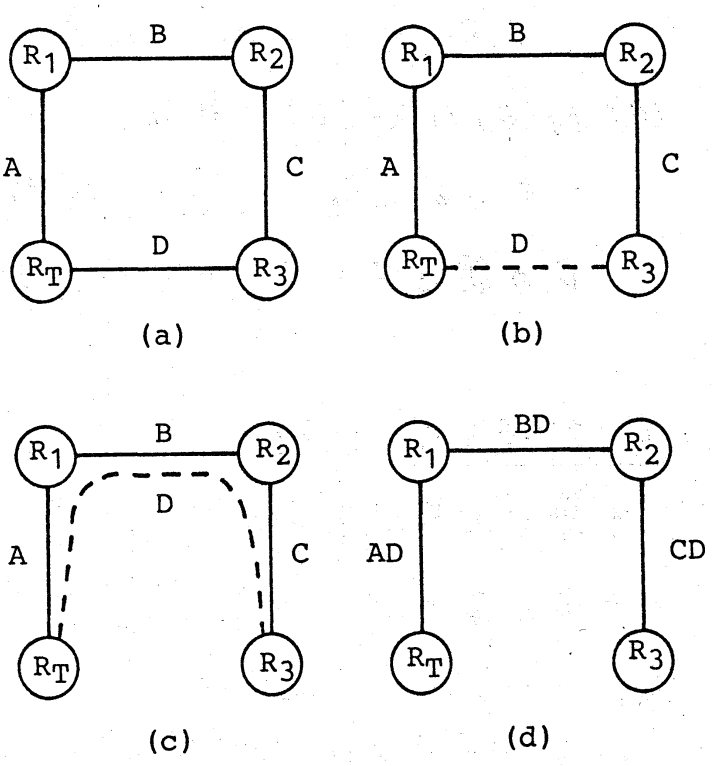


図1　巡回型質問の処理

転送データ量削減のため，次のようなデータ圧縮を行なう。すなわち，一般化準結合操作を実行する前に関係$R_3$と$R_T$の間で属性$D$の値を交換し，$R_3[D] \cap R_T[D]$の部分のデータ値をすべて数値に置き換える。より厳密には，次の手順に沿って属性値の交換を行なう。ここでは$|R_3[D]| \leq |R_T[D]|$の場合を示すが，逆の場合は$R_3$と$R_T$をすべて入れ換えればよい。

(1) $S(R_3)$から$S(R_T)$に$R_3[D]$を転送する。

(2) $S(R_T)$において$R_3[D] \cap R_T[D]$を求める。

(3) $R_3[D]$の各値を1ビットに対応させたビットベクトルにおいて，$R_3[D] \cap R_T[D]$の部分を1，他を0とし，それを$S(R_T)$から$S(R_3)$に転送する。

(4) $R_3[D] \cap R_T[D]$の部分のデータをソートし，それを小さいものから順に1, 2, …と数値符号化する．以後，属性$D$の値はすべてこの数値によって表現する。

## 4. 処理方式の比較

本章では単純巡回型質問を解くための基本的な処理方式を考え，その通信量を比較する。

図2に示す単純巡回型質問を考える。$\mathbb{R}_T = \{R_T\}$とし，前章

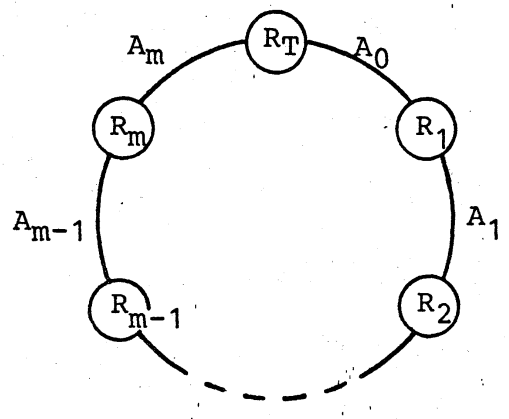


図2　単純巡回型質問

と同様に$S(R_T)$において答を得ることを目的とする。また各関係は2項関係であるとし，簡単のため各関係の組数は$n$，各属性のフィールド長は$d$(ビット)，選択度(selectivity)[CHEUN80]は$p$とし，これらはすべての関係にわたって同一であるとする。このような条件のもとで次のような処理方法を考える。

方法1：$m$個の関係$R_1, \ldots, R_m$を直接$S(R_T)$に転送する。

方法1'：隣接する関係間で属性値を交換し，データ圧縮を行なった後，$m$個の関係$R_1, \cdots, R_m$を直接$S(R_T)$に転送する。

方法2：$R_T$を根とする質問グラフの全木$R_T - R_m - \cdots - R_1$を考え，全域木に沿って$R_1$から$R_T$まで順に一般化準結合操作を繰り返す。

これらのうち，方法1はいわゆるIFS(Initial Feasible Solution)[CHEVNY7905]であり，巡回型質問を解くための最も単純な方法である。方法1'は方法1に前処理としてデータ圧縮を加え改良したものであり，方法2とともに我々がここで提案するものである，次にこれらの方法のデータ通信量を求める。

〈方法1〉 データ通信量を$T_1$とすると，$T_1 = 2mnd$となる.

〈方法1'〉 前処理のためのデータ圧縮手法としては前節で述べた方法を用いる。関係$R_i$と$R_{i+1}$ $(i=1, \cdots, m-1)$

の間でデータ交換を行なうための通信量は
$(m-1)nd+(m-1)n$ である。この前処理により，関係 $R_2,\cdots,R_{m-1}$ は組数が $p^2n$ に，関係 $R_1, R_m$ は組数が $pn$ になる。また属性 $A_1,\cdots,A_{m-1}$ の値はすべて数値符号化されているため，1つのデータを表現するためには高々 $\log pn$ ビットで十分である。したがって全体の通信量を $T_1'$ とすると

$$T_1' = (m-1)nd+(m-1)n+2(m-2)p^2n\log pn + 2pn(d+\log pn) \quad (1)$$

となる。$\log pn=\delta$ と置くと (1) 式は

$$T_1' = n\{(m-1)(d+1)+2(m-2)p^2\delta+2p(d+\delta)\} \quad (2)$$

と書くことができる。

〈方法2〉 まず，関係 $R_T$ と $R_1$ の間で属性 $A_0$ の値を交換するために $nd+n$ の通信量を必要とする．次に $R_1$ を $S(R_2)$ に転送するためには $pn(d+\log pn)$ $=pn(d+\delta)$ の通信量を要する．$S(R_2)$ では $R_1$ と $R_2$ の結合を行なうが，最悪の場合はそれにより組数が非常に増加する。そこで結合によって落ちる組を2つの関係から取り除き残った組の属性 $A_1$ の値をすべて数値符号化するにとどめ，実際には結合を行なわない。この操作により，

関係$R_1$, $R_2$の大きさはそれぞれ$2p^2n\delta$, $p^2n(d+\delta)$となり，したがって$S(R_2)$から$S(R_3)$への通信量は$p^2n(d+3\delta)$となる。以下，同様にして一般に$S(R_i)$から$S(R_{i+1})$ $(i=1, \cdots, m$，ただし$R_{m+1}=R_T$とする。$)$への通信量は$p^i n\{d+(2i-1)\delta\}$となる。したがって方法2による全通信量は

$$T_2 = nd + n + np(d+\delta) + \cdots + np^m\{d+(2m-1)\delta\}$$
$$< nd + n + np(d+\delta) + \cdots$$
$$= n\left\{1 + \frac{d}{1-p} + \frac{p(1+p)\delta}{(1-p)^2}\right\} \quad (3)$$

となる。

$n=10,000$として$T_1$, $T_1'$, $T_2$の比較を行なう。$T_1$と$T_1'$に関しては，$d \geq 4$バイトのときは必ず$T_1 > T_1'$が成立する。また$d=3$バイト，2バイト，1バイトのときは，それぞれ$p \leq 0.94$, $p \leq 0.76$, $p \leq 0.66$において$T_1 > T_1'$が成立する。このことからほとんどすべての場合において方法1よりも方法1'の方がデータ転送量が少ないと言える。

そこで次に$T_1'$と$T_2$の比較を行なう。$p \leq 0.29$においては$T_1' > T_2$となり，また$m=2$の場合を除けば$p \leq 0.48$において$T_1' > T_2$となる。これらの場合は必ず方法2の方が通信量が少なくなる。その他の場合は一般に$m$, $d$の値が大きく，$p$の値が小さい程，方法2の方が有利となると言える。$T_2$の計算

に関しては最悪の場合を見積っているために，実際に方法2が有利となる条件はより広いものと思われる。通常の質問処理では前処理として各関係に対して制約操作を施すため，選択度 $p$ は一般に小さな値を持つと考えられる。以上より，ほとんどの場合は方法2の方がデータ通信量が少ないが，特殊な状況（たとえば $m=2$, $p\simeq 1$ など）のもとでは方法1″の方が有利であると言える。したがって状況に応じて方法1″と方法2を使い分ける必要がある。

## 5. 全域木の選択

前章で行なった比較より，通常の場合単純巡回型質問は図3(a)に示す形の全域木（図中，●印は根を表わす）に基づいて処理を行なえば選択度の効果が蓄積して行くために全体のデータ通信量が小さくなるという結果が得られた。一般の質問グラフは複数個の閉路を含むが，全域木として深さ優先による全域木（depth-first spanning tree）を選べば，すべての閉路を図3(a)の形にすることができる[AHU-H74]（ただしこの場合，図中●印

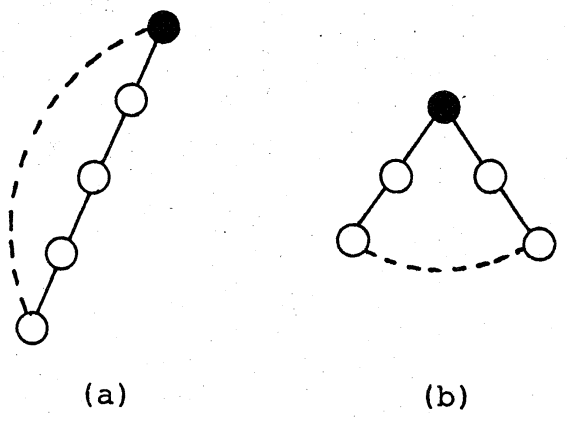


図3 全域木の選択

は根あるいは根に最も近い節点を表わし，点線は逆辺（back edge）を表わす。）したがって，深さ優先による全域木に基づけば，すべての閉路に関して選択度の蓄積を最大とすることができる。ところが，場合によっては全域木に属さない枝（全域木に埋め込む必要のある枝）にラベルとして多くの属性を持つものが残る可能性があり，その場合には全域木に沿って結合属性以外の多くの属性のデータを転送する必要が生じ，必ずしも効率的な処理とはならない。例を示す。

［例2］　図4(a)の質問グラフに基づいて考える。図中，●印の関係は$R_T$を表わす。図4(b)に示される深さ優先の全域木を考えた場合，全域木に属さない枝のラベルはABDである。したがって質問処理のためには属性A, B, Dの値を全域木に沿って$R_4$から$R_T$まで送する必要がある。□

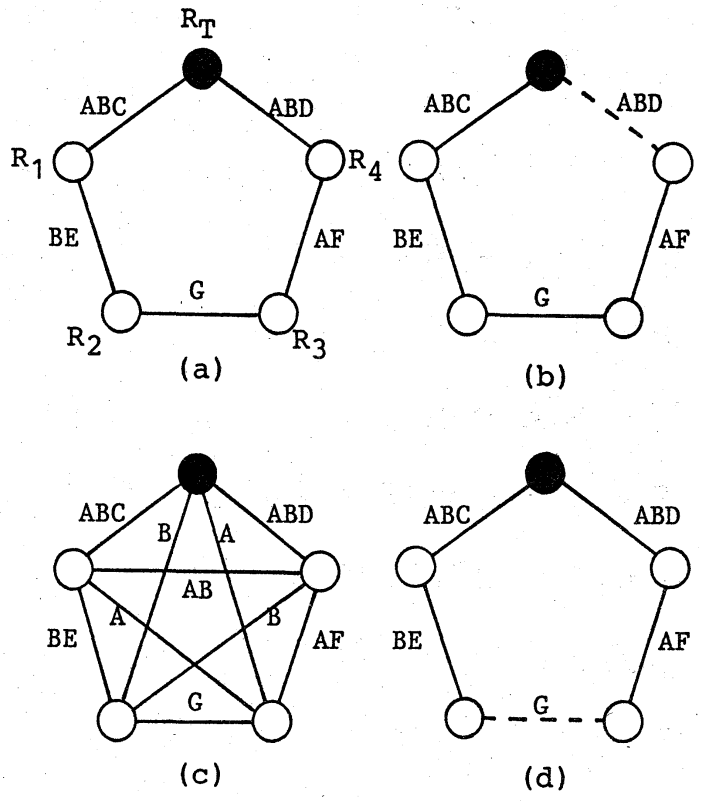


図4　質問グラフの深さ優先の全域木および極大全域木に基づく質問処理

深さ優先の全域木には上述のような問題点がある。そこで全域木に埋め込む必要のある枝数及びラベル数をなるべく少なくするために，各枝のラベル数をその枝の重みと見なした場合の質問グラフの推移的閉包 $G_g^+$ の極大全域木を考える[BERNG79/2]。次の命題が成立する。

[命題1]　巡回型質問の処理において，質問グラフ $G_g$ の全域木として推移的閉包 $G_g^+$ の極大全域木を選択すれば，全域木に埋め込む必要のあるラベル数は最小となる。　□

ただし命題1では同じ属性名でも異なる枝にあれば異なるラベルとして考えるものとする．以上の結果より，全域木として深さ優先による全域木及び $G_g^+$ の極大全域木を考えた場合，両者の間には選択度の蓄積及び全域木に埋め込む必要のあるラベル数に関してトレードオフが存在することがわかる。ところが全域木として $G_g^+$ の極大全域木を選択した場合でも，次のような処理方法により選択度の蓄積に関して深さ優先の全域木と同等の効果を得ることができる．既ち，$G_g^+$ の極大全域木に基づく場合は，質問グラフ中の閉路が図3(b)のような形で現われることがあるが，その場合には全域木の葉（或いは葉に最も近い節）から根（或いは根に最も近い節）に至る2つの経路のうち，一方の経路に沿った処理を先に行ない，次に根と他方の葉の間でデータを交換することにより最

初に処理を行なった経路の選択度の蓄積効果を他方に伝える。その後，他方の経路に沿って残りの処理を行なう。この方法によれば，実質的に閉路中のすべての関係を直列に並べ順に処理を行なう場合と同等の選択度の蓄積効果を得ることができる。

[例3]　図4(a)の質問グラフの推移的閉包およびその極大全域木をそれぞれ同図(c)，(d)に示す。全域木に埋め込む必要のあるラベル数は同図(c)の場合が3に対し，(d)の場合は1である。同図(d)の全域木に基づく処理としてはまず$R_2$から$R_T$に至る経路に沿って一般化準結合$R_1 \underset{BEG}{\boxtimes} R_2$，$R_T \underset{ABCG}{\boxtimes} R_1$を順に行ない，次にこの操作によって得られた選択度の蓄積効果を関係$R_3$に伝えるために，$R_T$と$R_3$の間で属性Gの値を交換する。その後はもう一方の経路に沿って$R_4 \underset{AFG}{\boxtimes} R_3$，$R_T \underset{ABDG}{\boxtimes} R_4$をこの順に実行する。□

## 6. 目的関係が複数の場合への拡張

前章までは，$|R_T| = 1$の場合のみを考察の対象として来たが，一般には目的関係が複数個存在する場合，或いは結合属性が目的属性となっている場合などを考える必要がある。

[例4]　4つの関係スキーマ$R_1[A,C]$，$R_2[B,C]$，$R_3[B,D]$，$R_4[C,D]$に対し，図5の質問グラフで与えられ

る巡回型質問を考える。目的属性を{A, B}とした場合，目的関係$R_T$としては{$R_1$, $R_2$}, {$R_1$, $R_3$}の2つの集合を考えることができる。□

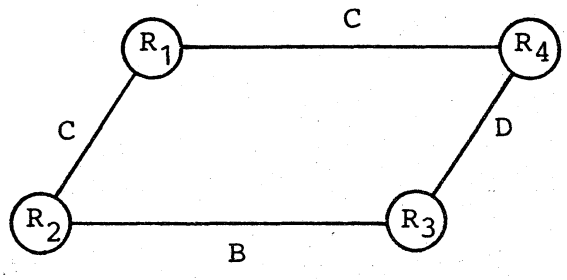


図5. 質問グラフ

上記の例から明らかなように，結合属性が目的属性となっている場合は目的関係$R_T$が一意には決まらない。目的属性の集合が与えられたとき要素数最小の$R_T$を求める問題は集合の最小被覆問題そのものであるが，$R_T$を求める場合はさらに$R_T$に属する関係が質問グラフ中で互いに近い位置に存在すべきであるという条件を加えることができる。例えば[例4]では$R_T$として{$R_1$, $R_2$}, {$R_1$, $R_3$}が考えられるが，前者の場合は質問グラフにおいて2つの関係が隣接しているのに対し，後者ではそうなっていない。最終的に答を$S_F$に転送することを考えると，$R_T$={$R_1$, $R_3$}の場合では，$R_1$と$R_3$の関連を知るために$R_2$或いは$R_4$も$S_F$に転送する必要がある。データ転送量削減のためにはこの方法は得策ではなく，したがって[例4]の場合は$R_T$={$R_1$, $R_2$}とすべきである。

## 7. データ圧縮による前処理の一般化

本章では，第3章で述べたデータ圧縮による前処理の手法の一般化を行なう。図6に示す巡回型質問を考える。$R_T=\{R_1\}$とし，図中の数字は前処理として各関係に対する制約操作を実行した後の各属性の異なる属性値の数を示す。図からわかるようにこの例では$R_3$の組数が小さい。このような場合は$R_5$から$R_1$に向かい順に一般化準結合操作を実行する処理手法よりも，以下に示すような順で処理を行なった方が全体の処理コストが小さくなると考えられる。

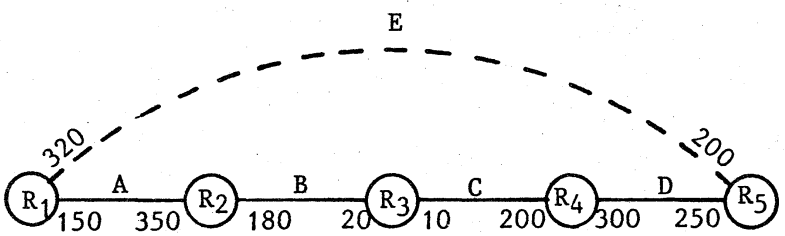


図6　データ圧縮による前処理の一般化

(i) $R_4 \ltimes R_3$　(ii) $R_5 \ltimes R_4$　(iii) $R_1 \ltimes R_5$

(iv) $*\ R_5 \ltimes R_1$　(v) $*\ R_4 \underset{DE}{\boxtimes} R_5$　(vi) $*\ R_3 \underset{CE}{\boxtimes} R_4$

(vi) $R_2 \underset{BE}{\boxtimes} R_3$　(vii) $R_1 \underset{AE}{\boxtimes} R_2$

*印を施した処理は，2章で述べたビットベクトルによるデータ圧縮手法を用いたものであり，そのデータ転送量は他の処理に比べてかなり少なくなる。より一般的には組数の小さい関係から処理を開始すべきである。

## 8. 仮性関数従属性の利用

### 8.1 巡回型質問の木型質問への変換

図7(a)の質問グラフ(同図(b)の質問グラフと等価)で表わされる巡回型質問について考える。今，関係$R_2$あるいは$R_3$においてSFD: D→Cが成立すると仮定する。このとき準結合 $R_3 \underset{CD}{\ltimes} R_2$ 及び $R_2 \underset{CD}{\ltimes} R_3$ を実行することにより，関係$R_2$及び$R_3$の地点において $R_2[CD] \cap R_3[CD]$ の部分を同定できる。$R_2[CD] \cap R_3[CD]$ ではSFD: D→Cが成立するため，これ以後，関係$R_2$と$R_3$の間では属性Dの値を転送するだけで十分である。すなわち図7(b)の質問は図7(c)の質問と等価となる。ところが図7(c)の質問は図7(d)の質問と等価であり，したがって木型質問である。すなわち，関係表において成立するSFDを利用することにより巡回型質問を木型質問に変換でき，準結合操作のみを用いて解くことが可能となる。

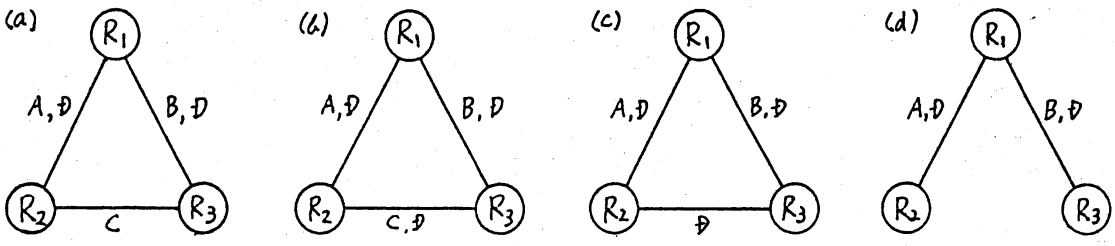


図7 SFDを用いた巡回型質問の変換

## §.2 転送データ量の削減

SFDを利用することにより，転送データ量を削減できる場合がある。関係$R_1$と$R_2$を属性A,Bで結合する場合を考える。この結合を準結合操作を用いて処理するためには$R_1 \underset{AB}{\ltimes} R_2$，$R_2 \underset{AB}{\ltimes} R_1$を実行する必要がある。ところが関係$R_2$においてSFD：$A \to B$が成立する場合は，$R_2$に対しては属性Aの値を指定するだけで属性Bの値が一意に決まる。したがって先の準結合操作では$R_2 \underset{AB}{\ltimes} R_1$を実行するかわりに$R_2 \underset{A}{\ltimes} R_1$を実行するだけで十分である。一般には質問で指定された関係において複数個のSFDが成立する場合が考えられるが，必ずしもすべてのSFDを転送データ量の削減に利用することはできない。例えば図8に示す質問グラフにおいてC→Aあるいは C→Bをそれぞれ単独に利用することは可能であるが，両者を同時には利用できない。

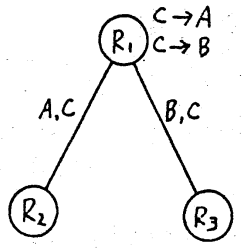

図8 質問グラフ

また結合属性をA,B,Cとし，各属性間にSFD：$A \to B$, $B \to C$が成立する場合は，転送する関係を2つに分解した方が一般にはデータ量が削減される。

## §.3 SFDの検出

実際の関係表においてSFD：$A \to B$が成立していることを検出するためには，関係表を属性Aの値でソートし，属性A

の一つの値に対しては属性Bの値がただ一つ決まることを調べる必要があるが，これは非常に時間のかかる処理となる。そこでSFDの検出のためには，まず圧縮関係［KAMBYのもの］などを用いてその成否を調べ，圧縮関係上で成立したSFDのみに関し，実際の関係上で検出操作を実行するなど，処理時間削減のための工夫が必要である。

謝辞　日頃，御討論頂く矢島研諸氏に深謝する。なお，本研究は一部，文部省科学研究費による。

参考文献

[AHO-H74] Aho,A., Hopcroft,J. and Ullman,J., The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.

[BERNC8101] Bernstein,P.A. and Chiu,D.M., "Using Semi-Joins to Solve Relational Queries", JACM, Vol.28, No.1, pp.25-40, Jan. 1981.

[BERNG7912] Bernstein,P.A. and Goodman,N., "Full Reducers for Relational Queries Using Multi-Attribute Semi-Joins", IEEE Computer Networking Symposium, pp.206-215, Dec. 1979.

[CHIUB81] Chiu,D.M., Bernstein,P.A. and Ho.Y.C., "Optimizing Chain Queries in a Distributed Database System", Harvard Rep., 1981.

[CHIUH8005] Chiu,D.M. and Ho.Y.C., "A Methodology for Interpreting Tree Queries into Optimal Semi-Join Expressions", ACM-SIGMOD'80, pp.169-178, May 1980.

[HEVN 80] Hevner,A.R., "The Optimization of Query Processing on Distributed Database Systems", Ph.D Diss., Tech. Rep. DB-80-02, Dept. of Computer Sci., Purdue Univ., 1980.

[HEVNY7905] Hevner,A.R. and Yao,S.B., "Query Processing in Distributed Database Systems", IEEE Trans. on Software Engineering, Vol.SE-5, No.3, pp.177-187, May 1979.

[KAMBY8109] 上林弥彦, 吉川正俊, 矢島脩三 "分散型データベースシステムにおける圧縮関係の利用", 信学技報, AL81-53, 1981年9月.

[KAMBY8110] 上林弥彦, 吉川正俊, 矢島脩三 "一般化準結合を用いた質問処理について" 信学技報, AL81-70, 1981年10月.

[ROTHB8003] Rothnie,J.B.Jr., Bernstein,P.A., Fox,S., Goodman,N., Hammer,M., Landers,Y.A., Reeve,C.L., Shipman,D.W. and Wong,E., "Introduction to a System for Distributed Database(SDD-1)", ACM Trans. on Database Systems, Vol.5, No.1, pp.1-17, Mar. 1980.

[SMITB8105] Smith,J.M., Bernstein,P.A., Dayal,U., Goodman,N., Landers,Y., Lin,K.W.T. and Wong,E., "Multibase --- Integrating Heterogeneous Distributed Database Systems", AFIPS NCC'81, Vol.50, pp.487-499, May 1981.