数理解析研究所講究録
第591巻 1986年 84-94
84

データベース問合せ言語の変換アルゴリズムについて

On Optimization in Query Translation from AQL to SQL

広島大学　工学部　　宮尾　淳一 (Jun'ichi MIYAO)
冨永　一幸 (Kazuyuki TOMINAGA)
菊野　亨　(Tohru KIKUNO)

1．　まえがき

筆者らは，現在，データベースシステムAIDE-Ⅱ (An Intelligent Database System for End Users-Ⅱ) を開発中である．このシステムでは従来の関係データベースとエンドユーザの間に意味データモデルを導入し，その意味データモデル上で問合せを記述する(3)．これにより，データベーススキームに関する知識を持たなくても問合せの作成ができる．

更に，筆者らが開発した表形式言語ＡＱＬを利用すれば，問合せを視覚的に作成することが可能となる(2)．言語ＡＱＬの特徴的な機能にユーザビューがある．これは機能的には関係データベースにおけるビューと同様であるが，ユーザビューの場合には予め定義しておく必要なく自由に指定できるという特長がある．そのため，エンドユーザでも容易にユーザビューを利用して問合せを記述することができる．この問合せは関係データベース上の問合せに変換された後，通常の問合せ処理の手順に従って実行される．ところが，この問合せの変換は一義的に定まらず，一般に，あいまいさを含む．本稿では，このあいまいさを実用性の立場から解消する試みについて議論する．

2．ユーザビュー

問合せ言語ＡＱＬ[2]はＱＢＥ[5]と同じ表形式言語であり，そのシンタックスはＱＢＥと同様である．関係データベースに対する一般的な機能である問合せ作成，ビュー定義，データ定義[4]などはＡＱＬにも含まれている．ＡＱＬでは，エンドユーザでも容易に問合せを作成できる機能としてユーザビューを新たに提供している．

［例１］　関係データベース内に存在する次の３つの関係について考える．

MEMBERS(#MEMBER, NAME, ADDRESS)

ORDERS1(#ORDER1, #MEMBER1, ITEM1, QUANTITY1)

ORDERS2(#ORDER2, #MEMBER2, ITEM2, QUANTITY2)

意味データモデル上で値域 number, name, address, itemが定義され，値域と属性との対応も与えられているとする．ここでは，特に，値域 itemに２つの属性 ITEM1, ITEM2が対応すると仮定する．今，次の問合せQ_1を考える．

Q_1＝"PC-9801を注文した人の名前と住所を求めよ".

このQ_1はＡＱＬのユーザビューを用いると，図１のように記述できる．ユーザビューでは表の左側に必要な値域の名前を列挙し，右側に値域が満たすべき条件を記述する．

	USER VIEW
name	
address	
item	PC-9801

図１　ユーザビューによる問合せQ_1の記述

問合せQ_1に対応する関係データベースの問合せの例を図2のQ_{11}, Q_{21}に示す．同図で，"_"は変数を表し，"P."は出力すべき属性を指定している．与えられたユーザビューQ_1に基づいて問合せQ_{11}, Q_{21}を構成する手順の詳細については3.で述べる．ここではその基本方針について直観的な説明を行う．

（A）先ず，ユーザビュー中の各値域名に対応する属性名を求める．同時に，関係データベース内に存在するその属性名を含む関係名の集合$\mathcal{R}_1$, $\mathcal{R}_2$, ・・・ を求める．

（B）次に，各$\mathcal{R}_i$に対し，$\mathcal{R}_i$中のすべての関係を結合する問合せQ_iを生成し，各Q_iの結合結果の和をとる．なお，余りに多くの結合を必要とするQ_iは予め削除する．

この例の場合には，ステップ(A)によって，

$\mathcal{R}_1$={ MEMBERS, ORDERS1 }, $\mathcal{R}_2$={ MEMBERS, ORDERS2 }

が得られる．次に，ステップ(B)によって，図2に示すQ_{11}, Q_{21}が構成される． □

MEMBERS	RETRIEVE
#MEMBER	_X
NAME	P.
ADDRESS	P.

ORDERS1	RETRIEVE
#ORDER1	
#MEMBER1	_X
ITEM1	PC-9801
QUANTITY	

Q_{11}

MEMBERS	RETRIEVE
#MEMBER	_X
NAME	P.
ADDRESS	P.

ORDERS2	RETRIEVE
#ORDER2	
#MEMBER2	_X
ITEM2	PC-9801
QUANTITY	

Q_{21}

図2　関係データベースの問合せQ_{11}とQ_{21}

3．変換手順

ここでは，ユーザビューで記述された問合せに対する変換手順の概要を述べる．但し，2．で説明した基本方針のステップ(B)についてのみ述べる．対象となる関係データベース$\mathcal{R}$と各関係R_iを構成する属性名の集合は与えられているとする．

［定義1］　関係データベース$\mathcal{R}$に対する結合グラフを無向グラフ$G=(V, E)$と定める．ここで，$V=\mathcal{R}$とし，$E=\{(R_i, R_j) \mid$ 2つの関係R_i，R_j $(\in \mathcal{R})$は，共に，ある属性Aを含んでいる$\}$とする．このとき，各無向枝 $(R_i, R_j) \in E$にはラベルAを付ける．以降では，このラベルを$\ell(R_i, R_j)$で参照する．　□

関係データベース上の結合操作のみを用いた問合せは結合グラフGの部分グラフとして表現できる．この性質を利用した変換の概要を次に示す．なお，与えられるユーザビューに現れる(値域名に対応する)属性名をすべて含む関係の集合をV_0で表し，ユーザビューで必要とされる関係の集合と呼ぶ．この集合V_0は2．のステップ(A)を適用して既に求まっていると仮定する．

ステップ1：結合グラフG上の連結な部分木であって，しかも，集合V_0の要素のすべてを節点として含む木を列挙する．今，列挙された木の集合を$\mathcal{T}$で表す．

ステップ2：$\mathcal{T}$中に含まれる木の枝が結合操作に対応しているので，$\mathcal{T}$に対し共通な枝を1つにまとめるという最適化を行う(問題JM)．問題JMの解として，結合処理に必要なページフェッチの回数が最小となる集合$\mathcal{T}'$を$\mathcal{T}$から構成する．

ステップ3：$\mathcal{T}'$中の木の各枝に対応する結合処理を実行し，それらの集合和をとって最終的に問合せの結果を求める．

筆者らは，既に，ステップ１を実用的時間内で解くアルゴリズムＪＴを提案している．シミュレーション実験によって，実用的環境下ではステップ１が1.5秒以内で解けることを確認している(2)．使用したコンピュータは日本・データゼネラル社のECLIPSE MV/4000である．

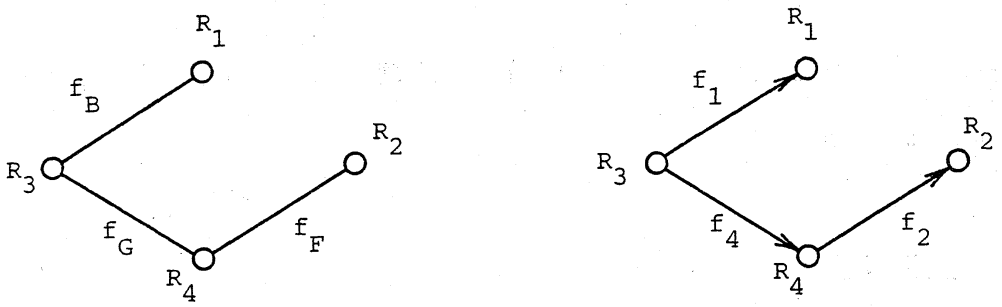
４．結合回数の最小化

関係データベースにおける結合は最も時間のかかる基本操作である．従って，問合せの処理に当たっては結合回数の減少を図ることが重要となる．但し，共通な結合操作を１度実行しておいてその結果を複数回参照する場合，必要なディスクのページフェッチの総数が増大する危険性がある．以下では，文献（１）に従って各結合木の処理に必要なページフェッチの数を見積り，それに基づいて最適化問題を定式化する．

４．１　処理コスト

変換のステップ１の出力である各結合木$T=(V_T, E_T)$に対して，（１）各節点$R_i \in V_T$が表す関係中に含まれるタプル数N_iとその格納に必要なディスクのページ数M_i，（２）各無向枝$(R_i, R_j) \in E_T$，$\ell(R_i, R_j)=\alpha$に対して，R_iとR_jに対する選択率f_α（２つの関係R_i，R_jの結合結果に含まれるタプル数の割合），が与えられているとする．

次に，結合木$T=(V_T, E_T)$とT上の任意の節点$r \in V_T$に対し，rを根とし，T上の各枝に根から葉への方向を与えて求まる有向木を$TR_r(T)=(V'_T, E'_T)$と表す(図３参照)．なお，各有向枝$(R'_i, R'_j) \in E'_T$に対しては，対応する無向枝$(R_i, R_j) \in E_T$と同じ値の選択率を与える．従って，図３において$f_1=f_B$，$f_2=f_F$，$f_4=f_G$となる．



(a) $T=(V_T, E_T)$　　(b) $TR_{R3}(T)=(V'_T, E'_T)$

図３　結合木Ｔと有向木$TR_{R3}(T)$

ここでは，各結合木に対する実行形式として文献(１)と同様，入れ子ループ法（nested loops method）を仮定する．今，$|V'_T|=n+1$とする．入れ子ループ法における一番外側のループをR_{i0}，一番内側のループをR_{in}と表す．文献(１)より，有向木$TR_r(T)$と系列$S=(R_{i1}R_{i2}\cdots R_{in})$が与えられたとき，$TR_r(T)$のＳに関する処理に必要なコストＣ(Ｓ)は次のように再帰的に定義される(1)．但し，系列Ｓに対し，各$R_{ij}\in V'_T$であり，かつＳ中の各節点$R_{ij}(1\leq j\leq n)$の現われる節点の順序は，有向木$TR_r(T)$上の枝の向きによって定まる節点上の半順序関係を満たすものとする．

(ｉ) $C(\Lambda)=0$　（Λは空系列）

(ii) $C(R_i)=f_iM_i$

(iii) $C(S_1S_2)=C(S_1)+T(S_1)C(S_2)$

但し，Ｔ(Ｓ)は次のように定義される．

(ｉ) $T(\Lambda)=1$　（Λは空系列）

(ii) $T(S)=f_{i1}N_{i1}f_{i2}N_{i2}\cdots f_{ik}N_{ik}$

（但し，$S=R_{i1}R_{i2}\cdots R_{ik}$）

１つの結合木$T=(V_T, E_T)$が単独に与えられたとき，コストＣ(Ｓ)が最小となる最適な結合順序Ｓは文献(１)のAlgorithm Ｂによって求めることができる．この最適な結合順序を$\mathcal{S}_T$と表すと，結合木

の集合$\mathcal{T}=\{T_1, T_2, \cdots, T_n\}$の処理コストは，$COST(\mathcal{T})=\sum_{T\in\mathcal{T}} C(\mathcal{S}_T)$と定義される．

４．２　問題ＪＭ

先ず，結合木の集合$\mathcal{T}$中に含まれる共通な結合を１つにまとめる変換ＣＪを定義する．

［定義２］　結合木の集合$\mathcal{T}$と木$T_k\in\mathcal{T}$上の無向枝(R_i, R_j)（$\ell(R_i, R_j)=\alpha$）に注目する．ラベル，及び，端点の節点名を含めて同じ枝(R_i, R_j)が$T_{k\ell}(\in\mathcal{T})$（$\ell=1,2,\cdots$）にも含まれているとする．このとき，$T_k$，$T_{k\ell}$（$\ell=1,2,\cdots$）上でそれぞれ枝$(R_i, R_j)$を短絡除去し，新しい結合木$T_{n+1}=(\{R_i, R_j\}, \{(R_i, R_j)\}), \ell(R_i, R_j)=\alpha$を$\mathcal{T}$に追加する操作を，ここでは枝$(R_i, R_j)$に対する変換ＣＪと定める．□

以上の準備の下に，結合回数の最小化問題ＪＭを次のように定式化する．

［問題ＪＭ］　ステップ１で求まった結合木の集合$\mathcal{T}$に対して，変換ＣＪを有限回適用して，$COST(\mathcal{T}')$が最小となる結合木の集合$\mathcal{T}'$を求めよ．□

［例２］　図４に示す$\mathcal{T}=\{T_1, T_2\}$を考える．今各T_i（$i=1,2$）に対するパラメータとして$N_1=1024$, $N_2=512$, $N_3=2048$, $N_4=4096$, $M_1=32$, $M_2=16$, $M_3=64$, $M_4=128$, 選択率$f_A=0.00025$, $f_B=0.01$, $f_C=0.005$, $f_D=0.0025$を考える．このとき，T_1とT_2上に存在する枝(R_2, R_3),

(R_1, R_4)に変換ＣＪが適用可能である．先ず，(R_2, R_3)に変換ＣＪを適用すると集合$\mathcal{T}'=\{T_1, T_2, T_3\}$が求まる．引き続き($R_1$, R_4)に適用すると集合$\mathcal{T}''=\{T_1, T_2, T_3, T_4\}$が求まる（図５参照）．このとき，$\mathcal{T}$, $\mathcal{T}'$, $\mathcal{T}''$の処理コストはそれぞれ次のようになっている．

$COST(\mathcal{T})=2.48*10^8$, $COST(\mathcal{T}')=1.74*10^8$, $COST(\mathcal{T}'')=3.54*10^9$

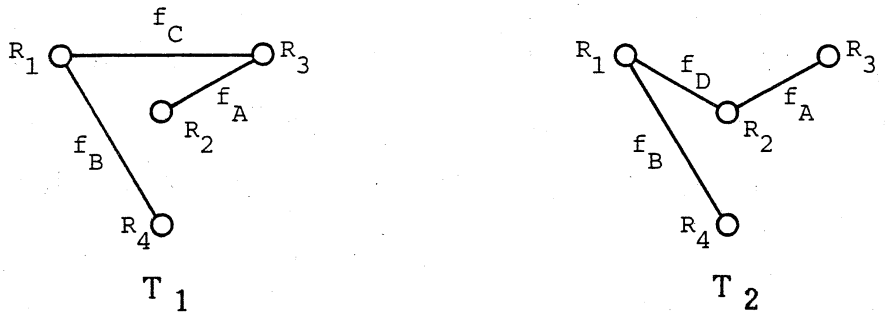
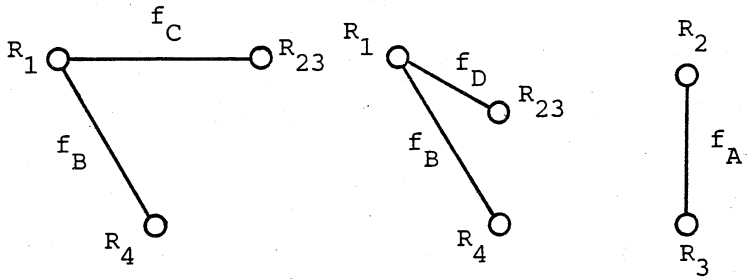
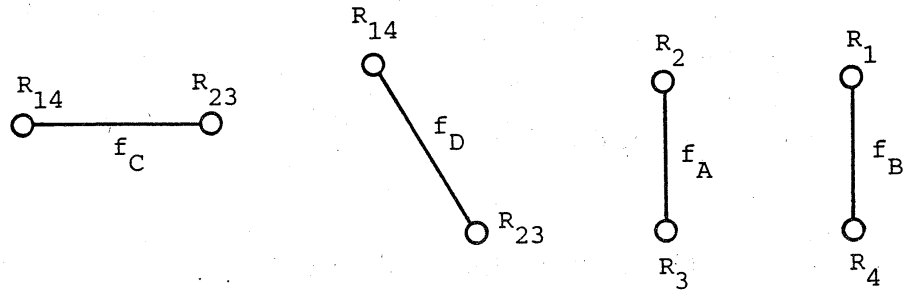


図４　結合木の集合$\mathcal{T}$



$\mathcal{T}'=\{T_1, T_2, T_3\}$



$\mathcal{T}''=\{T_1, T_2, T_3, T_4\}$

図５　結合木の集合$\mathcal{T}'$と$\mathcal{T}''$

結合回数だけ(すなわち，枝の総数だけ)に注目してみるとℑ"が最小であるが，ディスクのページフェッチの数まで考慮した場合はℑ'が最小となる．このように，結合回数の最小化だけでは必ずしも実際的な意味での最適化にまでは至らないことが分かる． □

４．３　ヒューリスティック解法

次に，問題ＪＭに対するヒューリスティックアルゴリズムＪＭを示す．アルゴリズム中で，$|(R_i, R_j)|, \ell(R_i, R_j) = \alpha$は，結合木の集合ℑ上で考えてラベルαの枝$(R_i, R_j)$を含む木の総数を表す．

［アルゴリズムＪＭ］

ステップ１：$\Im_{min} := \Im$，$C_{min} := COST(\Im)$，$A := \cup\{E_T | T = (V_T, E_T) \in \Im\}$と初期化する．

ステップ２：Ａに属する枝の中から$|(R_i, R_j)|$の値が最大の任意の枝を１つ選ぶ．いま，それを(R'_i, R'_j)とすると，$A := A - \{(R'_i, R'_j)\}$と更新する．もし，$|(R'_i, R'_j)| < 2$ならば，次はステップ４へ行く．

ステップ３：枝(R'_i, R'_j)に変換ＣＪを適用して，その結果をℑ'とおく．もしCOST(ℑ')$< C_{min}$ならば，$\Im_{min} := \Im'$，$C_{min} := COST(\Im')$，$\Im := \Im'$と更新する．次はステップ２へ行く．

ステップ４：$\Im_{min}$を出力する． □

上述のアルゴリズムＪＭの性能評価を行うため，日本・データゼネラル社のECLIPSE MV/4000上でＣ言語を用いてプログラムを開発した．各パラメータに対し一様乱数を用いてデータを生成し，シミュレーション実験を行った．図６の●にアルゴリズムＪＭで求めたCOSTを示

す．なお，同図の▲は(ページフェッチが結合回数に比例すると見なして結合回数の最小化を行った)文献(２)のヒューリスティックアルゴリズムで求めたCOSTを示している．この２つの結果を比較すると，結合回数だけでなく，ディスクのページフェッチも定式化の段階で考慮した方が問合せ処理コストのより一層の低減につながることが分かる．

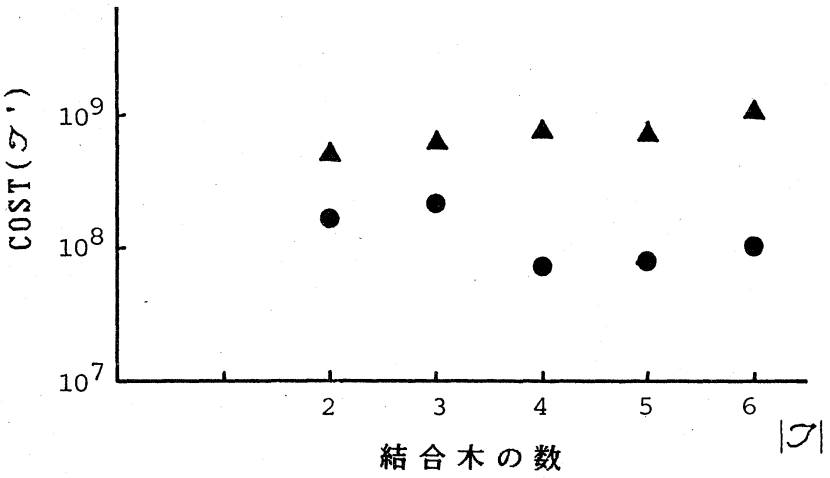


図６　シミュレーション実験

最後に，日頃ご指導頂いている広島大学工学部吉田典可教授，並びに，文献(１)に関してご教示頂いた京都大学工学部茨木俊秀教授に深謝します．

文献

(1) Ibaraki,T. and Kameda, T.:"On the optimal nesting order for computing N-relational joins," ACM Trans. Database Syst., 9, 3, pp.182-502 (1984).

(2) 宮尾,冨永,菊野,吉田:"表形式問合せ言語ＡＱＬからＳＱＬへの変換について," 信学技報，AL 85-66, pp.69-79 (1986).

(3) Sugihara, K., Miyao, J., Kikuno, T. and Yoshida, N.:"A semantic approach to usability in relational database systems," Proc. IEEE COMPDEC, pp.203-210 (1984).

(4) Ullman, J.D.:"Principles of Database Systems," 2nd Ed., Computer Sience Press (1982).

(5) Zloof, M.M.:"QBE/OBE:A language for office and business automation," Computer, 14, 5, pp.13-22 (1981).