関係データベースにおける関係演算の演算系とその実現方式†

清木 康†† 田中浩一††*
上林憲行†††* 相磯秀夫††

関係データベース・システムを実現する場合，関係演算の演算方式によってその基本的な性能は決定されてしまう．本論文は，関係に対する転置インデックスや関係間の結合テーブル等の補助データだけにより関係演算を遂行する，新たな演算系の提案である．従来，関係へのアクセス手段として補助的に用いられてきた補助データをこの演算系では関係演算遂行の中心的なデータとして用いている．本論文では，この演算系の基本データおよび基本演算を定義し，関係演算を基本演算の列に展開する方法を示す．さらに，この演算系を並列処理の環境で実現する一実現方式を示し，最後にこの実現方式および演算系の有効性を明らかにするためにこの実現方式と他の演算実行方式との間で性能評価を行った．

1. まえがき

関係モデル[1)~3)]はデータ独立性，アクセス対称性，非手続き的問合せを実現する高水準なユーザ・インタフェースを提供し，集合論に基づくデータ操作を完備した高水準なデータモデルである．しかし，関係データベース・システムを従来のフォンノイマン型計算機で実用化する場合，集合論に基づく演算の処理効率の悪さが問題となり，これに効果的に対処する関係演算方式およびそれを実現するアーキテクチャによる処理効率の改善が不可欠とされている．著者らはこうした背景から関係データベース・マシン（relational data base machine: RDBM）の研究を進めており，すでに効率よい関係演算方式[9)]と強力なデータステージング機構[10)]を柱とするRDBMのアーキテクチャ[12)]を発表している．関係データベースにおいては関係演算の実行方式によって基本的な性能が決定されてしまう．関係演算の実行方式としては，関係（リレーション）に対し直接に演算を施すことを基本とする方式と，転置インデックス[4),5)]等の補助データを用いて探索を高速化することを基本とする方式がある．前者の方式は関係が本来二次元表の形であるので並列処理を実現しやすいという利点があり，おもに並列プロセッサ構成のRDBM[6)~8)]で採用された．しかし，過去に開発されたこれらのRDBMシステムは，関係演算のなかでもとくに処理時間を多く費やす基本演算，すなわち，射影演算（projection），結合演算（join），割り算（division）等に対して，関係を構成している最小単位であるアイテム値に対する全数探索のアルゴリズムを用いて処理を行っており，十分な処理効率をあげるまでには至っていない．後者の方式は選択演算（selection）の高速化を実現する点では有効であるが，射影演算，結合演算，割り算等に対しての利用を効果的に行っておらず，さらに，補助データと並列処理とを組み合わせにくい点で処理効率の改善は十分ではない．

本論文は，転置インデックスや結合テーブル等の補助データに対して定義される数種の基本演算により関係演算を補助データだけで遂行する新たな演算系の提案である．関係を操作するためのアクセス手段として補助データを用いるのではなく，本演算系は補助データを中心としてこれに対する基本演算を遂行するために関係の方を補助的に用いるという特徴をもっている．

以下，本演算系の諸定義，関係データベースへの問合せの本演算系の基本演算列への展開，さらに本演算系を並列処理の環境で実現する一実現方式を示す．

最後に，この実現方式と，関係に対し直接に関係演算を並列処理の環境で施す方式との間で性能評価を行い比較検討する．

2. 演算系

ここで定義する関係演算系は，関係（リレーション）

† Operation System and Implementation Scheme of Relational Operations in Relational Data Base by YASUSHI KIYOKI, KOUICHI TANAKA (Department of Electrical Engineering, Faculty of Science and Technology, Keio University), NORIYUKI KAMIBAYASI (Faculty of Engineering, Hiroshima University) and HIDEO AISO (Department of Electrical Engineering, Faculty of Science and Technology, Keio University).

†† 慶応義塾大学理工学部電気工学科
††† 広島大学工学部第2類
* 現在 富士ゼロックス(株)
** 現在 ソニー(株)

内の有効な組（タプル）を識別するデータ，関係内の組の種別を示す転置インデックスに相当するデータ，および二つの関係間の組の対応関係を示す結合テーブル（リンク）に相当するデータを関係演算の対象データとして体系的に取り入れたものである．本演算系は以下に示す特徴をもつ．

（1） 補助データ（転置インデックス，結合テーブル），および補助データ間の演算により，問合せ実行途中の中間結果としての新しい関係を再構成する必要がない．たとえ，問合せが結合演算を含んでいる場合においても，結合演算の結果を示す中間結果としての関係は生成されない．

（2） 関係内の組を一意に識別する組織別子 TID (tuple identifier) を用いることにより，関係をカラム（属性に対応）単位に分割して処理できる．

（3） 従来，補助データとして用いられていた転置インデックス，結合テーブルを VTF (valid tuple flags) とよぶ TID の集合を示すデータを用いて表現することにより，それらを統一的に演算系内に組み込むことができる．

2.1 集合の諸定義

（1） 演算系には，α 集合，β 集合が存在する．

（2） α 集合は，関係の集合である．

（3） 関係 R_i は，組 $t_{ij}(j=1\sim n_i)$ を要素とする集合である．i：関係の識別子，n_i：R_i 内の組数，j：関係 R_i 内の組 t_{ij} を示す TID．

（4） 組 t_{ij} はアイテム値 d_{ijk} $(k=1\sim m_i)$ の並びである．m_i：R_i 内のカラム数，k：R_i 内のカラムを示す識別子．

$$t_{ij}=(d_{ij1}, d_{ij2}, \cdots, d_{ijm_i})$$

（5） カラム ${}_iC_k$ は，アイテム値 $d_{ijk}(j=1\sim p_i)$ の並びである．

$${}_iC_k(d_{j1k},\ d_{i2k}, \cdots, d_{in_ik})$$

（6） 関係 R_i 内における任意の l 個 $(1\leqq l\leqq m_i)$ のカラムの組合せを ${}_iE$ とする．すなわち，${}_iE$ は任意の l 個のカラムを R_i から抽出したものである．したがって，${}_iE$ は抽出された l 個のカラムについての複合アイテム値 $I_{ij}(l=1$ の場合は1アイテム値）の並びである．

$${}_iE=(I_{i1}, I_{i2}, \cdots, I_{ij}, \cdots, I_{in_i})$$

$$I_{ij}=(d_{ijk_1}, d_{ijk_2}, \cdots, d_{ijk_f}, \cdots, d_{ijk_l})$$

k_f：指定されたカラムの識別子

（7） ${}_iE$ 内の全要素 $I_{ij}(j=1\sim n_i)$ から重複を排除したときの複合アイテム値の集合を ${}_iE$ とする．

$${}_iE=\{I_{ij}\}$$

（8） β 集合は，VTF, VTFG, AIR, UIL の4種類の要素から成る．

（9） VTF^{R_i} は，関係 R_i 内の組の TID である $j(j=1\sim n_i)$ を要素とする集合である．

$$\mathrm{VTF}^{R_i}=\{j\}$$

VTF は関係 R_i 内の有効組の識別，選択演算 (selection), 制約演算 (restriction) の結果の保持，および削除 (delete) された組の識別等に用いられる．

(10) $\mathrm{UIL}^{\boldsymbol{D}}$ は ${}_iE$ を表現するデータである。$\mathrm{UIL}^{\boldsymbol{D}}$ は指定された任意の l 個のカラムについて重複を排除した後の複合アイテム値 ($l=1$ のときは1アイテム値) I_{ij} の集合 ${}_iE$ の各要素 I_{ij} を適当な順序（昇順，降順等）に並べた並びである。

$$\mathrm{UIL}^{\boldsymbol{D}}=(I_{ij_1}, I_{ij_2}, \cdots, I_{ij_h}, \cdots, I_{ij_p})$$

I_{ij_h}：UIL 内の h 番目の要素．

P：指定されたカラムについて重複を排除した後の複合アイテム値 ($l=1$ のときは1アイテム値）の個数．

$\boldsymbol{D}$：指定された l 個のカラムの名前の並び．

$\mathrm{UIL}^{\boldsymbol{D}}$ は，転置インデックスの索引項目の値を表現する．

(11) VTFG^{R_i} は $\mathrm{VTF}_q{}^{R_i}$ $(q=1\sim p)$ の並びである．VTFG^{R_i} の h 番目の要素 $\mathrm{VTF}_h{}^{R_i}$ は $\mathrm{UIL}^{\boldsymbol{D}}$ の h 番目の要素 I_{ij_h} をもつ組の TID の集合である．$\mathrm{UIL}^{\boldsymbol{D}}$ と VTFG^{R_i} の対応は2.2節の演算(5)によって付けられる．

$$\mathrm{VTFG}^{R_i}=(\mathrm{VTF}_1{}^{R_i}, \mathrm{VTF}_2{}^{R_i}, \cdots, \mathrm{VTF}_h{}^{R_i}, \cdots, \mathrm{VTF}_p{}^{R_i})$$

$\mathrm{UIL}^{\boldsymbol{D}}$ と VTFG^{R_i} の対は，転置インデックスと等価な情報をもつ．

(12) $\mathrm{AIR}^{R_i\cdot R_i'}$ は，VTFG^{R_i} と $\mathrm{VTFG}^{R_i'}$ の対である．ただし，両 VTFG に対応する $\mathrm{UIL}^{\boldsymbol{D}}$ は共通の $\mathrm{UIL}^{\boldsymbol{D}}$ であり，VTFG^{R_i} 内の h 番目の要素 $\mathrm{VTF}_h{}^{R_i}$ は $\mathrm{VTFG}^{R_i'}$ 内の h 番目の要素 $\mathrm{VTF}_h{}^{R_i'}$ と互いに対応している．

$$\mathrm{AIR}^{R_i\cdot R_i'}=(\mathrm{VTFG}^{R_i},\ \mathrm{VTFG}^{R_i'})$$
$$=((\mathrm{VTF}_1{}^{R_i}, \cdots, \mathrm{VTF}_h{}^{R_i}, \cdots, \mathrm{VTF}_p{}^{R_i}),$$
$$(\mathrm{VTF}_1{}^{R_i'}, \cdots, \mathrm{VTF}_h{}^{R_i'}, \cdots, \mathrm{VTF}_p{}^{R_i'}))$$

$\mathrm{AIR}^{R_i\cdot R_i'}$ は関係 R_i と R_i' の間の組の対応関係 (relationship) を示すものであり，結合テー

ブルと等価な情報をもつ．すなわち，$AIR^{R_i \cdot R_{i'}}$ は結合演算 (join) の結果を新しい関係を生成することなく保持するためのものである．

2.2 基本演算の諸定義

本演算系の演算は，[1] 関係を構成するカラム内のアイテム値の参照による補助データ (VTF^{R_i}, UIL^{D}, $VTFG^{R_i}$, $AIR^{R_i \cdot R_{i'}}$) の生成，[2] 補助データ間の演算に大別される．以下に示す演算の(1)～(6)は[1]，(7)～(9)は[2]に属する．各演算は次のような形式で記述される．演算名 (演算対象データ)→演算結果

θ: 比較演算子 ($=$, $\neq$, $>$, $<$, $\geqq$, $\leqq$)

e: アイテム値

j: 関係 R_i 内の TID

$VTF^{R_i}\text{-}n$, $VTFG^{R_i}\text{-}n$, $AIR^{R_i \cdot R_{i'}}\text{-}n$, $UIL^{D}\text{-}n$: n は各データの識別子．n が S1, S2, S3 のときは演算対象であることを示し，d のときは演算結果であることを示す．

(1) Selection (VTF^{R_i}-S1, ${}_i\boldsymbol{C}_k$, θ, "e")
$\rightarrow VTF^{R_i}\text{-}d$
$VTF^{R_i}\text{-}d=\{j \mid (d_{ijk}\theta \text{ "e"}) \wedge (j \in VTF^{R_i}\text{-S1})\}$

(2) Restriction (VTF^{R_i}-S1, ${}_i\boldsymbol{C}_{k_1}$, θ, ${}_i\boldsymbol{C}_{k_2}$)
$\rightarrow VTF^{R_i}\text{-}d$
$VTF^{R_i}\text{-}d=\{j \mid (d_{ijk_1}\theta d_{ijk_2}) \wedge (j \in VTF^{R_i}\text{-S1})\}$

(3) Generate-UIL (VTF^{R_i}-S1, ${}_i\boldsymbol{E}_1$)
$\rightarrow UIL^{D}\text{-}d$
$UIL^{D}\text{-}d=(I_{ij_1}, I_{ij_2}, \cdots, I_{ij_h}, \cdots, I_{ij_p})$
$I_{ij_q} \neq I_{ij_h}$ (q は任意)
$I_{ij_h} \in \{I_{ij} \mid (I_{ij} \in {}_iE_1) \wedge (j \in VTF^{R_i}\text{-S1})\}$

(4) Generate-VTF (VTF^{R_i}-S1, UIL^{D}-S1, ${}_i\boldsymbol{C}_k$)
$\rightarrow VTF^{R_i}\text{-}d$
$UIL^{D}\text{-S1}=(I_{i'j_1}, I_{i'j_2}, \cdots, I_{i'j_h}, \cdots, I_{i'j_p})$
$VTF^{R_i}\text{-}d=\{j \mid I_{ij} \in \{I_{i'j_1}, I_{i'j_2}, \cdots, I_{i'j_p}\} \wedge (j \in VTF^{R_i}\text{-S1})\}$

(5) Generate-UIL·VTFG (VTF^{R_i}-S1, ${}_i\boldsymbol{E}_1$, θ)
$\rightarrow UIL^{D}\text{-}d \cdot VTFG^{R_i}\text{-}d$
$UIL^{D}\text{-}d=(I_{ij_1}, I_{ij_2}, \cdots, I_{ij_h}, \cdots, I_{ij_p})$
$VTFG^{R_i}\text{-}d=(VTF_1{}^{R_i}\text{-}d, VTF_2{}^{R_i}\text{-}d, \cdots, VTF_h{}^{R_i}\text{-}d, \cdots, VTF_p{}^{R_i}\text{-}d)$
$VTF_h{}^{R_i}\text{-}d=\{j \mid (I_{ij}\theta I_{ij_h}) \wedge (j \in VTF^{R_i}\text{-S1})\}$

この演算により UIL^{D} と $VTFG^{R_i}$ が生成され，両者の対応がつけられる．

(6) Generate-UIL·AIR (VTF^{R_i}-S1, ${}_i\boldsymbol{C}_k$, θ ${}_{i'}\boldsymbol{C}_{k'}$, $VTF^{R_{i'}}$-S2)
$\rightarrow UIL^{D}\text{-}d \cdot AIR^{R_i \cdot R_{i'}}\text{-}d$
$UIL^{D}\text{-}d=(I_{ij_1}, I_{ij_2}, \cdots, I_{ij_h}, \cdots, I_{ij_p})$
$I_{ij_h} \in ({}_iC_k \cap {}_{i'}C_{k'})$
${}_i\boldsymbol{C}_k$: R_i の joining カラム (属性)
${}_{i'}\boldsymbol{C}_{k'}$: $R_{i'}$ の joining カラム (属性)
$AIR^{R_i \cdot R_{i'}}=(VTFG^{R_i}\text{-S1} \cdot VTFG^{R_{i'}}\text{-S2})$
$=((VTF_1{}^{R_i}\text{-S1}, \cdots, VTF_p{}^{R_i}\text{-S1}), (VTF_1{}^{R_{i'}}\text{-S2}, \cdots, VTF_p{}^{R_{i'}}\text{-S2}))$
$VTF_h{}^{R_i}\text{-S1}=\{j \mid (d_{ij_h}\theta I_{ij_h}) \wedge (j \in VTF^{R_i}\text{-S1})\}$
$VTF_h{}^{R_{i'}}\text{-S2}=\{j' \mid (d_{i'j'k'}\theta\ I_{ij_h}) \wedge (j' \in VTF^{R_{i'}}\text{-S2})\}$

(7) Multiply-AIR ($(UIL^{D}\text{-S1} \cdot VTFG^{R_i}\text{-S1})$, $AIR^{R_i \cdot R_{i'}}\text{-S1}) \rightarrow UIL^{D}\text{-}d \cdot VTFG^{R_{i'}}\text{-}d$
$VTFG^{R_i}\text{-S1}=(VTF_1{}^{R_i}\text{-S1}, \cdots, VTF_h{}^{R_i}\text{-S1}, \cdots, VTF_p{}^{R_i}\text{-S1})$
$AIR^{R_i \cdot R_{i'}}\text{-S1}=(VTFG^{R_i}\text{-S2}, VTFG^{R_i}\text{-S3})$
$=((VTF_1{}^{R_i}\text{-S2}, \cdots, VTF_r{}^{R_i}\text{-S2}), (VTF_1{}^{R_{i'}}\text{-S3}, \cdots, VTF_r{}^{R_{i'}}\text{-S3}))$
$VTFG^{R_{i'}}\text{-}d=(VTF_1{}^{R_{i'}}\text{-}d, \cdots, VTF_h{}^{R_{i'}}\text{-}d, \cdots, VTF_p{}^{R_{i'}}\text{-}d)$
$VTF_h{}^{R_{i'}}\text{-}d=(((VTF_h{}^{R_i}\text{-S1} \cap VTF_1{}^{R_i}\text{-S2}) \odot VTF_1{}^{R_{i'}}\text{-S3}) \cup ((VTF_h{}^{R_i}\text{-S1} \cap VTF_2{}^{R_i}\text{-S2}) \odot VTF_2{}^{R_{i'}}\text{-S3}) \cup \cdots \cup ((VTF_h{}^{R_i}\text{-S1} \cap VTF_r{}^{R_i}\text{-S2}) \odot VTF_r{}^{R_{i'}}\text{-S3}))$

ここで演算子$\odot$は次のように定義される．集合，A，Bに対して

$$A \odot B=\begin{cases}\{\phi\} & \text{if } A=\{\phi\} \\ B & \text{otherwise}\end{cases}$$

である。

$VTFG^{R_{i'}}\text{-}d$ と対応する $UIL^{D}\text{-}d$ は，$VTFG^{R_i}$-S1 と対応している UIL^{D}-S1 と等しい．

Multiply-AIR は関係 R_i の転置インデックス (UIL^{D}-S1・$VTFG^{R_i}$-S1) の UIL^{D}-S1 内の各要素（索引項目）が，R_i と $R_{i'}$ の間の組の結合状態を示している $AIR^{R_i \cdot R_{i'}}$ を介して関係 $R_{i'}$ のどの組と結合するかを求める演算であり，演算結果は

UIL^{D}-S1 の各要素とその各要素に対応する $VTF_h^{R_i'}$-d の集合 $VTFG^{R_i'}$-d によって示される．

(8) Multiply-VTFG ($(UIL^{D_1}$-S1, $VTFG^{R_i}$-S1), (UIL^{D_2}-S2, $VTFG^{R_i}$-S2))
$\rightarrow UIL^{D_3}$-d • $VTFG^{R_i}$-d

Multiply-VTFG は，2組の UIL^{D}•$VTFG^{R_i}$ の対の間の演算であり，結果として新たな UIL^{D}•$VTFG^{R_i}$ の対を生成する演算である．

$$VTFG^{R_i}\text{-S1}=(VTF_1^{R_i}\text{-S1}, \cdots, VTF_h^{R_i}\text{-S1}, \cdots, VTF_p^{R_i}\text{-S1})$$

$$VTFG^{R_i}\text{-S2}=(VTF_1^{R_i}\text{-S2}, \cdots, VTF_h^{R_i}\text{-S2}, \cdots, VTF_q^{R_i}\text{-S2})$$

$$VTFG^{R_i}\text{-}d=(VTF_1^{R_i}\text{-S1}\cap VTF_1^{R_i}\text{-S2}, VTF_1^{R_i}\text{-S1}\cap VTFG_2^{R_i}\text{-S2}, \cdots, VTF_1^{R_i}\text{-S1}\cap VTF_q^{R_i}\text{-S2}, VTF_2^{R_i}\text{-S1}\cap VTF_1^{R_i}\text{-S2}, \cdots, VTF_p^{R_i}\text{-S1}\cap VTF_q^{R_i}\text{-S2})$$

$$UIL^{D_1}\text{-S1}=(I_{ij_1}, I_{ij_2}, \cdots, I_{ij_h}, \cdots, I_{ij_p})$$

$$UIL^{D_2}\text{-S2}=(I'_{ij'_1}, I'_{ij'_2}, \cdots, I'_{ij'_h}, \cdots, I_{ij'_q})$$

$$UIL^{D_3}\text{-}d=(I_{ij_1}\oplus I'_{ij'_1}, I_{ij_1}\oplus I'_{ij'_2}, \cdots, I_{ij_1}\oplus I'_{ij'_q}, \cdots, I_{ij_p}\oplus I'_{ij'_q})$$

ここで⊕は，二つの複合アイテム値（またはアイテム値）を結合して，新たな複合アイテム値を作る演算子である。

$$(d_{ij_hk_1}, d_{ij_hk_2}, \cdots, d_{ij_hk_l})\oplus(d'_{ij'_hk'_1}, d'_{ij'_hk'_2}, \cdots, d'_{ij'_hk'_{l'}})\rightarrow(d_{ij_hk_1}, d_{ij_hk_2}, \cdots, d_{ij_hk_l}, d'_{ij'_hk'_1}, d'_{ij'_hk'_2}, \cdots, d'_{ij'_hk'_{l'}})$$

Multiply-VTFG は，カラム ${}_iC_{k_1}$, ${}_iC_{k_2}$, …, ${}_iC_{k_l}$ を索引項目としてもつ転置インデックス (UIL^{D_1}-S1, $VTFG^{R_i}$-S1) と，カラム ${}_iC_{k'_1}$, ${}_iC_{k'_2}$, …, ${}_iC_{k'_{l'}}$ を索引項目としてもつ転置インデックス (UIL^{D_2}-S2 • $VTFG^{R_i}$-S2) から，カラム ${}_iC_{k_1}$, …, ${}_iC_{k_l}$, ${}_iC_{k'_1}$, …, ${}_iC_{k'_{l'}}$ を索引項目としてもつ一つの転置インデックス (UIL^{D_3}-d, $VTFG^{R_i}$-d) を生成する演算である．

(9) Divide ($VTFG^{R_i}$-S1, $VTFG^{R_i}$-S2) $\rightarrow VTFG^{R_i}$-d

$$VTFG^{R_i}\text{-S1}=(VTF_1^{R_i}\text{-S1}, \cdots, VTF_h^{R_i}\text{-S1}, \cdots, VTF_p^{R_i}\text{-S1})$$

$$VTFG^{R_i}\text{-S2}=(VTF_1^{R_i}\text{-S2}, \cdots, VTF_q^{R_i}\text{-S2})$$

$$VTFG^{R_i}\text{-}d=(VTF_1^{R_i}\text{-}d, \cdots, VTF_h^{R_i}\text{-}d, \cdots, VTF_p^{R_i}\text{-}d)$$

$$VTF_h^{R_i}\text{-}d=\begin{cases} VTF_h^{R_i}\text{-S1}, & \text{if } \{(VTF_h^{R_i}\text{-S1}\cap VTF_1^{R_i}\text{-S2})\neq\{\phi\}\wedge(VTF_h^{R_i}\text{-S1}\cap VTF_2^{R_i}\text{-S2})\neq\{\phi\}\wedge\cdots\wedge(VTF_h^{R_i}\text{-S1}\cap VTF_q^{R_i}\text{-S2})\neq\{\phi\}\} \\ \{\phi\} & \text{otherwise} \end{cases}$$

$(h=1\sim p)$

$VTFG^{R_i}$-d と対応する UIL^{D} は，$VTFG^{R_i}$-S1 と対応している UIL と等しい．ここで，$VTF_h^{R_i}$-$d=\{\phi\}$ である $VTF_h^{R_i}$-d およびそれに対応する UIL^{D} 内の h 番目の要素 I_{ij_h} は削除される．

Divide は，割り算 (division) または限定作用素 $^{\forall}$(universal quantifier) により束縛された変数に対する演算のために用意されている．

3. 問合せ (query) の展開

関係データベースの問合せは，2章で定義した本演算系の演算を組み合わせることにより遂行できる．本章では，関係論理 (relational calculus)[2] に基づいて設計された ALPHA 言語[3]による問合せを図式化して表現する表記法 ALDD (ALPHA description diagram) を提案し，この ALDD で記述される基本的なダイヤグラムの形を本演算系の基本演算へ展開することにより，本演算系における問合せの遂行方法を示す．

3.1 ALDD の表記

ALDD は**表1**に示すような五つの要素から構成される．問合せを表現する ALDD の8種類の基本型とそれらの ALPHA による表現を**表2**に，また各基本

表1 ALDD の構成要素
Table 1 Elements of ALDD.

ALDDの構成要素	意味
	存在記号$^{\exists}$ (existential quantifier)によって束縛される組変数をもつ関係
	全称記号$^{\forall}$ (universal quantifier)によって束縛される組変数をもつ関係
	自由変数(free variable)である組変数をもつ関係
	二つの関係間の結合条件の指示子
	関係内にあるカラムに対する選択(selection)または制約(restriction)を示す指示子

表 2　ALDD 基本型と ALPHA による表現
Table 2　ALDD basic types and ALPHA representations.

基本型	ALDD表現	ALPHA表現	演算結果
1	A1 / R1(VTF)	GET W(R1.A1)	UIL^{A1}
2	A1,A2,…,An / R1(VTF)	GET W(R1.A1,R1.A2,…,R1.An)	$UIL^{A1\cdots An}$
3	A1='d1'…An='dn' / W1 / R1(VTF)	GET W(R1.W1): (R1.A1='d1' ∧ … ∧ R1.An='dn')	UIL^{W1}
4	A1=A2 / W1 / R1(VTF) R2(VTF)	GET W(R2,W1): ∃R1(R1.A1=R2.A2)	UIL^{W1}
5	A1=A2 / W1 W2 / R1(VTF) R2(VTF)	GET W(R1.W1,R2.W2): (R1.A1=R2.A2)	$UIL^{W1\cdot W2}$
6	A1=A2 A3=A4 / W1 / R1 R2 / A(2n−1)=A(2n) / W2 / R(n−1) Rn	GET W(R1.W1,Rn.W2): ∃R2,…,∃R(n−1)(R1.A1=R2.A2 ∧ … … ∧ R(n−1).A(2n−1)=Rn.A(2n))	$UIL^{W1\cdot W2}$
7	A2 / A1 W1 / R1(VTF) R2(VTF)	GET W(R2.W1): ∀R1(R1.A1=R2.A2)	UIL^{W1}
8	W1 / R1 / A(2n−1)=A(2n) A1=A2 / Wn … W2 / Rn … A4=A3 R2	GET W(R1.W1,R2.W2,…,Rn.Wn): (R1.A1=R2.A2 ∧ … … ∧ Rn.A(2n−1)=R1.A(2n))	$UIL^{W1\cdots Wn}$

(A1, A2……An,
W1, W2……Wn): カラム名

型の基本演算への展開を**表 3** に示す．実際の問合せは，これら 8 種類の基本型により，または基本型の組合せにより表現される．

表 2 に示されるように各基本型の演算結果は基本型 1 の結果と等価になる．したがって，問合せが複数の基本型の組合せによって表現される場合には，個々の基本型に対応する演算列を遂行しおのおのの結果が基本型 1 の形になるのでそれらの結果を組み合わせて基本型 2〜8 の形にした後，再び対応する演算列を遂行することにより基本型 1 の形に変換するという操作を繰り返す．

実際の問合せは上述したように本演算系の演算列に展開することができる．各基本型の展開で明らかなように，問合せの中に結合演算 (join) が含まれている場合においても，本演算系は中間結果としての関係を再構成する必要がない．さらに，各基本型の展開で，[1] 関係内のアイテム値の参照による補助データ (転置インデックス，結合テーブル) の生成，[2] 補助データ間の演算 (Multiply-VTFG, Multiply-AIR, Divide), という 2 種の演算により問合せが遂行されていることがわかる．このことは，データベースにあらかじめ補助データが用意されている場合には，[1] の演算は必要なく，[2] の演算すなわち補助データだけで問合せを遂行できるという本演算系の特徴を明らかにしている．

3.2 問合せの展開列

ALPHA で表現された問合せ例を ALDD により表記し，それを本演算系の演算列に展開する例を示す．例として用いる関係および問合せ ALDD の表現を**図 1** に示す．

(問合せ)

GET W (R.CITY, S.PART):
{R.S# = S.S#}

この問合せは，基本型 5 により処理される．これは，関係代数では結合演算と射影演算によって処理される問合せである。

① Generate-UIL·AIR (VTF^R-1 $= \{1, 2, 3, 4\}^R$, ${}_R\boldsymbol{C}_{S\#} = {}_S\boldsymbol{C}_{S\#}$, VTF^S-2 $= \{1, 2, 3\}^S$)
→ $UIL^{S\#}$-1·$AIR^{R\cdot S}$-1
$= ((S_1)(S_2))\cdot(\{1, 3\}^R - \{1\}^S, \{2\}^R - \{2, 3\}^S)$)

$UIL^{S\#}$-1 は，VTF^R-1 によって示される関係 R 内の有効な組 (TID 1, 2, 3, 4) の $S\#(S_1), (S_2), (S_3)$ と，VTF^S-2 によって示される関係 S 内の有効な

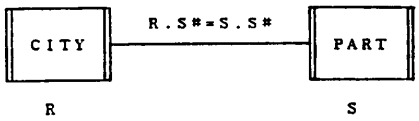
関係 R

TID	カラム (S#)	カラム (CITY)
1	S 1	London
2	S 2	Paris
3	S 1	Paris
4	S 3	London

関係 S

TID	カラム (S#)	カラム (PART)
1	S 1	Nut
2	S 2	Bolt
3	S 2	Nut

(a) 関係の例
An Example of relations.

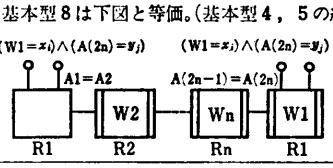


(b) 問合せ例
An example of a query.

図 1　関係の例と問合せ例
Fig. 1　An Example of relations and a query.

表 3 基本型の基本演算への展開

Table 3 Decomposition of basic types to basic operations.

基本型	基本演算への展開
1	G.-UIL(VTF^{R1}, A1)→UIL^{A1}
2	G.-UIL·VTFG(VTF^{R1}, A1, =)→UIL^{A1}·VTFG^{R1}-1 G.-UIL·VTFG(VTF^{R1}, A2, =)→UIL^{A2}·VTFG^{R1}-2 Multiply-VTFG((UIL^{A1}, VTFG^{R1}-1), (UIL^{A2}, VTFG^{R1}-2))→(UIL^{A1,A2}, VTFG-3) ⋮ G.-UIL·VTFG(VTF^{R1}, An, =)→UIL^{An}·VTFG^{R1}-(2n−1) Multiply-VTFG((UIL$^{A1,A2\cdots An-1}$, VTFG^{R1}-(2n−2)), (UIL^{An}, VTFG^{R1}-(2n−1))) →(UIL$^{A1,A2\cdots An}$, VTFG^{R1}-2n)
3	Selection(VTF^{R1}-1, A1, =, 'd1')→VTF^{R1}-2 Selection(VTF^{R1}-2, A2, =, 'd2')→VTF^{R1}-3 ⋮ Selection(VTF^{R1}-n, An, =, 'dn')→VTF^{R1}-(n+1) G.-UIL(VTF^{R1}-(n+1), W1)→UIL^{W1}
4	G.-UIL(VTF^{R1}, A1)→UIL^{A1} G.-VTF(VTF^{R1}-1, UIL^{A1}, A2)→VTF^{R2}-2 G.-UIL(VTF^{R2}-2, W1)→UIL^{W1}
5	G.-UIL·AIR(VTF^{R1}, (R1.A1=R2.A2), VTF^{R2})→UIL^{A1}·AIR$^{R1\cdot R2}$ G.-UIL·VTFG(VTF^{R1}, W1, =)→UIL^{W1}-1·VTFG^{R1}-1 Multiply-AIR(UIL^{W1}-1·VTFG^{R1}-1, AIR$^{R1\cdot R2}$)→UIL^{W1}-1·VTFG^{R2}-2 G.-UIL·VTFG(VTF^{R2}, W2, =)→UIL^{W2}-2·VTFG^{R2}-3 Multiply-VTFG((UIL^{W1}-1, VTFG^{R2}-2), (UIL^{W2}-2, VTFG^{R2}-3)) →(UIL$^{W1\cdot W2}$, VTFG^{R2}-4)
6	G.-UIL·VTFG(VTF^{R1}, W1, =)→UIL^{W1}·VTFG^{R1}-1 G.-UIL·AIR(VTF^{R1}, (R1.A1=R2.A2), VTF^{R2})→UIL^{A1}·AIR$^{R1\cdot R2}$-1 Multiply-AIR(UIL^{W1}·VTFG^{R1}-1, AIR$^{R1\cdot R2}$-1)→UIL^{W1}·VTFG^{R2}-2 ⋮ G.-UIL·AIR(VTF$^{R(n-1)}$, (R(n−1).A(2n−1)=Rn.A(2n)), VTF^{Rn}) →UIL$^{A(2n-1)}$·AIR$^{R(n-1)\cdot Rn}$ Multiply-AIR(UIL^{W1}·VTFG$^{R(n-1)}$, AIR$^{R(n-1)\cdot Rn}$)→UIL^{W1}·VTFG^{Rn}-1 G.-UIL. VTFG(VTF^{Rn}, W2, =)→UIL^{W2}·VTFG^{Rn}-2 Multiply-VTFG((UIL^{W1}, VTFG^{Rn}-1), (UIL^{W2}·VTFG^{Rn}-2)) →UIL$^{W1\cdot W2}$·VTFG^{Rn}-3
7	G.-UIL(VTF^{R1}, A1)→UIL^{A1} G.-VTF(VTF^{R2}, UIL^{A1}, A2)→VTF^{R2} G.-UIL·VTFG(VTF^{R2}, A2, =)→UIL^{A2}·VTFG^{R2}-1 G.-UIL·VTFG(VTF^{R2}, W1, =)→UIL^{W1}·VTFG^{R2}-2 Divide(VTFG^{R2}-1, VTFG^{R2}-2)→VTFG^{R2}-3·(UIL^{W1})
8	基本型8は下図と等価。(基本型4，5の組合わせにより展開)  i =1, …m_1(m_1：カラムW1のアイテム値の種類数) j =1, …m_2(m_2：カラムA(2n)のアイテム値の種類数) x_i：カラムW1内のアイテム値 y_j：カラムA(2n)内のアイテム値

Generate=G.

組 (TID 1, 2, 3) の $S\#$(S1), (S2) について，両者に共通に含まれている $S\#$ すなわち (S1), (S2) の並びである．UIL$^{S\#}$-1 はこのように演算対象の両関係の joining カラムに共通に含まれるアイテム値の並びである．AIR$^{S\cdot R}$-1 は関係 R の組 1, 3 が関係 S の組1と (S1) により結合し，関係 R の組2が関係 S の組2, 3と (S2) により結合することを示す．

② Generate-UIL·VTFG (VTF^{R}-1={1, 2, 3, 4}^{R}, $_{R}\boldsymbol{E}_1$=($_{R}\boldsymbol{C}_{\text{CITY}}$), =)
→UIL$^{\text{CITY}}$-2·VTFG^{R}-1
=((London), (Paris))·({1, 4}^{R}, {2, 3}^{R})

UIL$^{\text{CITY}}$-2 と VTFG^{R}-1 の対は，関係 R のカラム CITY による転置インデックスに相当する．

③ Generate-UIL·VTFG (VTF^{S}-2={1, 2, 3}^{S}, $_{S}\boldsymbol{E}_2$=($_{S}\boldsymbol{C}_{\text{PART}}$), =)
→UIL$^{\text{PART}}$-3·VTFG^{S}-2=((Nut), (Bolt))·({1, 3}^{S}, {2}^{S})

UIL$^{\text{PART}}$-3 と VTFG^{S}-2 の対は，関係 S のカラム PART による転置インデックスに相当する．

④ Multiply-AIR ((UIL$^{\text{CITY}}$-2·VTFG^{R}-1), AIR$^{R\cdot S}$-1)
→UIL$^{\text{CITY}}$-2·VTFG^{S}-3
((London), (Paris))·({1}^{S}·{1, 2, 3}^{S})

UIL$^{\text{CITY}}$-2 と VTFG^{S}-3 の対は，関係 R と S が

$S\#$ によって結合されるときに，関係の R カラム ${}_R\boldsymbol{C}_{\text{CITY}}$ のアイテム値 "London", "Paris" が関係 S のどの組と結合するかを示す．$\text{UIL}^{\text{CITY}}\text{-}2\cdot\text{VTFG}^S\text{-}3$ は，"London" は $S\#$="S1" により関係 S の組1と，また "Paris" は $S\#$="S1""S2" により関係 S の組 1, 2, 3 と結合することを示している．

⑤ Multiply-VTFG $((\text{UIL}^{\text{CITY}}\text{-}2\cdot\text{VTFG}^S\text{-}3), (\text{UIL}^{\text{PART}}\text{-}3\cdot\text{VTFG}^S\text{-}2))$
$\rightarrow \text{UIL}^{\text{CITY·PART}}\text{-}4\cdot\text{VTFG}^S\text{-}4$
$=$((London, Nut), (London, Bolt), (Paris, Nut), (Paris, Bolt))·
$(\{1\}^S, \{\phi\}^S, \{1,3\}^S, \{2\}^S)$

$\text{UIL}^{\text{CITY·PART}}\text{-}4$ と $\text{VTFG}^S\text{-}4$ の対は，関係 R のカラム ${}_R\boldsymbol{C}_{\text{CITY}}$ と関係 S のカラム ${}_S\boldsymbol{C}_{\text{PART}}$ のアイテム値間の結合関係を示している．(London, Nut) は関係 S の組1により，(Paris, Nut) は関係 S の組 1,3 により，また (Paris, Bolt) は関係 S の2組によりおのおの結合され，それら以外の組合せは存在しない ($\{\phi\}$) ことを示す．したがって (London, Nut), (Paris, Nut), (Paris, Bolt) がこの問合せの結果である．

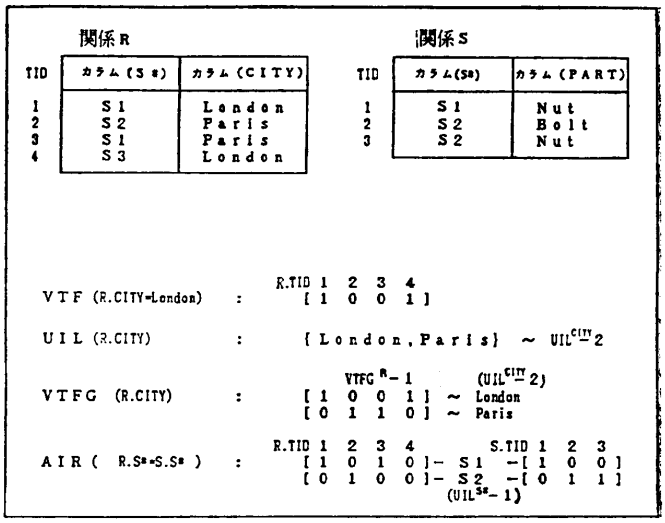


図 2　基本データ構造
Fig. 2　Basic data structures.

4. 演算系の実現方式

本章では2章で定義した演算系の具体的な実現方式の一構成について述べる．ここで述べる実現方式は本演算系の一実現例であり，本演算系は他の方式によっても実現可能である。

以下に示す実現方式は，本演算系の特徴である補助データに対する演算を並列処理の環境で遂行するものである．

4.1 データ構造とデータ配置

2.1 節で定義した α, β 集合のデータ構造（物理表現例）を図2に示す．以下に各集合のデータ構造を定義する．

${}_i\boldsymbol{C}_k$：カラム ${}_i\boldsymbol{C}_k$ はアイテム値 $d_{ijk}(j=1\sim n_i)$ を2.1節(3)で定義した TID の順序に配列することにより表現される．したがって，関係 R_i における組としてのつながりは，各カラム内のアイテム値 d_{ijk} の配列順序によって保たれる．

VTF：VTF^{R_i} は関係の R_i の TID の集合をビット列で示すことにより表現される．関係 R_i 内の TID の値 j の有無はビット列の j ビット目の値により判明する．

UIL：UIL^D は指定された任意の l 個のカラムについて重複を排除した後の複合アイテム値 ($l=1$ のときは1アイテム値) I_{ij} の集合 ${}_iE$ の各要素 I_{ij} を適当な順序（昇順，降順等）に並べることにより表現される．

VTFG：VTFG^{R_i} は，ビット列で表現された VTF をその VTFG に対応する UIL^D 内の I_{ij} の並び順にしたがって並べることによって表現される．

AIR：$\text{AIR}^{R_i\cdot R_i'}$ は二つの VTFG によって表現される．両 VTFG 間の VTF の対応は各 VTFG 内の VTF の並び順によって保たれる．

本実現方式は，上述のデータ構造に対し並列処理の環境で2.2節において定義した各基本演算を遂行するものである．

4.2 基本演算の処理方法

2.2 節で定義した基本演算は次の2種に分けることができる．

[1] 関係 R_i を構成するカラム内のアイテム値の参照により VTF, UIL, VTFG, AIR を生成する演算 (2.2 節(1)～(6))．

[2] VTFG, AIR 間，すなわち補助データ間の演算 (2.2 節(7)～(9))．

[1] の演算は，データベースにあらかじめ補助データが用意されている場合には必要ない．補助データがない場合には，関係 R_i 内の演算対象カラム内の各アイテム値を参照し，比較操作を行うことによって

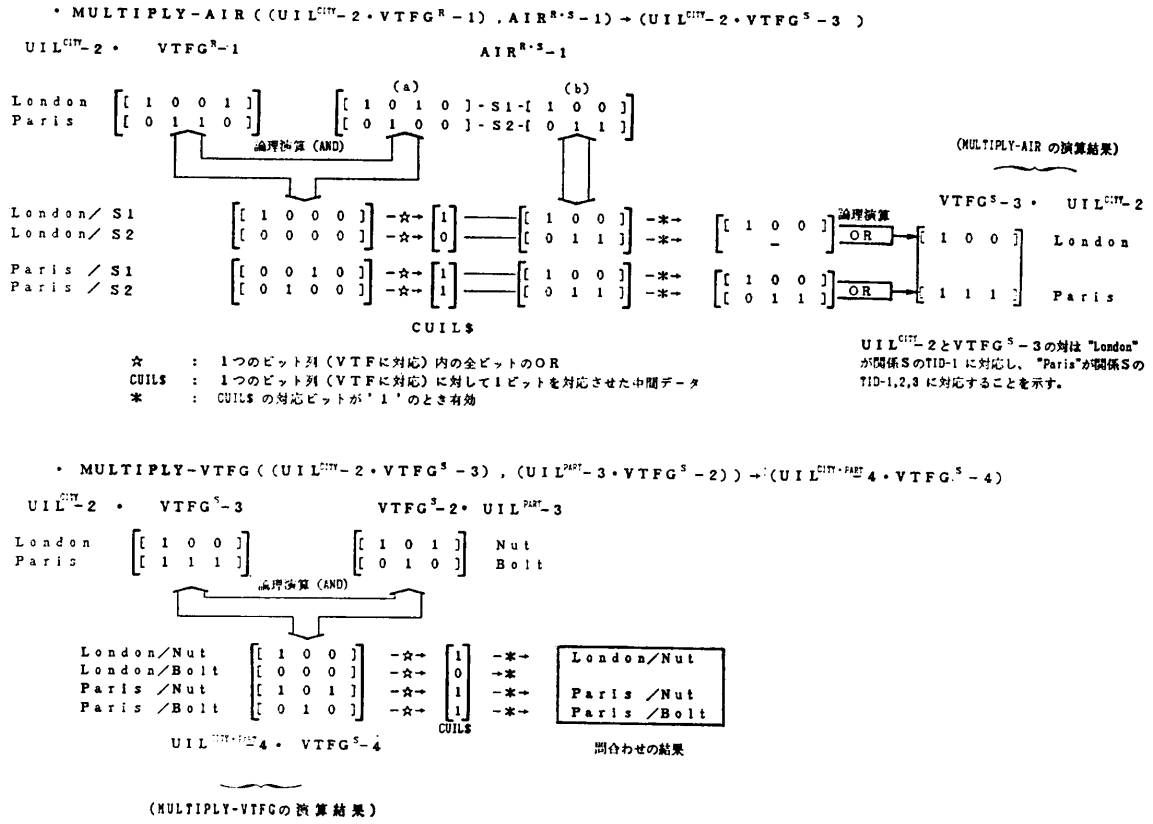


図 3 問合せ処理方式
Fig. 3 Processing scheme of a query.

VTF, UIL, VTFG または AIR を生成する.

[2] の演算は, 二つの VTF 間すなわち二つの TID 集合間の集合演算 (∪, ∩) を基本としている. 本実現方式では VTF がビット列で表現されているので, この集合演算はビット列間の単純な論理演算 (logical operation) AND または OR によって処理できる. 3.2 節で用いた問合せ例をビット列間の論理演算によって処理する過程を**図 3** に示す. 3.2 節で述べた問合せ処理において①～③の演算は Generate-UIL・AIR または Generate-UIL・VTFG であり, それらは [1] の演算に属している. ここでは, [1] の演算によって生成される UIL, VTFG, AIR はすでに存在しているものとし 3.2 節で述べた問合せ処理における④, ⑤の演算すなわち [2] に属する Multiply-AIR, Multiply-VTFG の処理過程を示した.

本実現方式では, これら [1], [2] の演算を並列処理により実現する. 並列プロセッサを構成する PE (プロセッシング・エレメント) 群へのデータ配置の例として VTFG の各 PE への分割配置の様子を**図 4** (A)に示す. また本演算系の典型的な演算である Multiply-AIR を並列プロセッサで処理する場合の, 1PE 内での処理の様子を図 4 (B)に示す. 各 PE に分割配置された VTFG と AIR のビット列間の論理演算 (AND) は並列に行われる. 図 3 における CUIL$ を生成するために, 各 PEj(j=1～m) で分割して行われた VTF 間の論理演算の結果に対して, 図 4 (B)に示すように PE 間での論理演算 (OR) が行われる. このように, 本実現方式ではビット列 (VTF) 間の論理演算を並列処理することにより, Multiply-AIR, Multiply-VTFG, Divide を遂行する.

5. 性能評価

2 章で定義した本演算系の有効性を評価するために, 4 章で述べた実現方式をモデル A とし, また関係に対し直接に関係代数演算を施し, 各演算の結果として新しい関係を毎回再構成する一般的な方式モデル B を設定し, 問合せの処理時間について両モデルの比較

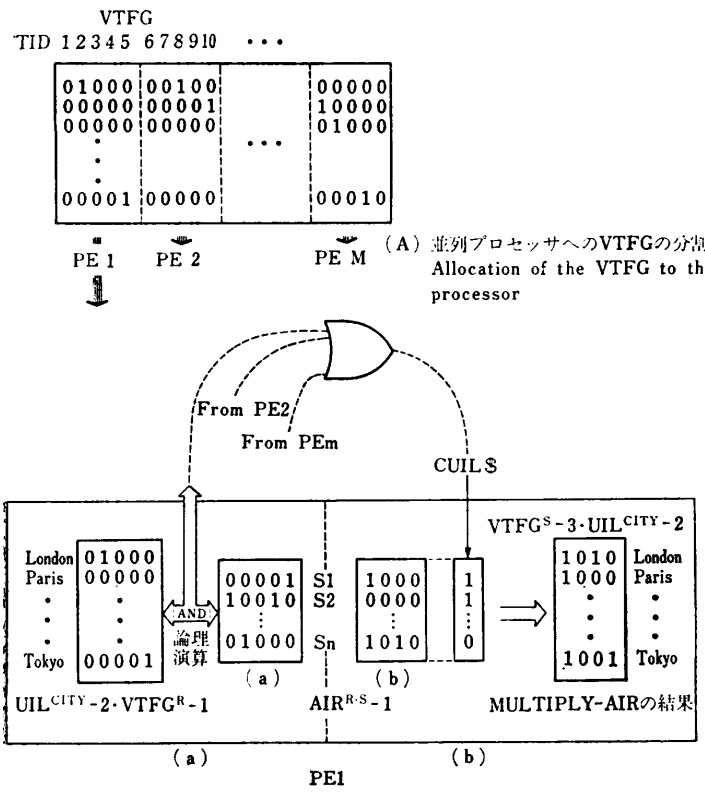


図 4 並列プロセッサへの VTFG の分割配置と MULTIPLY-AIR の PE 内処理
Fig. 4 Allocation of the VTFG to the parallel processor and MULTIPLY-AIR execution in a single PE.

を行った.

5.1 評価モデル

モデルA，モデルBの環境を以下のように設定した.

(1) SIMD (Single Instruction Multiple Data stream) 方式の並列処理を行う. モデルAでは，演算対象データは並列プロセッサを構成する PE へ図4(A)に示したように配置される. モデルBでは，演算対象の関係内の組は各 PE へ**図5**に示すように均等に分配される. 以下，分割配置された関係の各 PE が担当する部分を部分関係 (partial relation) とよぶ.

(2) 問合せで用いるすべての演算対象データは並列プロセッサ内の等速読み出し可能な主記憶に格納される.

(3) モデルBでは，関係に対して直接に関係代数演算を施し，各関係代数の演算結果は新しい関係を再構成することにより表す. モデルBでは，結合演算，射影演算，割り算等の関係代数演算の処理にソート (sorting) およびマージ (merging) のアルゴリズムを応用する. モデルBの並列処理方式を結合演算を例にとり以下に示す.

(ステップ1) 各 PE に分割配置された結合演算の演算対象の関係 (R_i, R_i') の部分関係を，各 PE は独立に joining カラムについてソートする.

(ステップ2) PE-n ($n=1\sim M$, M: は PE の台数) によりソートされた関係 R_i の部分関係の全組が全 PE へブロードキャストされる.

(ステップ3) 各PEは PE-n からブロードキャストされた関係 R_i 内の組と自 PE 内でソートした関係 $R_{i'}$ の部分関係をマージし，結合演算の条件 (=, ≠, ≧, ≦, >, <) を満たす組を結合することにより結合演算結果としての新しい関係を生成する.

(4) モデルAは，2章で定義した演算系を4章で示した実現方式により実現するものである. ここでは，データベースに補助データ (転置インデックス，結合テーブル) は用意されておらず，問合せ処理時にそれらは動的に生成されるものとする. 転置インデッ

T 1	T11	—	—	—	$Tm1$
T 2	T12	—	—	—	$Tm2$
T 3	T13	—	—	—	$Tm3$
T 4	T14	—	—	—	$Tm4$
T 5	T15	—	—	—	$Tm5$
T 6	T16	—	—	—	$Tm6$
T 7	T17	—	—	—	$Tm7$
T 8	T18	—	—	—	$Tm8$
T 9	T19	—	—	—	$Tm9$
T10	T20	—	—	—	$T(m+1)0$
PE-1	PE-2	—	—	—	PE-M

Tj: 関係 Ri 内のタプル
j: TID
M: PE の台数
$m=M-1$
PE: プロセッシング・エレメント

図 5 演算対象の関係 (R_i) の並列プロセッサへの分割
Fig. 5 Allocation of an operand relation (R_i) to the parallel-processor.

演算対象の関係	関係R (S♯, CITY) 関係S (S♯, PART)
問合せの ALPHA 表現	GET W (R. CITY, S. PART): {R. S♯=S. S♯)
問合せの関係代数表現	Join (R. S♯=S. S♯) ⟶RS (CITY, S♯, PART) Projection (RS. CITY, RS. PART) ⟶RR (CITY, PART)

図 6 問 合 せ 例
Fig. 6 An example of a query.

クスに対応する UIL・VTFG の生成および結合テーブルに対応する UIL・AIR の生成はモデルBと同様のソートおよびマージのアルゴリズムを適用するものとする.

モデルAについては，2章で示した各基本演算の処理時間，またモデルBについては各関係代数演算の処理時間を求める評価関数（付録）を以下の事項を基本に設定した.

1) 各 PE 内の処理時間は，PE 内のメモリへの参照回数に比例する.

2) PE 間の交信処理時間は PE 間の交信操作の回数および1回の交信におけるデータ転送量に比例する.

5.2 性 能 評 価

図6に示した結合演算と射影演算から構成される問合せ例のモデルAとモデルBによる処理時間を比較検討する. この問合せは，モデルAとモデルBの処理方式の特徴を明確にする例である. モデルA，モデルBによる問合せの処理時間は，付録に示した評価関数を用いて求めた. モデルBによるこの問合せの処理において，関係Rと関係Sに対し結合演算を行った結果の関係を RS とし，関係 RS に対して射影演算を行った結果の関係を RR とする. 関係 R, S, RS, RR の組数をおのおの NR 1, NR 2, NR 3, NR 4 とする. ここで NR 1=NR 2=NO とし，関係 R, S の組の長さは 64 バイトとした. 結合演算の結果である関係 RS の組数 NR 3 は NO, 10×NO, 100×NO の三つの場合について評価した. また，結合演算後の射影演算の結果である関係 RR の組数は NR 4=NO とした. モデルAではこの問合せは表2に示した基本型5に対応する例であり表3に示した本演算系の基本演算列により処理される.

NO=16000, 64000 の場合について結合演算の結果の関係 RS の組数 NR 3 の変化に対するモデルBの処理時間の推移，およびモデルBにおける NR 3 の変化に対する状況でのモデルAの処理時間の推移を**表4** (NO=16000 の場合)，**表5** (NO=64000 の場合に示す.

表 4 評価結果 (NO=16000)
Table 4 Evaluation result (NO=16000).

NR3 (結合演算の結果RSの組数) NO=16000	処理時間 (単位：秒)	
	モデルA	モデルB
NO	0.49	0.54
10*NO	0.35	1.69
100*NO	0.33	8.04

表 5 評価結果 (NO=64000)
Table 5 Evaluation result (NO=64000).

NR3 (結合演算の結果RSの組数) NO=64000	処理時間 (単位：秒)	
	モデルA	モデルB
NO	9.41	3.85
10*NO	7.76	6.97
100*NO	7.57	34.00

モデルBによる処理時間は NR 3 の増大にともなって増加している. モデルBにおいては，結合演算によって生成される関係 RS の組数 NR 3 が 10×NO, 100×NO と増大するにしたがって，関係 RS に対して行われる射影演算は組数 10×NO, 100×NO に対する重複の排除という大きな処理となってしまう. 一方，モデルAでは結合演算の結果の関係 RS は生成されず，結合演算，射影演算に対応する処理を Multiply-AIR, Multiply-VTFG によって遂行するので処理対象データ量は増大しない. 逆に，結合演算 (S. S♯=S. S♯) の結果を表現している $AIR^{R\cdot S}$ の大きさは，モデルBで NR 3 が大きくなる状況において，小さくなる傾向にある. なぜならば，一般に関係 R と S の joining カラム内のアイテム値の種別数 $S♯=\{S1, S2, \cdots, S_m\}$ における m の値) が小さくなるにつれて NR 3 は大きくなるのに対し，$AIR^{S\cdot R}$ を構成する VTF の総数は小さくなる傾向にあるからである. この $AIR^{S\cdot R}$ を処理対象とする Multiply-AIR の処理時間は，$AIR^{S\cdot R}$ のデータ量の減少に伴い短くなる. モデルAの処理時間が，モデルBの処理時間が増大する状況で逆に減少するのは，Multiply-AIR の処理時間が減少していくからである.

以上のように，モデルBにおいて結合演算の結果が大きくなり，それに続く演算が大きな中間結果の関係に対して遂行される必要がある問合せの場合，モデルAは有効であり，処理効率を大きく改善できる.

また NO=16000 (表4) と NO=64000 (表5) の

比較においてモデルBの処理時間は約 1：4 であるのに対しモデル A では約 1：20 になっている．すなわち，モデル B の処理時間が NO にほぼ比例するのに対し，モデルAの処理時間は NO^2〜NO^3 に比例している．したがって，モデルAは，莫大な組数をもつ関係に対しては処理効率を大きく改善しない．

6. むすび

本論文では，関係データベース・システムを実現する場合に基本的な性能を左右する関係演算の演算方式として，転置インデックス，結合テーブル等，従来は補助的に用いられてきたデータに，関係演算遂行の中心的役割を与えた新しい演算系を提案した．本演算系は，関係データベース処理において用いられていた既存の補助データを演算系内に統一的に組み込むことによって，補助データをさらに有効に応用するものである．したがって，本演算系で定義されたデータおよび演算は既存の関係データベース・システムにおいても容易に適用できるものである．

本論文で提案した演算系はおもに複雑な関係演算（結合演算，射影演算，割り算）の処理効率の改善を目指したものである．著者らは，単純な処理である選択演算，制約演算およびデータベースの更新，挿入，削除については従来の方式と同様である関係内アイテム値を全数探索するアルゴリズムを前提としている．著者らは，これらの単純な演算およびデータベースの更新，挿入，削除の処理の機構は logic-per-track 方式のようにデータベースの格納媒体であるディスク等の2次記憶装置に直接組み込まれるべきであると考えている．

現在，著者らは関係データベースマシン SPIRIT[11)〜13)]の検討を進めており，本演算系は関係演算処理部の演算方式として採用されている．著者らは今後，関係演算に対するさまざまな演算方式に対する評価，および二次記憶，主記憶間のデータ供給方式に対する評価を行い関係データベース・マシンの評価システムの設計を進めたいと考えている．

謝辞　日ごろご指導いただいている，慶応義塾大学理工学部専任講師 所 真理雄博士に感謝いたします．また，本研究に当たり，有益な討論とご協力をいただいた本学卒業生加藤 洋（富士ゼロックス），小沢裕之（三菱電気）および本学大学院生の峰松彩子，磯田道男，小島清信の諸氏に深謝いたします．

参考文献

1) Codd, E. F.: A Relational Model of Data for Large Shared Data Banks, *Comm. ACM*, Vol. 13, No. 6, pp. 377-397 (June 1970).
2) Codd, E. F.: Relational Completeness of Database Sublanguages, Courant Computer Science Symposia 6, "Database Systems", pp. 65-98 (May 1972).
3) Codd, E. F.: A Data Base Sublanguage Founded on the Relational Calculus, Proc. 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control, pp. 35-68 (Nov. 1971).
4) Date, C. J.: *An Introduction to Data Base Systems*, Addison-Wesley, Reading, Mass. (1975).
5) Martin, J.: *Computer Data-Base Organization*, Prentice-Hall, Reading (1975).
6) Schuster, S. A., Ozkarahan, E. A. and Smith, K. C.: RAP-An Associative Processor for Database Management, Proc. 1975 NCC, Vol. 45, AFIPS Pres, Montvale, N. J., pp. 379-387 (1975).
7) Schuster, S. A. et al.: RAP. 2-An Associative Processor for Databases and Its Applications, *IEEE Trans. Comput.*, Vol. C-28, pp. 446-457 (June 1979).
8) Dewitt, D. J.: DIRECT-A Multiprocessor Organization for Supporting Relational Database Management Systems, *IEEE Trans. Comput.*, Vol. C-28, pp. 395-406 (June 1979).
9) 清木 康，田中浩一，上林憲行，相磯秀夫：動的マークビットによるリレーショナル演算の並列処理方式とその評価，電子通信学会電子計算機研究会資料 EC 79-66, pp. 61-70 (1980).
10) 加藤 洋，瀬尾和男，上林憲行，相磯秀夫：リレーショナルデータベースマシンにおけるデータステージング機構の考察，電子通信学会電子計算機研究会資料 EC 79-67, pp. 71-80 (1980).
11) 上林憲行，小沢裕之，清木 康，加藤 洋，瀬尾和男，田中浩一，相磯秀夫：リレーショナルデータベース演算のハードウェア化アルゴリズムとそれに基づくデータベースマシンアーキテクチャ，情報処理学会計算機アーキテクチャ研究会資料 35-4, pp. 1-12 (1979).
12) Kamibayashi, N., Kato, H., Kiyoki, Y., Ozawa, H., Seo, K. and Aiso, H.: SPIRIT: A New Relational Database Computer Employing Functional-Distributed Multi-Microprocessor Configulation, 1st International Conference on Distributed Computing Systems, pp. 757-771 (Oct. 1979).
13) Kiyoki, Y., Tanaka, K., Kamibayashi, N. and Aiso, H.: Design and Evaluation of Relational Database Machine Employing Advanced Data Structures and Algorithms, 8 th International Symposium on Computer Architecture, pp. 407-423 (May 1981).

付　録

(1) モデルA

Selection=TR*N1/M+TW*L1*d1*N1/M

G.-UIL・VTFG=TSORT*(LC*N1/M)
×log(N1/M)+TBC*LV1*K1
+TR*(LC*K1+LC*N1)
+TFW*(K1+N1/M)

G.-UIL・AIR=TSORT*((N1/M)*log(N1/M)
+(N2/M)*log(N2/M))
+TBC* min(K1, K2)
+TR*(min(K1, K2)+N1+N2)
+TFW*2* min(K1, K2)
+TFW*(N1+N2)/M

Multiply-AIR=TFR*(KV1+KKV1*KA1)
+TFBC*KA1*KV1
+TBR*(N1/M)*KKA1*KKV1
+TFR*KA1*KV1
+TFW*KV1
+TBW*(N2/M)*KKAL*KV1*q

Multiply-VFTG=TFR*(KV1+KKV1*KV2)
+TFBC*KV1*KV2
+TBR*(N1/M)*KKV1
×KKV2+TFR*KV1*KV2
+TFW*KV3
+TBW*KKV3*(N1/M)

(2) モデルB

B-Selection=TR*N1/M+TW*L1*d1*N1/M

B-Projection=TSORT*(LC*N1/M)*log(N1/M)
+TBC*L1*K1+TR*LC*(K1
+KK1*M)+TW*L1*K1

B-Implicit Join=TSORT*((N1/M)* log (N1/M)
+(N2/M)*log((N2/M))
+TBC*L2*N2+TR*(N1
+N2)+TW*L1*d1*N1/M

B-Join=TSORT*((N1/M)*log(N1/M)
+(N2/M)*log(N2/M))
+TBC*min(L1, L2*N2)
+TR*(N1+N2)+TW*(L1
+L2)*N1*N2*ALPHA/M

(パラメータ)

TR: 1アイテム値の参照・比較時間 (1 μsec)

TW: 1アイテム値の参照・書き込み時間 (2 μsec)

TSORT: ソート処理に要する時間 (ソート処理は TSORT×$n\log_2 n$ で表される. n: アイテム値数) (3 μsec)

TFR: CUIL$ 内の1ビットの参照時間 (0.01 μsec)

TFW: CUIL$, VTFG 内への1ビットの書き込み時間 (0.01 μsec)

TBR: 二つの VTF 間の各1ビットの論理演算処理に要する時間 (0.01 μsec)

TBW: VTF の1ビットの書き込み時間 (0.01 μsec)

TBC: PE 内の1アイテム値の全 PE への転送時間 (2 μsec)

TFBC: CUIL$ を生成するための全 PE 間での論理演算処理において1ビットについて要する時間 (0.02 μsec)

M: PE の台数 (64台)

N1, N2: 演算対象の関係 R1, R2 の組数

d1*N1: Selection 演算により有効となった組数 (d1≤1)

K1, K2: 演算対象カラム (joining カラムまたは Projection の演算対象カラム) 内のアイテム値または複合アイテム値の種別数

KK1: 1PE 内の部分関係内の演算対象カラム内のアイテム値または複合アイテム値の種別数

L1, L2: 関係 R1, R2 内のカラム数

LC: ソート, マージ処理において比較の対象となるカラム数

KV1, KV2, KV3: AIR または VTFG 内の VTF の数

KKV1, KKV2, KKV3: 1PE 内の AIR または VTFG 内の VTF 数

KA1: 関係 R1, R2 の joining カラムに共通に含まれるアイテム値の種別数

KKA1: 関係 R1, R2 の joining カラムに共通に含まれるアイテム値の 1PE 内の種別数

LV1: Generate-UIL・VTFG 処理において演算対象となるカラム数

q: Multiply II 処理時に VTFG 内で OR 演算の対象となる VTF の割合

ALPHA: 結合演算 (join) における結合率
(結合演算により生成される関係 R3 の組数 N3 は N3=N1×N2×ALPHA となる)

(昭和56年10月15日受付)
(昭和57年5月19日採録)