

非正規関係の問合せ変換について

鶴岡 邦敏

日本電気(株) C & C システム研究所

非正規関係を既存の関係データベース上に構築するという観点から、非正規問合せの表現と実現方式とを提案する。初めに、組関係論理を拡張し、非正規関係に対する問合せ表現を定義する。次に、非正規関係に対する問合せを正規関係に対する問合せに変換するアルゴリズムを提案する。また、その変換アルゴリズムによって得られる問合せ間の等価性を示す。本報告で記述する問合せ変換方式は、フォーム管理や情報検索等で重要となる非正規関係データモデルを、既存の関係データベース上に実現するための手がかりを与えるものである。

ON QUERY TRANSFORMATION FOR NON-FIRST-NORMAL-FORM RELATIONAL DATABASES

Kunitoshi TSURUOKA

C&C Systems Research Laboratories, NEC Corporation
4-1-1 Miyazaki, Miyamae-ku, Kawasaki, 213 Japan

Query representation and implementation strategy are proposed for non-first-normal-form (NF^2) relational databases, from the viewpoint that they are implemented on existing relational databases. At first, tuple relational calculus is extended to define query representation for NF^2 relations. Next, a query transformation algorithm is proposed which transforms a query for a NF^2 relation into one for a first-normal-form (1NF) relation. Additionally, the equivalence of transformed queries is shown. The query transformation strategy, proposed, can be a clue for the safe implementation of NF^2 relational data model, which is important for such applications as office forms management or information retrieval, on existing relational databases.

1. はじめに

近年注目を集めている非正規関係 (NF²) モデルに関しては、NEST/UNNEST等の非正規化/正規化操作とその性質[Jae82, Kit82, Sch82, Fis83]、NF²関係代数[Fis83, Sch85]、NF²従属性[Mak77, Jae82, Ari83]等の理論的研究が進んでいる。一方応用面からは、オフィス情報システムにおけるフォーム管理[Luo81, Tsi82, Lum82, Yao84, Tsu85, Tsu87, Wat87]、情報検索[Sch82]等を初めとしてその重要性が認識されつつある。オフィス業務処理の立場から見ると、非正規形入力帳票の処理[Tsu85]は必須機能であり、またデータベースからの非正規形出力帳票(レポート)の生成処理[Tsu87, Wat87]は、見やすい報告書の作成という観点から重要な意味を持つ。

非正規関係モデルを実現する場合に、内部モデルとしても非正規関係を用いる方法[Sch85]もあるが、既存システムとの整合性を考慮すると、正規関係(1NF)を内部モデルとしてその上に非正規関係の概念・外部モデルを構築する方法が当面は現実的であろう。この場合、非正規関係上に表現される問合せを正規関係上の問合せに変換する必要が生ずる。問合せ表現に関しては“nested query”[Sch82, Abi84]や“nested relational algebra”[Sch85]が提案されており、また問合せ変換に関しては[Kiy87]が考察している。しかしながら、(条件だけでなく)コマンド自体がネスト形式をもつ問合せは使い易いとは言えず、また非正規関係の階層構造に立脚したモデル化は不十分と思われる。

本資料では、初めにリスト構造により非正規関係をモデル化し、NEST/UNNEST操作を定義する。次に組関係論理(TRC)を拡張した「拡張組関係論理」(ETRC)により、非正規関係に対する単純な問合せ表現を定義する。さらに、非正規関係に対する問合せ(ETRC)を正規関係に対する問合せ(TRC)に変換するアルゴリズムを示し、その変換アルゴリズムによって得られる問合せ間の等価性を示す。この等価性とは、ETRCをSELECT_N、TRCをSELECT_Uとする時、非正規関係Rに対してUNNEST(SELECT_N(R))=SELECT_U(UNNEST(R))を保証するものである。

本資料で述べる問合せ表現は、(条件はネストするが)問合せ自体はネストしないため、単純で意味的に理解が容易である。また以下で提案する問合せ変換方式は、非正規関係データモデルを既存の関係データベース上に実現するための基礎となりうる。なお、本文中で記述したリスト型データモデルは、非正規関係の構造を適切にとらえることが可能で、非正規関係処理のモデル化のための良好なツールとなりうる。

2. 基本的考え方

以下では、本資料で述べる方式の概要/考え方を例とともに簡単に述べる。非正規関係に対する関係代数とその性質に関しては、既に研究が進んでいる[Jae82, Fis83, Sch85]。“SELECT”に関しては、選択条件がUNNESTされる属性を含む場合には、等式

$$\text{UNNEST}(\text{SELECT}(R)) = \text{SELECT}(\text{UNNEST}(R)) \quad [*]$$

は成立しない[Fis83](NESTされる関係とUNNESTされる関係の両方に同一の選択条件は適用できないため)。本資料で提案するETRCと問合せ変換アルゴリズムは、変換前後の選択条件が等式[*]を満たすものである。図1.に例を示す。

図でSELECT_NはETRC形式の問合せであり、タプル変数間の「子孫-祖先関係」が使用されている(例えば、T₂₂ ≤ T₁₁)。この条件は、2つのタプル間に(前者が後者の構成要素であるという)包含関係があることを意味する。即ち、サブタプル(例: T₂₂)はスーパータプル(例: T₁₁)の構成要素であるか、構成要素の構成要素であるか、・・・である。SELECT_N(ETRC)は、UNNESTされた関係(第1正規形)に対する問合せ表現であるSELECT_Uに変換可能であり、2つのSELECTは等式UNNEST(SELECT_N(従業員))=SELECT_U(UNNEST(従業員))を満足する。

3. 非正規関係モデル

(1) 非正規関係

本節では、非正規関係モデルを与えるために、非正規関係をLISPのリスト構造を基に定義し、またそ

のためにいくつかの関数 (append等) を用いる。まず、(拡張) 関係を定義する。

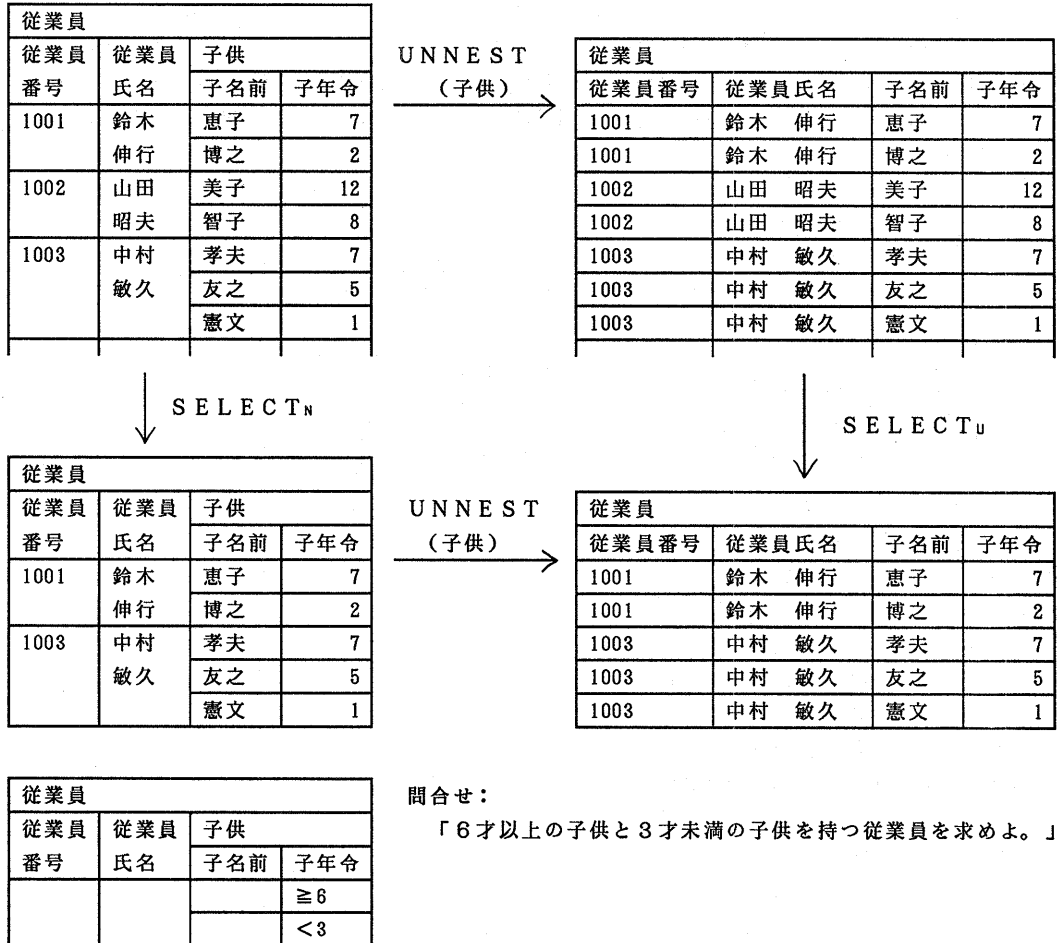
関係: ::= (スキーマ インスタンス)

スキーマ: ::= 属性集合 | (append 属性集合 スキーマ集合)

属性集合: ::= (属性) | (append (属性) 属性集合)

スキーマ集合: ::= (スキーマ) | (append (スキーマ) スキーマ集合)

上記で、appendはリスト結合のためのLISPの標準関数である。インスタンスは以下で定義される。属性



SELECT_N : {T₁₁ | (T₁₁ ∈ 従業員) ∧ (∃ T₂₁ ∈ 子供) ∧
 (∃ T₂₂ ∈ 子供); (T₂₁ < T₁₁) ∧ (T₂₂ < T₁₁) ∧
 (T₂₁ [子年令] ≥ 6) ∧ (T₂₂ [子年令] < 3)}

SELECT_U : {U₁₁ | (U₁₁ ∈ 従業員) ∧ (∃ U₁₂ ∈ 従業員) ∧
 (∃ U₁₃ ∈ 従業員); (U₁₂ [従業員番号] = U₁₁ [従業員番号]) ∧
 (U₁₃ [従業員番号] = U₁₁ [従業員番号]) ∧ (U₁₂ [子年令] ≥ 6) ∧
 (U₁₃ [子年令] < 3)}

図1. 拡張組関係論理と問合せの変換

は関係データベースという属性と同様の意味とする（単純値を持つ）。重要な点としては、スキーマは少なくとも1つの属性を持つことである（後述する主キーに関する制約による）。

次に、値と（拡張）タプルを定義する。

インスタンス ::= タプル集合

タプル集合 ::= (タプル) | (append (タプル) タプル集合)

タプル ::= 値集合 | (append 値集合 インスタンス集合)

値集合 ::= (値) | (append (値) 値集合)

インスタンス集合 ::= (インスタンス) | (append (インスタンス) インスタンス集合)

値は、関係データベースという属性値と同様の意味とする。あるインスタンスは、特定の「下位レベル」のスキーマに属する複数のインスタンスを含みうることに注意が必要である。（実際には、インスタンスはリストではなくセットとすべきであるが、これらの相違はここでは重要でない。） 図2. に上記で定義した対象の例を与える。

関係-N₁

N ₁			
A ₁	A ₂	N ₂	
		A ₃	A ₄
a ₁₁	a ₂₁	a ₃₁	a ₄₁
		a ₃₂	a ₄₂
a ₁₂	a ₂₂	a ₃₃	a ₄₃
		a ₃₄	a ₄₄

タプル-N₁₁

a ₁₁	a ₂₁	a ₃₁	a ₄₁
		a ₃₂	a ₄₂

インスタンス-N₂₁

a ₃₁	a ₄₁
a ₃₂	a ₄₂

タプル-N₂₁

a ₃₁	a ₄₁
-----------------	-----------------

関係-N₁ = (スキーマ-N₁ インスタンス-N₁)

スキーマ-N₁ = (属性-A₁ 属性-A₂ スキーマ-N₂)

スキーマ-N₂ = (属性-A₃ 属性-A₄)

インスタンス-N₁ = (タプル-N₁₁ タプル-N₁₂)

タプル-N₁₁ = (値-A₁₁ 値-A₂₁ インスタンス-N₂₁)

インスタンス-N₂₁ = (タプル-N₂₁ タプル-N₂₂)

タプル-N₂₁ = (値-A₃₁ 値-A₄₁)

タプル-N₂₂ = (値-A₃₂ 値-A₄₂)

タプル-N₁₂ = (値-A₁₂ 値-A₂₂ インスタンス-N₂₂)

インスタンス-N₂₂ = (タプル-N₂₃ タプル-N₂₄)

タプル-N₂₃ = (値-A₃₃ 値-A₄₃)

タプル-N₂₄ = (値-A₃₄ 値-A₄₄)

図2. 非正規関係

次に、いくつかの関数の定義を与える。小文字の名前はキーワード、大文字の名前は変数を意味する。

(owner MEMBER) = OWNER ; リスト (OWNER ... MEMBER ...) がある時、
= null ; // 無い時。

(val TUPLE ASSET) = VALINSSET ; ASSETはあるスキーマのサブリスト、VALINSSETはASSETに対応する
タプルTUPLEのサブリストであり、スキーマはTUPLEがメンバとなっているインスタンスを持つ。

(ancestor D) = (A₁ A₂ ... A_i ...) ; A_i=(owner ... (owner (owner D))...) が成り立つ時 (A_iを「祖先」、Dを「子孫」と呼ぶ)、

=null ; それ以外。
 (compid C) =null ; (owner C) =null の時、
 = (リスト内要素番号) ; (owner (owner C))=null の時、
 = (c₁ c₂ . . . c_k) ; Cの(そのownerの下での)リスト内要素番号がc_kで、かつ
 (compid (owner C))=(c₁ c₂ . . . c_{k-1}) の時。

compidは要素IDを意味し、ある関係の中で、特定の対象を一意に識別する。

図2. の例では、以下が成り立つ。

(owner タブル-N₂₂) = インスタンス-N₂₁、
 (val タブル-N₁₂ (属性-A₁₂ スキーマ-N₂)) = (値-A₁₂ インスタンス-N₂₂)、
 (ancestor タブル-N₂₃) = (インスタンス-N₂₂ タブル-N₁₂ インスタンス-N₁ 関係-N₁)、
 (compid タブル-N₂₃) = (2 2 3 1)。

なお、添字で暗黙の関係を示す場合があるものとする。例えば、スキーマ-R、インスタンス-Rは関係-Rのスキーマとインスタンスである。

タブルはあるインスタンスに対する拡張された行なので、主キーが必要となる。あるスキーマに対する主キーは、2つの関数 (keyattributes スキーマ-R) = 属性集合、(keyvalues タブル) = 値集合、として定義される。値集合は、タブルがメンバであるインスタンス中で、一意でなければならない。

(2) NEST、UNNEST操作

NEST、UNNEST操作は、正規関係と非正規関係との(または非正規関係相互の)変換を与える演算子として知られている[Jae82, Sch82, Kit82, Fis83]。ここでは、関数としてそれらを定義する。

(unnest 関係-N スキーマ-N_a) = 関係-U

ここで、スキーマ-N_aはunnestされるスキーマである(スキーマ-Nの子孫)。スキーマ-N_aはそのメンバによって置換される。インスタンス-Uは、下記の関数unnesttupで定義されるタブルを持つ。

スキーマ-N_i、スキーマ-U_i (i=1,2,...) は各々スキーマ-N、スキーマ-U中のスキーマ(それ自身を含む)であり、スキーマ-N_aのownerをスキーマ-N_bとする。また、TN_{ij}、TU_{ik}をそれぞれスキーマ-N_i、スキーマ-U_iのタブルとし(j,k=1,2,...)、(compid TN_{aj}) = (c₁ c₂ . . . c_n)とする。また、ASSETをスキーマ-U_bのサブリストでスキーマ-N_aに対応するもの、ASSETXをスキーマ-U_bのサブリストで (append ASSETX ASSET) = スキーマ-U_b を満たすものとする。

(図. 3参照。)

この時、関数unnesttupを以下のように定義する。

(unnesttup TN_{ij}) = TU_{ik} ; i≠a かつ i≠b の時、但し (compid TU_{ik}) = (compid TN_{ij}) かつ TN_{ij} と TU_{ik} は同一の要素集合を持つ、

(unnesttup TN_{aj}) = TU_{bk} ; 但し (compid TU_{bk}) = (c₁ c₂ . . . c_{n-3} c_n) で、c_nは TU_{bk} のリスト内要素番号、また

(val TU_{bk} ASSETX) = (val (owner (owner TN_{aj})) ASSETX)
 かつ (val TU_{bk} ASSET) = (val TN_{aj} ASSET)、

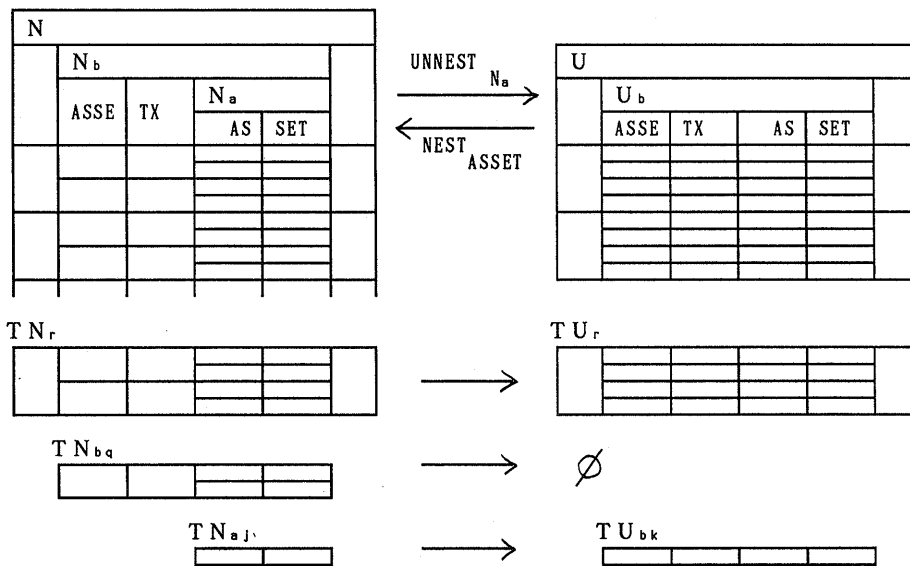
(unnesttup TN_{bj}) = null。

ここでunnesttupは、インスタンス-N_iからインスタンス-U_i (i=1,2,...; i≠a, i≠b) への、及びインスタンス-N_aからインスタンス-U_bへの、全単射である。また、スキーマ-U_bの主キー属性は、スキーマ-N_bの主キー属性とスキーマ-N_aの主キー属性とを結合したものである。

次に、NESTは以下の関数により定義される。

(nest 関係-U ASSET NAME) = 関係-N

ここで、ASSETはスキーマ-U_bのサブリストで、少なくとも一つの属性を含まねばならない。ASSETの構成要素は、スキーマ-N_b中においては、NAMEという名前のスキーマで置換される。タブルTN_{ij}は、unnesttupの逆写像によって構成される。



$TN_{aj} \leq TN_{ba} \leq TN_r$
 $TN_{ba} = (\square \square \text{インスタンス-}N_{ap})$
 $\text{インスタンス-}N_{ap} = (TN_{aj} \quad TN_{aj+1})$

図. 3 UNNEST/NEST操作

4. 拡張組関係論理

(1) 定義

本節では、組関係論理に「タプル変数間の子孫-祖先関係」(のみ)を導入した拡張組関係論理を考え、select (選択) という関数の形式で定義する。

(select 関係-N VARDEF VARDSC COND) = 関係-S

VARDEF = ((QUANTIFIER TN_{ij} スキマ- N_i) ...) ; $i=1,2,\dots,n$; $j=1,2,\dots,m_i$;

VARDSC = ((dsc TN_{ij} TN_{ki}) ...) ; $1 \leq i,k \leq n$; $1 \leq j \leq m_i$; $1 \leq l \leq m_k$;

ここで、関係-Nと関係-Sは同一のスキーマを持つ。VARDEFはスキマ- N_i に対するタプル変数 TN_{ij} の宣言であるが、以下ではQUANTIFIERは“ \exists ”または“null”に限定して考える。VARDSCは、「タプル変数間の子孫-祖先関係」を定義し、これ(のみ)が正規形の組関係論理からの拡張となっている。即ち関数dscは、タプル TN_{ij} がタプル TN_{ki} の子孫(の一つ)であることを意味する(図. 3参照)。また、CONDは選択条件式を表わし、論理演算子、算術演算子、及び (val TN_{ij} ATTRIBUTESET) の形の式を含む(タプル変数はこの形式のみ使用される)。なお、選択される対象は、最上位レベルのインスタンス(インスタンス-N)に属するタプルに限定される。

(2) 問合せ変換

まず2つの非正規関係 関係-N、関係-Uを考え、(unnest 関係-N スキマ- N_a) = 関係-U かつ

(owner スキマ- N_a) = スキマ- N_b と仮定する。この時、関係-Nに対するselectを、それとある意味で等価な、関係-Uに対するselectに変換する関数transformqueryを与える。

SELEXPR-N = (select 関係-N VARDEF-N VARDSC-N COND-N)、

SELEXPR-U = (select 関係-U VARDEF-U VARDSC-U COND-U) とおくと、

(transformquery (SELEXPR-N スキマ- N_a)) = SELEXPR-U。

関数transformqueryの変換手順を以下に示す。

- ①スキーマ名のスキーマ- N_i をスキーマ- U_i で置換する ($i=1,2,\dots,n$)。但し、 $i=a$ は除く。
- ②スキーマ名のスキーマ- N_a をスキーマ- U_b で置換する。
- ③タプル変数の TN_{ij} を TU_{ij} で置換する ($i=1,2,\dots,n; j=1,2,\dots,m_i$)。但し、 $i=a$ は除く。
- ④タプル変数の TN_{aj} を TU_{bmb+j} で置換する ($j=1,2,\dots,m_a$)。(添字の“bmb+j”は、“b,mb+j”の意味である。)
- ⑤VARDSC-U中の表現 (dsc TU_{bmb+j} TU_{b1}) (上記③、④から派生)を削除する。また、代わりに新しい式 (val TU_{bmb+j} KEYATR-B) = (val TU_{b1} KEYATR-B) をCOND-Uに付加し、論理演算子“and”で連結する ($1 \leq j \leq m_a, 1 \leq l \leq m_b$)。ここでKEYATR-Bは、スキーマ- N_b の主キー属性である。

上記の変換の結果、SELEXPR-Uは以下ようになる。

```
VARDEF-U = ( (QUANTIFIER  $TU_{ij}$  スキーマ- $U_i$ ) ... )  $i=1,2,\dots,n; i \neq a;$   
            $j=1,2,\dots,m_i$  for  $i \neq b; j=1,2,\dots,m_b+m_a$  for  $i=b;$   
VARDSC-U = ( (dsc  $TU_{ij}$   $TU_{k1}$ ) ... )  $1 \leq i,k \leq n; i,k \neq a; 1 \leq j \leq m_i$  for  $i \neq b;$   
            $1 \leq l \leq m_k$  for  $k \neq b; 1 \leq j,l \leq m_b+m_a$  for  $i,k=b;$   
COND-U = ( [  $TU_{ij}$  に対する条件 ] and  
           [ (val  $TU_{bmb+j}$  KEYATR-B) = (val  $TU_{b1}$  KEYATR-B) ] ... )  
            $1 \leq j \leq m_a; 1 \leq l \leq m_b;$ 
```

関数transformqueryは、以下の補題を満たす。

[補題]

インスタンス-Nに含まれるタプル TN_r がSELEXPR-Nを満たす時、またその時に限り、以下が成り立つ。

- ①スキーマ- $N_b \neq$ スキーマ-Nの時、

$TU_r = (\text{unnest tup } TN_r)$ を満たしインスタンス-Uに含まれるタプルを TU_r とすると、 TU_r はSELEXPR-Uを満たす。

- ②スキーマ- $N_b =$ スキーマ-Nの時、

$TU_e = (\text{unnest tup } TN_{ae})$ かつ (owner (owner $TN_{ae})) = TN_r$ を満たしインスタンス-Uに含まれるタプル集合を $\{TU_e\}$ ($e=1,2,\dots$)とすると、 $\{TU_e\}$ 中の任意の TU_e はSELEXPR-Uを満たす。

[証明の概要]

インスタンス- N_i には、SELEXPR-Nのタプル変数に対応するタプル TN_{ij} ($i=1,2,\dots,n; j=1,2,\dots,m_i$)が存在する。この時、インスタンス- U_i には関数 (unnest tup TN_{ij}) で得られるタプル集合 $\{TU_{ij}\}$ が存在する。この関数が示す写像はtransformqueryにおけるタプル変数の置換と完全に合致し、また $\{TU_{ij}\}$ はVARDSC-Uを満たす。従って、 $\{TU_{ij}\}$ はSELEXPR-Uにおけるタプル変数を実体化したものとみなせるので、 $\{TU_{ij}\}$ はSELEXPR-Uを満たす。 TN_r は $\{TN_{ij}\}$ に含まれるので、 TU_r または TU_e は $\{TU_{ij}\}$ に含まれ、従ってSELEXPR-Uを満たす。なお、この証明は(dsc TN_{aj} TN_{b1})と(val TU_{bmb+j} KEYATR-B) = (val TU_{b1} KEYATR-B)とが同値であることに依存しているので、主キー属性の概念が本質的に重要である。

[命題]

上記のように、SELEXPR-Nと、それにtransformqueryを適用して得られるSELEXPR-Uとが与えられた時、
(unnest (select 関係-N VARDEF-N VARDSC-N COND-N) スキーマ- N_a) =
(select (unnest 関係-N スキーマ- N_a) VARDEF-U VARDSC-U COND-U) が成り立つ。

[証明の概要]

補題から証明できる。これにより、UNNEST(SELECT_N(R))=SELECT_U(UNNEST(R)) が示されたことになる。

5. 結論

非正規関係処理に関して、本報告で得られた結果とその特長とを以下にまとめる。

- 単純な拡張組関係論理を提案した。そこでは「タプル間の子孫-祖先関係」を用い、理解し易い問合せ表現が可能である。
- 非正規関係に対する問合せを正規関係に対する問合せに変換するアルゴリズムを提案し、その問合せ間の等価性を示した。この変換方式により、非正規関係を正規形の関係データベース上に実現するための基礎

が与えられる。

- リスト表現による非正規関係モデルを提案した。このモデルは、スキーマとデータとを同様に記述でき、複合対象としての非正規関係の構造を容易に表現できる。

参考文献

- [Abi84] Abiteboul, S. and N. Bidoit "Non First Normal Form Relations to Represent Hierarchically Organized Data," Proc. PODS, 1984, pp.191-200.
- [Ari83] Arisawa, H., K. Moriya and T. Miura "Operations and the Properties on Non-First-Normal-Form Relational Databases," Proc. VLDB, 1983, pp.197-204.
- [Dad86] Dadam, P. et. al. "A DBMS Prototype to Support Extended NF² Relations : An Integrated View on Flat Tables and Hierarchies," Proc. SIGMOD, 1986, pp.356-367.
- [Fis83] Fischer, P. C. and S. J. Thomas "Operators for Non-First-Normal-Form Relations," Proc. COMPSAC, 1983, pp.464-475.
- [Jae82] Jaeschke, G. and H. -J. Schek "Remarks on the Algebra of Non First Normal Form Relations," Proc. PODS, 1982, pp.124-138.
- [Kam82] Kambayashi, Y. et. al. "Representation of Relations for Database Output Utilizing Data Dependencies," Proc. Hawaii Intl. Conf. System Sciences, 1982, pp.69-78.
- [Kit82] Kitagawa, H. and T. L. Kunii "Form Transformer - Formal Aspects of Table Nests Manipulation -," Proc. Hawaii Intl. Conf. System Sciences, 1982, pp.132-141.
- [Kiy87] 木山、中野 「非正規形ビューに基づく検索の非正規形操作への変換法」 情報処理学会データベースシステム研究会、59-1、昭和62年5月
- [Lum82] Lum, V. Y. et. al. "OPAS: An Office Procedure Automation System," IBM Systems Journal, Vol.21, No.3, 1982, pp.327-350.
- [Lum85] Lum, V. Y. et. al. "Design of an Integrated DBMS to Support Advanced Applications," Proc. Foundations of Data Organization, 1985, pp.21-31.
- [Luo81] Luo, D. and S. B. Yao "Form Operation by Example -- a Language for Office Information Processing," Proc. SIGMOD, 1981, pp.212-223.
- [Mak77] Makinouchi, A. "A Consideration on Normal Form of Not-Necessarily-Normalized Relation in the Relational Data Model," Proc. VLDB, 1977, pp.447-453.
- [Pis86] Pistor, P. and F. Andersen "Designing a Generalized NF² Model with an SQL-Type Language Interface," Proc. VLDB, 1986, pp.278-285.
- [Sch82] Schek, H. -J. and P. Pistor "Data Structures for an Integrated Data Base Management and Information Retrieval System," Proc. VLDB, 1982, pp.197-207.
- [Sch85] Schek, H. -J. "Towards a Basic Relational NF² Algebra Processor," Proc. Foundations of Data Organization, 1985, pp.173-182.
- [Tsi82] Tsichritzis, D. "Form Management," CACM, Vol.25, No.7, July 1982, pp.453-478.
- [Tsu85] Tsuruoka, K. et. al. "PALET : A Flexible Office Form Management System," Journal of Information Processing, Vol.8, No.4, pp.280-287, March 1985.
- [Tsu87] 鶴岡、渡部、他 「オフィスフォーム自動生成システム(1)-(3)」 情報処理学会第34回(昭和62年前期)全国大会、7M-1~7M-3
- [Wat87] 渡部、鶴岡 「オフィスフォーム自動生成システム」 情報処理学会データベースシステム研究会、59-1、昭和62年5月
- [Yao84] Yao, S. B. et. al. "FORMANAGER : An Office Forms Management System," ACM T001S, Vol.2, No.3, July 1984, pp.235-262.