データベース・システム 58－2
（1987. 3. 16）

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

木山 稔　中野良平
（NTT 電気通信研究所）

非正規形モデルは柔軟なデータ構造を持ち、知識処理、データ工学等の分野で注目されている。本稿では、既存のデータベースを非正規形ビューを通してみせることを目的とし、非正規形操作を正規形操作に変換する方法について述べる。非正規形操作は関係論理を拡張したものを用い、非正規形ビューの定義はこの操作を以て行う。変換アルゴリズムは、変換の中継点として中間アルファを導入し、非正規形操作から中間アルファへの変換と、中間アルファから関係論理＋構造化操作への変換の２段階から成る。変換結果の形態は、既存のシステムを有効に活用できるものにした。

Reduction of NF2 Operation Based on NF2 View to Relational Calculus

Minoru KIYAMA　Ryohei NAKANO

NTT Electrical Communications Laboratories
1-2356, Take, Yokosuka-shi, Kanagawa-ken, 238-03 Japan

NF^2(Non First Normal Form) model has received considerable attention because of its flexible data representation. We intend to make it possible to use existing normal-form databases as NF^2 databases through NF^2 view. This paper mainly describes reduction algorithm from NF^2 operation to relational calculus and nest operation. NF^2 operation is extended relational calculus. NF^2 view is also defined by NF^2 operation. In reduction algorithm, we first create intermediate alpha from NF^2 operation, and then transform it into relational calculus and nest operation.

１．はじめに

関係モデルは第１正規形を前提としているため、実世界を自然に表現できない場合がある。非正規形モデルは、柔軟なデータ構造をもつことから実世界をより自然に表現でき、わかりやすいユーザインタフェースを提供することが可能であり、知識処理への応用も試みられている［１］。ここでは、既存の正規形データベースを非正規形ビューを通してみせることを考える。この結果、正規形データベースを非正規形（知識）の一部として利用することができ、また正規形データベースをユーザにわかりやすい形で提供することができる。

本稿では、主に非正規形操作を正規形操作に変換するアルゴリズムについて述べる。［２］では、問合わせを分解し、データベースシステムで実行した後、結果を非正規化して結合する方法が提案されているが、本変換では非正規形の結合を避け、問合せの分解は行わない方法をとる。

２章では非正規形の構造、操作について述べ、３章で非正規形ビューの定義法、４章では、非正規操作の正規形操作への変換法について説明する。

２．非正規形

２．１　構造

非正規形は関係モデルを一般化したものである。従来の関係モデルが属性値に単純値しか許さなかったのに対し、非正規モデルでは関係自体も属性値になり得る（図１）。

従業員

名前	給与	子供	
		名前	年令
a	１０	c１	１５
b	１２	c２	１０
		c３	２０
c	１５	c４	７
d	３０	c５	１０
e	２０	c６	５
		c７	１５

管理者

名前	給与	子供		役職	部下
		名前	年令		
d	３０	c５	１０	課長	a
					c
e	２０	c６	５	係長	b
		c７	１５		

図１　非正規形の構造の例

２．２　操作

非正規形の操作は、従来の正規形操作を拡張したものが幾つか提案されている［３，４］が、ここでは、正規形の関係論理［５］を非正規形に拡張したものを用いる［６］。関係論理の形式について簡単に説明した後、拡張点について述べる。

関係論理の基本要素は、アルファと呼ばれ次のような形式をしている。

目標リスト　：範囲規定　：条件式

範囲規定で、タプル変数（リレーション内を動く変数）の変域を指定し，条件式でタプル変数の満足すべ

き条件を指定する。条件式を満たすタプルだけが選択され、そのタプルのうち目標リストで指定された属性だけが取り出されアルファを構成する。アルファを解釈するとリレーションになる。条件式中の限量子（∀またはヨ）は範囲と組にした 限量子 範囲式（論理式）の形式に限る。また、属性値の参照は、タプル変数［属性］で行い、これを項と呼ぶ。

非正規形操作の関係論理からの主な拡張点は以下の２点である。

①非正規形では属性値としてリレーションをとり得るため、項を範囲として定義できる。この結果、項を変域とするタプル変数が出現する。

［例１］

すべての子供が１０歳以上である従業員名を得よ。

(x[1]):従業員(x):∀x[3](y)(y[2]>=10)

②目標リストにアルファを置くことを許容する。この結果リレーションを値とする属性を作ることが可能になる。

［例２］

１０歳以上の子供を持つ従業員名とその子供の名前、年令を得よ。

(x[1],((y):x[3](y):y[2]>=10)):従業員(x):()

なお、操作では簡略化のため、属性名の代わりに属性番号を用いる。

２．３　構造化操作

非正規形操作は関係代数に変換され、その実行結果を非正規化し、これを最終結果とする。正規形から非正規形を生成する操作として構造化操作を用いる。構造化操作<x1;(x2),(x3),…,(xk)>は属性x1（ネストバイ属性と呼ぶ）の値の等しいものを、属性 x2,x3,… ,xk ごとにまとめて１タプルとする。図２に構造化操作の例を示す。

親	子供	
	名前	趣味
a	c 1	野球
		水泳
	c 2	野球
		水泳
b	c 3	卓球
		野球

①
<---

親	子供	趣味
a	c 1	野球
a	c 1	水泳
a	c 2	野球
a	c 2	水泳
b	c 3	卓球
b	c 3	野球

②
--->

親	子供	趣味
a	c 1 — c 2	野球 — 水泳
b	c 3	卓球 — 野球

①：<#1,#2;(#3)><#1;(#2,#3)>
②：<#1;(#2),(#3)>

図２　構造化操作の例

3．非正規形ビューの定義

前章で定義した非正規形操作は正規形構造から非正規形構造を作ることが可能である。そこで、本稿では非正規形操作をビュー定義として用いる。図３に非正規形ビューの例を示す。便宜上、非正規形、正規形の構造をそれぞれ日本語、英字で表記する。なお、図中の管理者のように、非正規形の操作に正規形だけでなく既に定義したビューの使用を可能とする。このことは、ビュー定義の冗長性を削減する効果を持つ。

emp

name	sal	mname
a	１０	d
b	１２	e
c	１５	d
d	３０	−
e	２０	−

manager

name	post
d	課長
e	係長

child

name	pname	age
c１	a	１５
c２	b	１０
c３	b	２０
c４	c	７
c５	d	１０
c６	e	５
c７	e	１５

（ａ） 正規形リレーションの例

```
従業員＝
    (x[1], x[2], ((y[1],y[3])
                   :child(y)
                   :y[2]=x[1]
                  ))
    :emp(x)
    :()
```

```
管理者＝
    (x, y[2], ((u[1])
                :emp(u)
                :u[3]=y[1]
               ))
    :従業員(x),manager(y)
    :x[1]=y[1]
```

（ｂ） 図１と図３（ａ）を対応づける非正規形ビュー

図３ 非正規形ビューの例

4．操作の変換法

4．1 変換の概要

非正規形操作を正規形の関係代数表現に変換する方法について述べる。関係代数に変換する理由は、変換後の操作をデータベースマシンで実行することを想定しているためである。

（１）変換結果の形態

非正規形操作の結果は一般に非正規形リレーションになるので、関係代数に構造化操作を加わえる必要がある。変換結果として次の２形態が考えられ、各々別アプローチとなる。

①関係代数と構造化操作が混在（図４①）

②関係代数と構造化操作を分離（図４②）

①の操作は非正規形の関係代数になる。データが初めから非正規形構造で格納されている場合にも同じ変換法を使えるという利点はあるが、非正規形関係代数のインプリメントが必要になる。②は、まず正規形関係代数を実行し、その結果に対し構造化操作をおこなうので、既存システムの機能を有効に活用できる。本変換では、②を変換結果として作成することを目標とする。

①混在　　　　②分離

（実線は関係代数、破線は構造化操作）

図４　変換結果の形態

（２）変換経路

非正規形の操作は関係論理を拡張したものである。従って非正規形からいきなり関係代数に変換するより一旦関係論理に変換し、それを関係代数に変換する方が見通しがよい。関係論理から効率のよい関係代数を生成する変換法は既に確立されている［５，７］ので、本変換では非正規形操作を関係論理＋構造化操作に変換する部分を対象とする。

（３）変換の基本的考え方

非正規形操作と関係論理の差異という観点から変換の際の問題点は次の２点に整理できる。

①範囲規定中のアルファの処理。

②目標リスト中のアルファの処理。

今、中間アルファを「正規形アルファは中間アルファ。目標リスト中に中間アルファが存在するアルファは中間アルファ」と再帰的に定義する。中間アルファは①の問題を解決し、②のみを残したアルファと考えることができる。例えば、図３（ａ）のリレーションから、上司とその部下から成る非正規形リレーションを作るための操作は、

(x[1],((y[1]):emp(y):y[3]=x[1])):manager(x):()

であり、中間アルファである。これは、次の構造化操作を含む関係論理に対応する。

((x[1],y[1]):manager(x),emp(y):y[3]=x[1]))<#1;(#2)>

即ち、中間アルファからは比較的容易に関係論理と構造化操作に変換できる。

以上から、本変換法を大きく、非正規形操作から中間アルファへの変換と、中間アルファから関係論理＋構造化操作への変換に分けて考える。

４．２　変換アルゴリズム

具体的な変換アルゴリズムについて述べる。簡略化のため、非正規形関係論理、中間アルファ，正規形関係論理，正規形関係代数、構造化操作（射影を含む）をそれぞれα_{NF2}、α_{T}、α_{1NF}，Ａ、Ｎと表記する。また、α_{NF2}の範囲規定内の∧、∨、∧～（それぞれリレーションの積、和、差集合を表す）を範囲演算と呼ぶ。

［定理］　非正規形ビューに対する任意の非正規形操作α_{NF2}に対し関係代数表現ＡＮが存在する。

（証明）

以下の処理手順によりα_{NF2}をＡＮに変換可能である（図５）。

・α_{NF2}の範囲規定に範囲演算を含まない場合

［変換則１］によりα_{NF2}はα_{T}に変換でき、次いで［変換則２］によりα_{T}はα_{1NF}Ｎに変換でき、最後に関係論理／関係代数変換によりＡＮに変換できる。

・α_{NF2}の範囲規定に範囲演算を含む場合

α_{NF2}を t:(r_1 B r_2)(x):f（Bは範囲演算）とすると［変換則３］により t:r_1(x):f と t:r_2(x):f のふたつのα_{NF2}に分解し、それぞれのα_{NF2}を関係代数と構造化操作に変換し，A_1N_1，A_2N_2を得、その後、再び［変換則３］によりひとつのＡＮにまとめる。

（証明終り）

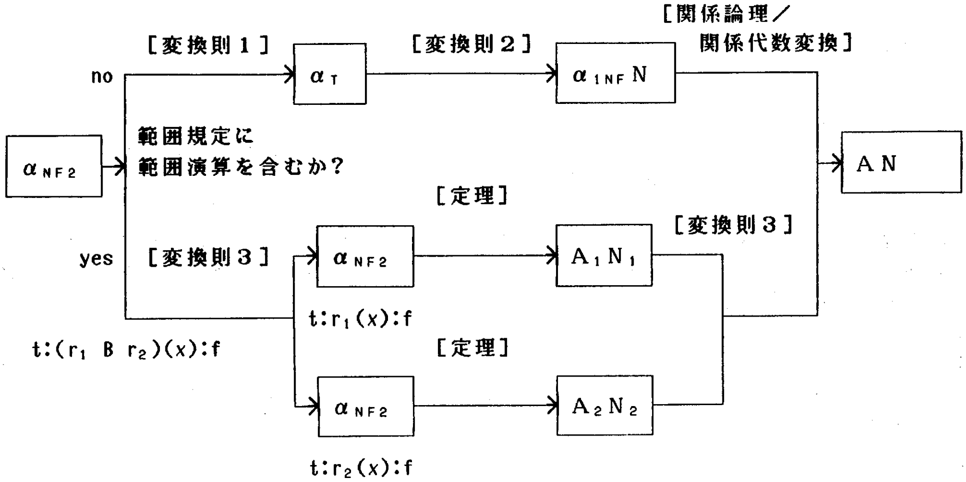

図５　非正規形操作から関係代数＋構造化操作への変換の流れ

<u>[変換則１]</u>　非正規形ビューに対する任意の非正規形操作α_{NF2}はα_Tに変換できる。但し、α_{NF2}の範囲規定は範囲演算を含まない。

α_{NF2}を $t:r(x):f$ とする。まず、範囲規定に非正規形ビューを代入する。その結果、元のα_{NF2}は、$t:(t_1:r_1(x_1):f_1)(x):f$ のような形になる。次に、範囲規定内のアルファ（閉α_{1NF}以外）を対象に以下の変換を施し、アルファを平坦化する。

$t:(t_1:r_1(x_1):f_1)(x):f \; = \; t':r_1(x_1):f_1 \wedge f'$

但し、 t'は t中の項$x[j]$をt_1中のj番目の項またはアルファで置換したもの
f'は f中の項$x[j]$をt_1中のj番目の項またはアルファで置換したもの

上記の代入、平坦化の処理を範囲がα_{1NF}になるまで繰り返す。この結果、α_{NF2}内のすべてのアルファはα_Tとなる。さらに、限量子に付帯した範囲が閉α_{1NF}以外の時に以下の変換を施す。

$\forall(t:r(x):f)(y)(g) \; = \; \forall r(x)(\sim f \vee g')$
$\exists(t:r(x):f)(y)(g) \; = \; \exists r(x)(f \wedge g')$

但し、g'はg中の項$y[j]$をt中のj番目の項またはアルファで置換したもの

限量子に付帯した範囲がα_{1NF}になるまでこの処理を繰り返す。

以上の変形により、α_{NF2}をα_Tに変換可能である。図６（ａ）にα_{NF2}をα_Tに変換する例を示す。

なお、ビュー定義はα_{NF2}で行われているので、あらかじめ［変換則１］を使ってこれをα_Tに変換しておく。これにより、問合せ操作の変換時に、ビューのα_Tへの変換が不要となる。

<u>[変換則２]</u>　任意のα_Tは$\alpha_{1NF}N$に変換できる。

まず、α_Tを次の変形を使い$\alpha_{T1}N_1$に変換する。

$(t_1, (t_2: r_2(x_2): f_2)):r_1(x_1):f_1 \; =$
$\qquad ((t_1, \Delta, t_2):r_1(x_1), r_2(x_2):f_1 \; \wedge \; f_2) < t_1, \Delta; (t_2) > [t_1, t_2]$

但し、$\Delta =$ （f_2に現れる$x_1[i]$の集合）－（t_1に現れる$x_1[i]$の集合）

さらに、α_{T1}に同じ変形を適用して$\alpha_{T2}N_1N_2$を得る。この処理をα_{Ti}がα_{1NF}になるまで繰り返す。Δを構造化操作におけるネストバイ属性に含める点に注意。従って、構造化操作の対象となるα_{1NF}においてΔを取り出すことと、構造化操作の終った後にΔを除去することが必要になる。但し、Δが空の場合は、構造化操作後の射影は省略できる。図６（ｂ）にα_Tを$\alpha_{1NF}N$に変換する例を示す。

①非正規形の操作

［説明］図１の構造に対する操作（例２）

(x[1], ((y):x[3](y):y[2]>=10)):従業員(x):()

②ビュー定義（従業員）代入

［説明］図３（ｂ）中の従業員のビュー定義を代入する。

(x[1], ((y):x[3](y):y[2]>=10))

:((u[1],u[2],((v[1],v[3]):child(v):v[2]=u[1])):emp(u):())(x):()

③範囲規定中のアルファの平坦化処理

［説明］empを範囲とするアルファを平坦化する。empは外側のアルファの範囲とし、対象リスト中の項x[1],x[3]はそれぞれu[1],((v[1],v[3]):child(v):v[2]=u[1]))で置換する。

(u[1], ((y):((v[1],v[3]):child(v):v[2]=u[1])(y):y[2]>=10)):emp(u):()

④範囲規定中のアルファの平坦化処理

［説明］childを範囲とするアルファを平坦化する。childは外側のアルファの範囲とし、対象リスト中の項yはv[1],v[3]で置換する。また、条件式中の項y[2]はv[3]で置換する。

(u[1], ((v[1],v[3]):child(v):v[2]=u[1] ∧ v[3]>=10)):emp(u):()

（ａ）非正規形から中間アルファへの変換例

⑤目標リスト中のアルファを構造化操作に置換

［説明］Δが空であるため、構造化操作後の射影は省略することができる。

((u[1], v[1], v[3]):emp(u),child(v):v[2]=u[1] ∧ v[3]>=10)<#1;(#2,#3)>

（ｂ）中間アルファから関係論理＋構造化操作への変換例

図６　操作の変換例

［変換則３］　範囲規定に範囲演算を含む任意のα_{NF2}を

$t:(r_1\ B\ r_2)(x):f$　（Bは範囲演算、ｔは統計関数を含まない）

とし、 $t:r_1(x):f$, $t:r_2(x):f$ を関係代数と構造化操作に変換した結果をそれぞれA_1N_1, A_2N_2 とするとき、$N_1=N_2$（$=N$）が成立し、

$t:(r_1\ B\ r_2)(x):f = A_1N_1\ [B]\ A_2N_2 = (A_1[B]A_2)N$

但し、[B]はBが∧、∨，∧～のときそれぞれ[＊],[＋],[－]

範囲規定に範囲演算を含む場合、α_{NF2}にそのまま［変換則１］を適用してもα_Tには成り得ない。範囲演算の対象となる非正規形は同一の構造を持つことが要求されるため、構造を一致させるための操作（アルファ）が範囲演算の前に必要となり、このアルファをむやみに壊すことはできない。従って、［変換則３］のように一旦α_{NF2}を範囲演算を含まない２つのα_{NF2}に分解し、それぞれをα_{1NF}と構造化操作に変換したした後に２つの結果を結合してひとつの関係代数＋構造化操作に変換する方法をとる。

5．非正規形ビューの利用

以上のような変換機構は、図７のシステム構成の一要素として使用できる。フレーム論理［６］はフレームモデルと関係モデルを統合したモデルであり、その構造は非正規形に写像できる。従って、非正規形ビューを用いれば、既存のデータベースをフレーム論理構造と同一視することができ、知識の一部としてデータベースをアクセス可能である。

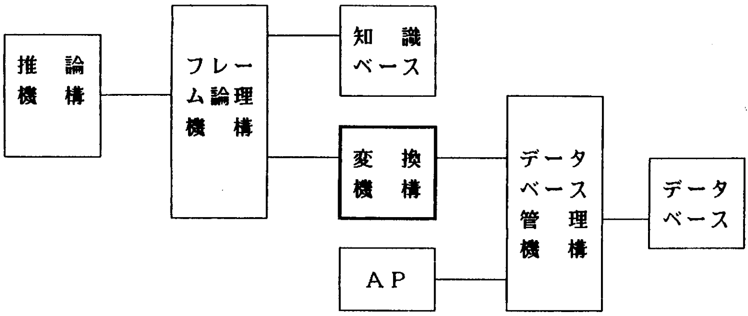

図７　システム内での変換機構の位置

６．おわりに

非正規形ビューを使い、既存データベースを非正規形としてみせる方法について検討した。非正規形ビューは関係論理を拡張した非正規形操作を使って定義可能であることを示した。。また、非正規形ビューに対する操作を関係論理と構造化操作に変換するアルゴリズムを考案した。現在、変換プログラムを作成し、具体例による検証を進めている。今後は、変換結果の実行効率の評価、最適化処理等を行う予定である。

［参考文献］

［１］金枝上,三石,加藤：知識ベース管理システムＫＡＰＰＡのデータモデル，情報処理学会第３３会全国大会 5M-6, 1986.

［２］山本,上林：関係データベースの非正規出力のための最適化，データベースシステム研究会，86-DB-54, 1986.

［３］P.Fischer,S.Thomas: Operators for Non-First-Normal-Form Relations, Proc. Computer Software Application Conf., Nov. 1983.

［４］P.Pistor,F.Andersen: Designing a Generalized NF^2 Model with an SQL-type Language Interface, Proc. VLDB 86, Kyoto, Aug. 1986.

［５］中野,斉藤：ルールに基づいた関係論理／関係代数変換法，データベースシステム研究会，86-DB-54, 1986.

［６］中野,木山：フレーム論理，知識工学と人工知能研究会，87-AI-50, 1987.

［７］中野,斉藤：集約関数まで拡張した関係論理／関係代数変換法，データベースシステム研究会，87-DB-57, 1987.