置換群論的考察によって関係データベースの情報を分解することの意味

杉 本 英 二

目　　次

1. はじめに

今日のデータベースの設計において最も大きな影響を与えたのは，数学概念の関係でデータベースを記述しようとする関係データベースである。1970年，E. F. Codd[1]によって提案されて以来今日に至るまで，データベースの形式的取扱いが研究され続けている。これらの研究の中でも最近，関係データベースに随伴する置換群を定義して，その関係データベースがどれだけの質問に答えられるかを明確にしようという理論が，J. Paredaens によって提案された。[2]

(1) Codd, E. F. "A Relational Model of Data for Large Shared Data Banks," *Comm. ACM,* 13, 6, 1970.

(2) Paredaens, J. "On the Expressive Power of the Relational Algebra," *Information Processing Letters*, 7, 2, 1978.

この理論によれば，勝手に与えられた関係データベースが持っているすべての情報はこれに随伴する置換群で示される，さらに相異なる二つの関係データベースのそれぞれの随伴群が一致するなら，二つの関係データベースのもつ情報は等しいことがわかるというものである。

データベースの設計者にとって，この理論は非常に有効である。すなわち，まったく別々の異なるデータベースが，同一の質問が与えられるといつも同一の回答を示すかどうかを判定する手段を提供する。実は，この理論の有効性は，同じ情報をもつデータベースの中で最も簡単なデータベースを導びく手がかりを与えるという設計手段を与える点にある。

ＮＨＫ総合技術研究所の相沢は随伴群を不変にする関係データベースの変換を与え，これによって関係データベースの正規化の手続は情報を保存しながら，つぎつぎと基本的な関係へ分解することだと意味づけた。[3]

本稿は，シンボルの置換という形式的方法によって与えられるデータベースの分解がどのような意味を与えるのか，この分解が従来のデータベース正規化とどのように異なるのかを明らかにするものである。

2. 基本概念

関係データベースの基本的概念を紹介する。これは大別して，データベースのモデルに関する事柄とデータベース操作言語に関する事柄の二つある。

（i） データベース・モデル

関係Ｒとは，対象のクラスC_1，$C_2\cdots$，C_nの直積の有限部分集合：

$$R \subseteq C_1 \times C_2 \times \cdots \times C_n.$$

Ｒの要素をｎ組とよぶ。例１はこの様子を示している。

直観的には，関係Ｒはファイル，ｎ組はそのレコードに対応している。

関係データベースとは関係の有限集合のこととする。

関係Ｒの第ｊデータ域D_j(Ｒ)を次式で定義：

(3) 相沢輝昭，“関係データベースの等価変換”，情報処理投稿中。

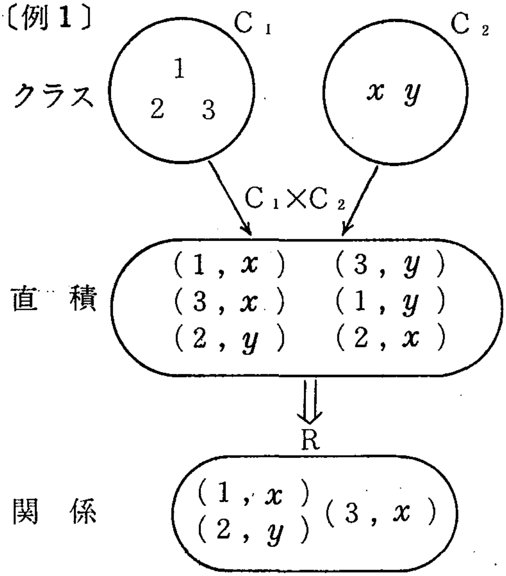
$$D_j(R)=\{\gamma_j \mid (\gamma_1,\cdots,\gamma_j,\cdots,\gamma_n)\in R\}.$$

これは Codd が定義した active domain[1] と一致している。

関係Rのデータ域D(R)を次式で定義：

$$D(R)=\bigcup_{j=1,n} D_i\ (R).$$

〔例1〕



ある与えられたデータベースを$\{R_1, R_2, \cdots, R_p\}$ とすれば，これのデータ域を次式で定義：

$$D(R_1,\cdots,R_p)=\bigcup_{i=1,p} D(R_i).$$

〔例2〕 例1のRで考える。

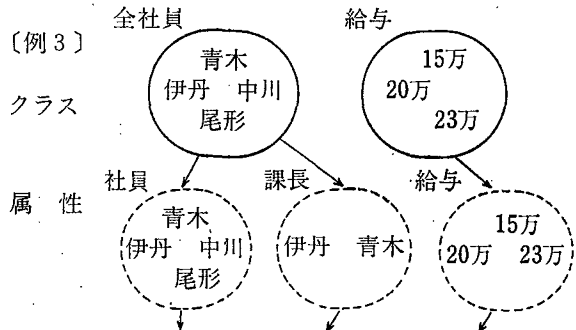
$D_1(R)=\{1,\ 2,\ 3\}$

$D\ (R)=\{1,\ 2,\ 3\}\ \cup\ \{x, y\}$

$\quad=\{1,\ 2,\ 3,\ x, y\}$ //

関係Rの属性A_1，A_2，…，A_n とは，関係R を構成するクラスC_1，C_2，…，C_n を一意に識別できるようにつけられた役割名とする。例3にこれを示す。

〔例3〕



属性名	社　員	課　長	給　与
関係表	青　木	青　木	20　万
	伊　丹	伊　丹	23　万
	尾　形	伊　丹	15　万
	中　川	青　木	15　万

社員関係をRとすれば，

$$R \subseteq C_1 \times C_1 \times C_2$$

のことであるが，第1成分 C_1，第2成分 C_2 の関係Rで果している役割を，社員属性，課長属性で表示するのが属性，属性名である。しかしながら，これはRを解釈するときにつけられた都合上の名前であって，成分の識別が十分つけれる機構があれば，特に属性を導入する必要はない。データベースの設計を意味概念の方面から行なうならば，この属性や役割が主体となるが，本稿では主に形式論を展開するので，概念的設計論は省略する。 //

(ii) データベース操作言語

データベース操作言語族には大別して6種の族がある。[4]

(1) 関係代数型[1]

(2) 関係論理型[5]

(4) 植村俊亮,『データベースシステムの基礎』, オーム社, 1979.
(5) Codd, E. F. “A Data Base Sublanguage Founded on the Relational Calculus,” Proc. 1971 ACM SIGFIDET Workshop, ACM, 1971.

(3) 写像型[6]

(4) 要素操作型[7]

(5) 例題型[8]

(6) 自然言語型[9]

これらの詳細については各論文にまかせるとして，これらの言語族は情報表現力においてどれも等価であることが言える。特に Codd が関係完備の概念を導入して[10]以来，表現力について注意が十分されるようになった。

以上を前提として，関係代数型の言語をこれからの議論の基礎とする。すなわち，関係代数言語で成立する事柄は，他の言語に拡張できることを予定している。

関係代数言語は，Codd が数学上の関係をデータベースのモデルとして採用したときにそれらの関係を操作する数学上の作用素を基礎に，データベースの操作言語として体系づけられたものである。Codd は，集合演算（すなわち，union，intersection，difference，cartesion product，selection）と，関係演算（すなわち，projection，join，division）を関係代数言語の演算として提案した。実は理論的取扱いを簡単にするため，これを整理すれば，以下の5種の演算で足りることが証明されている。[10][11]

(1) 関係RとSの直積R×Sを次式で定義：

$$R \times S = \{\widehat{rs} \mid r \in R,\ s \in S\}.$$

$\widehat{rs}$とはrとsをこの順序で並べ置いたもの。

---

(6) Astrahan, M. M. and Chamberlin, D D. "Implementation of a Structured English Query Language," *Comm. ACM,* 18, 10, 1975.

(7) Lorie, R. A. "XRM-An Extended (N-ary) Relational Memory," IBM Cambridge Scientific Center, G 320-2096, 1974.

(8) Zloof, M. M. "Query by Example," Proc. AFIPS NCC, 44, 1975.

(9) Codd, E. F. "Seven Steps to Rendezvous with the Casual User," in *TC2 74,* North-Holland, 1974.

(10)——"Relational Completeness of Data Base Sublanguages," in Courant Computer Siencec Symposius 6, *Data Base Systems,* Prentice-Hall, 1972.

(11) Paredaens, J. "Some Results on the Relational Algebra," MBLE R335, 1976.

(2) 関係RとSの合併R∪Sは通常の集合和とする。

(3) 関係Rの座標$(j_1,\cdots,j_g)$への射影

$R\pi(j_1,\cdots,j_g)$ を次式で定義：

$$R\pi(j_1,\cdots,j_g)=\{(\gamma_{j1},\cdots\gamma_{jg})\mid(\gamma_1,\cdots\gamma_n)\in R\}.$$

もちろん，j_iは$1\leq j_i\leq n$を満たす。また$j_1\cdots j_g$の中には同じ添字が繰返されることもある。

(4) 関係Rの座標 j_1，j_2 上の一致制限

$(R\mid j_1=j_2)$ を次式で定義：

$$(R\mid j_1=j_2)=\{(\gamma_1,\cdots,\gamma_n)\in R\mid\gamma_{j_1}=\gamma_{j_2}\}.$$

(5) 関係Rの座標j_1，j_2上の不一致制限

$(R\mid j_1\neq j_2)$を次式で定義：

$$(R\mid j_1\neq j_2)=\{(\gamma_1,\cdots,\gamma_n)\in R\mid\gamma_{j_1}\neq\gamma_{j_2}\}.$$

これらの5種の演算を有限回繰返してデータベースから取出せる情報は，他の言語を使って取出せる情報と同等であることを前提に検索可能を定義する。

すなわち，関係Sがデータベース$\{R_1,\cdots,R_p\}$から検索可能であるとは，Sが$R_1,\cdots,R_p$に上記の5種の演算を有限回適用して得られることと定義する。データベース$\{R_1,\cdots,R_p\}$から検索可能である関係の全体を基本情報$BI(R_1,\cdots,R_p)$と表わす。

3. 関係の随伴群（Paredaensの理論）

簡単のため$p=1$の場合について述べる。

D(R)上の対称群を$\mathcal{S}(R)$と書く。

$\sigma\in\mathcal{S}(R)$がR両立とは，次式の成立と同値である：

$$(\forall(\gamma_1,\cdots,\gamma_n)\in R)\ (\sigma(\gamma_1),\cdots,\sigma(\gamma_n))\in R.$$

Rの随伴群CG(R)を次式で定義：

$$CG(R)=\{\sigma\in\mathcal{S}(R)\mid\sigma\text{はR両立}\}.$$

Paredaensは，CG(R)を群として把握することよりも，特殊な関係――Rの随伴群関係――として次のようにとらえた：

$D(R)=\{d_1, d_2, \cdots, d_k\}$,

$CG(R)=\{\sigma_1, \sigma_2, \cdots, \sigma_l\}$

とすれば，Rの随伴群関係 $C\widetilde{G}(R)$を

$$\begin{pmatrix} \sigma_1(d_1), & \sigma_1(d_2), \cdots, \sigma_1(d_k) \\ \vdots & \vdots \\ \sigma_l(d_1) & \sigma_l(d_2), \cdots, \sigma_l(d_k) \end{pmatrix}$$

とする。

〔例4〕

$$R_1=\begin{pmatrix} a & d & a \\ b & a & c \\ a & e & a \\ c & a & b \end{pmatrix} \qquad C\widetilde{G}(R_1)=\begin{pmatrix} a & b & c & d & e \\ a & c & b & d & e \\ a & b & c & e & d \\ a & c & b & c & d \end{pmatrix} \quad //$$

補題1 $C\widetilde{G}(R)\in BI(R)$. //

証明は本稿にとって長すぎるので省略する。[2]

補題2 R, S_1, S_2を3つの関係とし，

$C\widetilde{G}(R)\subseteq \widetilde{C}G(S_1)$, $C\widetilde{G}(R)\subseteq C\widetilde{G}(S_2)$

とする。このとき，

(1) $C\widetilde{G}(R)\subseteq C\widetilde{G}(S_1\cup S_2)$.

(2) $C\widetilde{G}(R)\subseteq C\widetilde{G}(S_1\times S_2)$.

(3) $C\widetilde{G}(R)\subseteq C\widetilde{G}(S_1\pi(j_1, \cdots, j_g))$.

(4) $C\widetilde{G}(R)\subseteq \widetilde{C}G(S_1 \mid i=j)$.

(5) $C\widetilde{G}(R)\subseteq C\widetilde{G}(S_1 \mid i\neq j)$. //

証明は容易であるので省略する。

定理3 $S\in BI(R)\Leftrightarrow C\widetilde{G}(R)\subseteq C\widetilde{G}(S)\ \&\ D(S)\subseteq D(R)$. //

証明：$\Rightarrow D(S)\subseteq D(R)$は自明。$C\widetilde{G}(R)\subseteq C\widetilde{G}(S)$はRからSを導くのに必要な5種の演算に関する補題2より明らか。

$\Leftarrow$補題1より，$CG(\widetilde{R})\in BI(R)$. そこで，Sを$CG(R)$から導びこう。$s\in S$とする。$C\widetilde{G}(R)$の各要素は，その成分に$D(R)$の要素をすべて含むので，$D(S)$の要素をすべて含む。よって，$C\widetilde{G}(R)$の適当な射影$T_s$で$s$

$\in T_s$となるものが存在する。今，$t \in C\widetilde{G}(R)$の射影がsに等しいとすると$\forall t' \in C\widetilde{G}(R)$，$\exists \sigma \in CG(R)$，$t' = \sigma(t)$．ところが$C\widetilde{G}(R) \subseteq C\widetilde{G}(S)$により，$\forall s' \in T_s$，$\exists \sigma \in CG(R) \subset CG(R)$，$s' \in \sigma(s)$ よって，$S = \bigcup_{s \in S} s \subset \bigcup_{s \in S} T_s \subset S$となるから，$S = \bigcup_{s \in S} T_s$が成立する。すなわち，$S \in BI(R)$である。 //

〔例5〕例4を参照

$$S_1 = \begin{pmatrix} b \\ c \\ d \\ e \end{pmatrix}, \quad S_2 = \begin{pmatrix} b \\ c \\ d \end{pmatrix}$$ とすると，

$CG(S_1)$は$\{b, c, d, e\}$上の対称群，

$CG(S_2)$は$\{b, c, d\}$上の対称群となり，

$C\widetilde{G}(R_1) \subset C\widetilde{G}(S_1)$ゆえに$S_1 \in BI(R_1)$．

一方，

$(a, c, b, c, d) \in C\widetilde{G}(R_1) - C\widetilde{G}(S_2)$ゆえに$S_2 \notin BI(R_1)$．すなわち，$S_1$は$R_1$から検索されるが，$S_2$は$R_1$から検索できないことがわかる。

この結果は，R_1，S_1，S_2をながめただけでは自明ではない。 //

Paredaens は定理3の系を提出したが，われわれは，この系をPを一般に拡張して定理として示す。証明は容易であるので省略する。

定理 (Paredaens) 2つのデータベース$\{R_1, \cdots, R_p\}$，$\{S_1, \cdots, S_q\}$に対して，

$$BI(R_1, \cdots, R_p) = BI(S_1, \cdots, S_q) \Leftrightarrow C\widetilde{G}(R_1, \cdots, R_p) = C\widetilde{G}(S_1, \cdots, S_q)$$
$$\& \ D(R_1, \cdots, R_p) = D(S_1, \cdots, S_q).$$

ここに使われている記号$C\widetilde{G}(R_1, \cdots, R_p)$は，

データベース$\{R_1, \cdots, R_p\}$の随伴群の関係であって，これは次式で定義される。

$$CG(R_1, \cdots, R_p) = \{\sigma \in \mathcal{S}(R_1, \cdots, R_p) \mid \sigma は R_i 両立，1 \leqq i \leqq p\}.$$

明らかに$CG(R_1, \cdots, R_p) = \bigcap_{i=1,p} CG(R_i)$．

特に$\mathcal{S}(R_1, \cdots, R_p)$は，$D(R_1, \cdots, R_p)$上の対称群である。

このParedaensの定理は,「データベースの随伴群が等しいなら,二つのデータベースの基本情報は等しい」という重要な知見をもたらしてくれる。実際基本情報が等しいかどうかを判定するアルゴリズムは,5種類の関係代数演算を有限回適用して得られる関係の全体だという基本情報の定義からは得られないものである。

以上がParedaensの理論の骨子である。

4. 関係データベースの設計

(i) データ独立の概念と意味

関係データベースの原点を一担振返ってみよう。まずCoddの提案以前の歴史的な二つの提案を紹介しておく。第1は,CODASYLによる情報代数[(12)]である。これは非手続的で抽象化されたデータ処理方式の提案であったが,Coddの提案まで無視され続けた。第2は,Bachmanによるデータベース構築技法[(13)]を採用したIDSの出現である。この方法はデータベースの基本技法として今日に至るまで,データの論理的構造を物理的装置上に記憶させるために使われてきている。前者は利用者の使用する言語としての方向を,後者はデータの蓄積モデルとしての方向を示している。一方が無視され,他方が主流となる状態がCoddの関係データベースの提案で終止したことは,Coddの提案するモデルがいかに強力であったかを示している。この強力な説得力は彼が提案した「データ独立」の概念にある。それは,結局のところ「プログラムとデータベースとの中間に標準的な機構を設けて両者を隔離すること」である。それが意味するところは,「各企業体において一旦開発された数百,数千個ものプログラムが,データベースの性能向上のために,あるいは企業体の環境変化のためにデータの間にある現実の関係が変化し蓄積しているデータが事実を伝えなくなるために発生するデータベースの修正・改善によって,実動しなくなるとい

(12) CODASYL, "An Information Algebra Phase 1 Report-Language Structure Group of the CODASYL Development Committee," *Comm. ACM,* 5, 4, 1962.

(13) Bachman, C. W. and Williams, S. B. "A General Purpose Programming System for Randam Access Memories,"Proc. AFIPS FJCC, 26, 1964.

う事態を最小限に防止する」ことにある。

（ii） データベースの正規化

データベースの正規化は，単にそこからデータを検索するだけのことなら重要でない。しかし，データベースへデータを追加すること，すでに蓄積されたデータを更新すること，削除することのように，記憶の状態を変えようとするとき，重大な問題が発生する。その典型例でこの問題を解説しておこう。

〔例6〕 部品受注関係表

部品番号	部品名	会社名	会社住所	受注量
1	電球	松下	福岡	300
2	電池	東芝	東京	400
3	ソケット	松下	福岡	200
1	電球	日立	札幌	200

追加　この会社は新しく山陽会社と受注契約をしたが，まだどの部品を受注するかは決っていない。だから，この山陽会社が得意先になったことをこのデータベースに記録できない。

更新　松下会社は住所を福岡から大阪へ移したので，データベースの記録を変える必要が生じた。そこで第1行目のレコードの福岡を大阪にしたがそれだけで済むであろうか？第3行目のレコードも更新する必要がある。この場合このデータベースを全部調べる必要が出てくるが，実際の企業で使われている数千，数万のレコードを調べ上げることは経済的でない。さらに更新が必要な全レコードを書きなおすまでには時間がかかるが，その間（松下，福岡）と（松下，大阪）の二つの記述が存在することになり，記録に矛盾が生じる。

削除　この会社は電池を取扱わなくなったので，第2行の電池に関するレコードを削除しようとした。ところが，これを削除すると（東芝，東京）という会社と住所の情報も消えてしまう。この情報の削除により，東芝は得意先会社でなくなることになり，不都合である。この事態は異なる二つの情報が並記されていることに原因があると考えられよう。 //

以上の例によって，データベースの設計が現実の必要性を満たしていないこ

とがわかる。そこで，以上のような不都合が生きないように設計すれば，例7のようになる。

〔例7〕

部品関係表

部品番号	部品名
1	電球
2	電池
3	ソケット

会社関係表

会社名	会社住所
松下	福岡
東芝	東京
日立	札幌

受注関係表

部品番号	会社名	受注量
1	松下	300
1	日立	200
2	東芝	400
3	松下	200

//

〔例8〕例7に例6の追加，更新，削除を行った結果を示す。

部品関係表

部品番号	部品名
1	電球
3	ソケット

会社関係表

会社名	会社住所
松下	大阪
東芝	東京
日立	札幌
山陽	小樽

受注関係表

部品番号	会社名	受注量
1	松下	300
1	日立	200
2	東芝	400
3	松下	200

例6の不都合と比較して，追加，更新，削除を例7で試みてみれば，このデータベースの設計の利点が十分理解される。　//

実は，例7，8はCoddが第3正規形と定義したもので，正規形の最終段階の関係である。この三つの関係をつなげば例6の関係を作り出すことができるので，例7は例6の関係よりもより基本的な情報単位に分解された関係であると言えよう。詳細はCoddなどの文献[(4)(14)(15)]を参照することにして，以上の分解は射影演算で行ない，以上の結合は直積と一致制限演算の組合せ，すなわち結合演算で行なう。正規化の手続も，正規化された関係からもとの情報を復元する手

(14) Codd, E. F. "Further Normalization of the Data Base Relational Model," in Courant Computer Science Symposium 6, *Data Base Systems*, Prentice-Hall, 1972.

(15) Date, C. J. "*An Introduction to Database Systems*," Addison-Wesley, 1975, or, 2nd-ed. 1977.

〔例9〕　関係 R_0

社　員	職　種	勤務地
フジ三太郎	営　業	東　京
ジョージさだお	営　業	大　阪

射影

関係 R_1

社　員	職　種
フジ三太郎	営　業
ジョージさだお	営　業

関係 R_2

職　種	勤務地
営　業	東　京
営　業	大　阪

結合

関係R

社　員	職　種	勤務地
フジ三太郎	営　業	東　京
フジ三太郎	営　業	大　阪
ジョージさだお	営　業	東　京
ジョージさだお	営　業	大　阪

続も関係代数言語の演算で行われることはシステムとして首尾一貫した点は評価できる。ところがこれらの演算はデータの意味をいっさい考慮しない形式的な操作をするので例9のように無意味な演算結果をもたらすことがある。

$R_1 = R_0 \pi$（社員，職種）

$R_2 = R_0 \pi$（職種，勤務地）

$R = R_1$（職種＝職種）R_2

この例はRが R_0 に一致すると期待されるのであるが，裏切られる例で，結合のわなと呼ばれている。[4]

例9は Fagin の第4正規形[16]を考慮すれば防ぐことができる。現在までのデータベースの研究では第4正規形を最終結果としているが，本稿では正規化の理論を述べるのは目的としていないので割愛する。しかし，正規化によって情報が整理され，直観的に納得のいく情報単位に関係が分解されることがわかる。

5. データベースの等価変換（相沢の拡張）

随伴群を不変に保つ関係データベースに対する演算ないしは操作を等価変換とよぼう。

直積を取ったり，逆にそれを成分に分解したりしても本質的情報が損われないことは直観的にも理解できる。これを命題1とする。n項関係Rにおいて2つの項が全く重複している場合その重複を落しても情報は変らないだろう。これを命題2とする。群論では軌道分割という概念があるが，これを適用してタテ長の表をいくつかヨコ方向に分断してそれぞれを関係としても情報はもともとあったものである。これを命題3とする。特に命題3は従来データベース技術では考えられなかった新しい結果である。Codd の正規化の方法の基本操作は，射影による関係の分割であって，その分割によっても情報は不変である。これを命題4とする。

命題1[3]　2つの関係 R_1, R_2 に対して，

$$CG(R_1\times R_2)=CG(R_1,\ R_2). \qquad //$$

証明：$\sigma\in CG(R_1\times R_2)$

$\Leftrightarrow(\forall(\gamma_1,\ \gamma_2)\in R_1\times R_2)\ (\sigma(\gamma_1),\ \sigma(\gamma_2))\in R_1\times R_2$

$\Leftrightarrow(\forall\gamma_i\in R_i)\ \sigma(\gamma_i)\in R_i\quad i=1,\ 2$

$\Leftrightarrow\sigma\in CG(R_i)\qquad i=1,\ 2$

$\Leftrightarrow\sigma\in CG(R_1,\ R_2)\qquad //$

命題2[3]　n項関係Rに対して，

(16) Fagin, R "Multivalued Dependencies and a New Normal Form for Relational Database Design," *ACM TODS,* 2, 3, 1977.

$$R^j=\{(\gamma_1,\cdots,\gamma_{j-1},\gamma_{j+1},\cdots,\gamma_n)\mid(\gamma_1,\cdots,\gamma_n)\in R\}$$

とおく。もし$CG(R^j)\subset CG(R)$ならば，すなわちR^j両立な置換が同時にR両立であるならば，

$$CG(R^j)=CG(R). \quad //$$

証明：補題2の(3)により，$CG(R)\subseteq CG(R^j)$。

ところが，命題の条件は$CG(R)\subseteq CG(R^j)$を意味している。//

命題3 (3)　関係Rの$CG(R)$による軌道分割を

$$R=R_1\cup\cdots\cup R_m$$

とすれば，$CG(R)=CG(R_1,\cdots,R_m)$。 //

証明：$\sigma\in CG(R)\Leftrightarrow(\forall\gamma\in R)\ \sigma(\gamma)\in R$

$\Leftrightarrow(\forall i\ \forall\gamma_i\in R_i)\ \ \sigma(\gamma_i)\in R_i$

$\Leftrightarrow(\forall i)\ \ \sigma\in CG(R_i)$

$\Leftrightarrow\sigma\in CG(R_1,\cdots,R_m)$ //

再度，軌道分割を補足しておこう。Rの要素γは$\sigma_\alpha\in CG(R)$によって，Rのどれかの要素γ_αに移されるだろう。さらにγ_αは$\sigma_\beta\in CG(R)$によって$\gamma_{\beta\alpha}$へと移されるだろう。以下続けていくと，再びγへもどってくる。このような移り合える仲間をひとまとめにすることによってRを分類したものを，Rの軌道という。

命題4 (3)　関係Rをn項関係とする。添字の被覆：

$$\{1,2,\cdots,n\}=I\cup J \quad I\cap J=K\neq\emptyset$$

を考え，

$$R_1=R\pi(I),$$

$$R_2=R\pi(J)$$

とおく。このとき次の関係が成立する。

$$CG(R)=CG(R_1,\ R_2) \quad //$$

証明：一般性を失うことなく，$I=\{1,\cdots,n_1\}$，$J=\{n_2,\cdots,n\}$，$1\leqq n_2\leqq n_1$とする。補題2の(3)により，$CG(R)\subseteq CG(R_1)$，$CG(R)\subseteq CG(R_2)$. よ

って，$CG(R) \subseteq CG(R_1, R_2)$。逆に$\sigma \in CG(R_1, R_2)$とすると，$\forall \gamma = (\gamma_1, \cdots, \gamma_n) \in R$に対して，$\gamma' = (\gamma_1, \cdots, \gamma_{n_1})$，$\gamma'' = (\gamma_{n_2}, \cdots, \gamma_n)$とおけば$\gamma' \in R_1$，$\gamma'' \in R_2$. よって$\sigma(\gamma') \in R_1$，$\sigma(\gamma'') \in R_2$。しかも$\sigma(\gamma')$と$\sigma(\gamma'')$の添字がKの要素の成分は一致している。これは$\sigma(\gamma) \in R$を意味する。すなわち$CG(R) \subseteq CG(R_1, R_2)$。　//

この命題によって Codd の正規化は等価変換であることがわかる。

〔例10〕例4のR_1を考える。

$$CG(R_1) \overset{\text{命題3}}{=} CG\left(\begin{pmatrix} a & d & a \\ a & e & a \end{pmatrix}, \begin{pmatrix} b & a & c \\ c & a & b \end{pmatrix} \right),$$

$$\overset{\text{命題2}}{=} CG\left(\begin{pmatrix} d & a \\ e & a \end{pmatrix}, \begin{pmatrix} a & c \\ a & b \end{pmatrix} \right),$$

$$\overset{\text{命題1}}{=} CG\left((a), \begin{pmatrix} d \\ e \end{pmatrix}, \begin{pmatrix} c \\ b \end{pmatrix} \right).$$

すなわち，

$$\text{データベース}R_1 = \begin{pmatrix} a & d & a \\ b & a & c \\ a & e & a \\ c & a & b \end{pmatrix}$$

とデータベース$\left\{ (a), \begin{pmatrix} d \\ e \end{pmatrix}, \begin{pmatrix} c \\ b \end{pmatrix} \right\}$とは同じ基本情報であるので，取出せる情報は同じであることがわかる。　//

例10の結論を考察しよう。直観的には単純なものの方が理解しやすい。だから分解された関係の集合の方が，そのデータベースが持つ情報を理解できる。従って，R_1のもつ情報を$\left\{(a) \begin{pmatrix} d \\ e \end{pmatrix}, \begin{pmatrix} c \\ b \end{pmatrix}\right\}$と表わすのは可能である。この情報表現が一意的かどうかはまだ検討はしていないが，一意的でないようだ。

相沢はもうひとつ興味ある結果を提出している[(3)]。これを例11に示そう。

〔例11〕$\{a, b, c\}$上の関係として次の二つを考える：

$$B_1 = \begin{pmatrix} a & c & a \\ c & c & b \end{pmatrix}, \quad B_2 = \begin{pmatrix} a & c & a \\ a & c & b \\ c & c & a \\ c & c & b \end{pmatrix}.$$

これらを (1,2) 項，(2,3) 章に射影すると，いずれからも

$$T_1 = \begin{pmatrix} a & c \\ c & c \end{pmatrix},\quad T_2 = \begin{pmatrix} c & a \\ c & b \end{pmatrix}$$

が得られる。二つの関係の結合を＊で表わすと，$B_2 = T_1 * T_2$であるが，$B_1 \neq T_1 * T_2$である。よって結合のわなを避けるようにするというのが Codd 流の正規化理論であるので，

分解：$B_2 \to \{T_1, T_2\}$は許されるが，

分解：$B_1 \to \{T_1, T_2\}$は許されない。

しかるに，B_1とB_2とは多少の考察によって本質的には同じ情報をもつように見える。

事実，$CG(B_1) = CG((a), (b), (c))$,

$CG(B_2) = CG((a), (b), (c))$,

$CG(T_1, T_2) = CG((a), (b), (c))$

である。これは直観とよく合う結果が得られる。

以上，二つの結果を考慮すれば，一般には

$$R = R_1 * R_2 \underset{\not\rightarrow}{\leftarrow} CG(R) = CG(R_1, R_2)$$

であるが，分解の条件はCodd 流よりもゆるめて，$CG(R) = CG(R_1, R_2)$の方が望ましいといえる。 //

6. 考察と結論

Codd 流の正規化理論のもくろみは，実世界での情報の発生，変更，消失を忠実に反映するような基本的な情報単位に細分すれば安定したデータベースを設計できること，さらにこれら基本情報へ単位化された情報を関連づけるのに各情報単位につけられた主キーを利用できることであった。例7では，基本情報単位は次の3つである：

部品関係表（主キーは部品番号），

会社関係表（主キーは会社名），

受注関係表（主キーは部品番号と会社名）。

ここで，受注関係の主キーは受注番号ではないことに着目することが肝要であ

る。受注番号を導入することは可能だが，それでは部品関係表，会社関係表との関連をつけられなくなる。主キーの成分として他の関係の主キーが参照される構造こそ，利用者が関連する情報への接近の際に手がかりを与えるものだから，この構造は明示されなければならない。

さて，Paredaens は一方の情報が他方の情報から導出できるかどうかを判定する方法を与えた。しかし，この方法は導出を証明できはするが，主キー概念を使わないので導出の具体的アルゴリズムを与えるものではない。この点は相沢の拡張でも同様である。随伴群の理論は情報の基本単位の考え方に大きく貢献した点は本文§3，§5で示されたように十分な評価ができる。しかしながら，データベース構築の技術としては情報の構造化の観点が欠けている点で総合的な手法とはなっていないことは残念なところである。

今後，主キーと随伴群の関係についての考察，随伴群が単位元だけのときと対称群となるときとの比較検討がされよう。現在のところ，群の構造からの考察は模索の段階にあり，あまり有意義な結果は導びけそうもない。