Mem. Fac. Sci. Eng. Shimane Univ.
Series A
33, pp. 73-88 (1999)

意味データモデルにおける集合値経路関数従属性

曽田　直樹[†]・黒岩　大史[††]・小林　康幸[††]

Set-Valued Path Functional Dependency on Semantic Data Model

Naoki SOTA[†], Daishi KUROIWA[††] and Yasuyuki KOBAYASHI[††]

Abstract

A data model based on semantic data model which can treat set-valued functions is introduced, and path functional dependency for such data model is investigated. Also an effective algorithm of the closure calculation in the data model is proposed and considered.

概　　　要

意味データモデルにおける経路を集合値写像として与えることを提唱し，その経路関数従属性について考察する．また，このモデルでの閉包計算を効率よく行うアルゴリズムを提案し，その有効性について検討する．

1. まえがき

現在，データベースの応用分野は文字や数値を中心とした古典的なデータ処理の分野からフル・テキスト，図形，イメージ，音声データなどを含む，いわゆるマルチメディアデータベースにまで広がってきた．それにともなって，最も汎用的であるとされていたリレーショナルモデルに基づくデータベースシステムでも，応用分野からの要求に応えられないことが多くなってきた．そこで，高度な応用分野に対応できるようなデータモデルとして研究されているのが意味データモデル，オブジェクト指向データモデルなどの様々なデータモデルである．意味データモデルは，データ表現の枠組みが自然で直観的で，データベースユーザがデータを扱うときの意味（meaning）がデータベースの構造に素直に反映できるという利点

† (株)出雲村田製作所，島根県
Izumo Murata Manufacturing Co., Ltd., 2308 Kaminaoe, Hikawa-cho, Hikawa-gun, Shimane, 690–0696 Japan
†† 島根大学総合理工学部，島根県
Faculty of Science and Engineering, Shimane Univ., 1060 Nishikawatsu-cho, Matsue-shi, Shimane, 690–8504 Japan

を持つものである[3].

G. E. Weddell[1]は，リレーショナルモデルにおける一貫性制約の中で重要な関数従属性を，意味データモデルに経路を用いることによって経路関数従属性として表現した．しかし，この経路は通常の単値写像で表されているため，実世界においては意味を自然に表現出来ない例も多数生じていた．

本論文では，意味データモデルにおける経路を集合値写像として与えることを提唱し，その経路関数従属性について考察する．また，[1]で提案された閉包計算アルゴリズム（スキーマに与えられた経路関数従属性から極小なものを導出するためのアルゴリズム）を元にし，より効率よく行うアルゴリズムを提案し，その有効性を検証する．

2. モデルの定義

2つの集合X, Yに対しfがXからYへの写像（単値写像）であるとは，fがXの各要素に対し，Yの1つの要素を対応させるような規則のことをいい，このとき$f: X\to Y$と表す．また$A\subset X$に対して，

$$f(A) := \{f(a) \mid a\in A\}$$

をAのfによる像という．

[1]では，経路として通常の写像（単値写像）が用いられていたが，例えば，後述の学生スキーマにおける趣味などにおいては，自然にその意味を表現することは出来なかった．そこで本論文では経路を集合値写像としてもよい意味データモデルを考察する．その準備として，集合値写像および本論文で用いる特別な集合値写像の像について定義する．

2.1 集合値写像

2つの集合X, Yに対しfがXからYへの集合値写像であるとは，fがXの各要素に対し，Yの巾集合（$P(Y)$ または2^Yと書く）の1つの要素を対応させるような規則のことをいう．（すなわち$f: X\to 2^Y$である．）$P(Y)$ は，集合Yの部分集合の全体がつくる集合のことである．

一般には集合値写像$f: X\to 2^Y$による$A\subset X$の像は次のように定義される．

$$f(A) := \bigcup_{a\in A} f(a)$$

しかし，上述した像の定義では第3.3節における推論則を満たさない為(注1)，本論文では以下の定義を採用することによって集合値写像においても従来の推論則を考慮できるようにした．

[定義1] 集合Xが与えられたとき，

(注1)：推論則（A5）を満たさない．

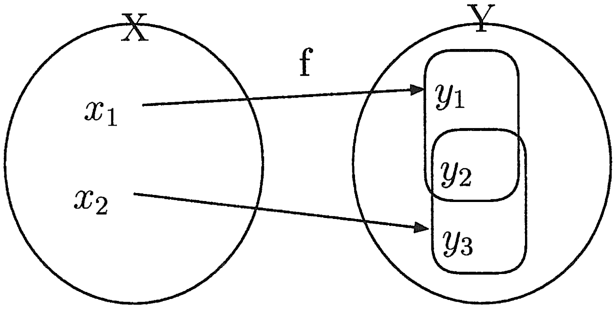

図1　集合値写像の像の例
Fig. 1.　Example of Image of Set-Valued Function

$$P^i(X) := \begin{cases} P(X), & i=1 \\ P(P^{i-1}(X)), & i \geqq 2 \end{cases}$$

［定義2］（集合値写像の像）
$f: X \to 2^Y$ について
$A \in P(X)$ のとき，

$$f(A) := \{f(a) \mid a \in A\}$$

$\alpha \in P^i(X)\,(i \geqq 2)$ のとき，

$$f(\alpha) := \{f(a) \mid a \in \alpha\}$$

［例1］
図1を例に説明する．
$f: X \to 2^Y$, $\{x_1, x_2\} \in P(X)$

$$f(\{x_1, x_2\}) = \{f(x_1), f(x_2)\} = \{\{y_1, y_2\}, \{y_2, y_3\}\}$$

2.2　スキーマと Path に関する定義

［定義3］（基本定義）
String：全ての文字列の集合
Integer：全ての数値の集合
Boolean：真偽値の集合

以下に，値として *String*, *Integer*, *Boolean* をもつ *AttClass* と，オブジェクト識別子を要素にもつ集合の集まりである *IdClass* を定義する．オブジェクト識別子とは，各オブジェクトを一意に識別するものである．

［定義 4］（*AttClass*）

$$Att\ Class := \{String, Integer, Boolean\}$$

［定義 5］（*IdClass*）

IdClass とは，オブジェクト識別子を要素にもつ集合を元とする集合族のことをいう．

［定義 6］（*Property*）

$C_1 \in IdClass$, $C_2 \in IdClass \cup AttClass$, $P : C_1 \to C_2$ または $P : C_1 \to 2^{C_2}$ であるとき，P を *Property* という．

クラススキーマを以下のように定義する．

［定義 7］（クラススキーマ（Class Schema））

- $CS := C\{P_1 : C_1, P_2 : C_2, \cdots, P_m : C_m, P_1' :: C_1', P_2' :: C_2', \cdots, P_n' :: C_n'\}$
  $(C \in Id\ Class,\ C_i \in Id\ Class \cup Att\ Class,\ P_i : C \to C_i,\ P_i' : C \to 2^{C_i})$
- $Class(CS) := \{C, C_1, \cdots, C_m, C_1', \cdots, C_n'\}$
- $Props(C) := \{P_1, P_2, \cdots, P_m, P_1', \cdots, P_n'\}$

または，

- $CS := C\{\quad\}(CC \in Att\ Class)$
- $Class(CS) := \{C\}$
- $Props(C) := \emptyset$

のとき CS を**クラススキーマ**と呼ぶ．

スキーマ S をクラススキーマの有限集合によって以下のように定義する．

［定義 8］（スキーマ（Schema））

スキーマ S $: = \{CS_1, CS_2, \cdots, CS_n\}$

$(CS_i$: クラススキーマ$)$

［定義 9］（*Classes*(*S*)）

スキーマ S が与えられたとき，

$$Class(S) := \bigcup_{CS \in S} Class(CS)$$

我々は，解釈と呼ぶデータベースインスタンスはラベル付き有向グラフの形式に一致していると仮定する．

［定義 10］ ラベル付き有向グラフ $G(V, A)$, $v \in V$ が与えられたとき，

$l(v)$: ＝頂点 v のクラスネームラベル

[1]において述べられた解釈に集合値写像を扱うように拡張した定義を以下に示す．

［定義 11］ ラベル付き有向グラフ $G(V, A)$ がスキーマ S の解釈であるとは以下の条件を満足するものである．但し，アークのうち単値写像は→，集合値写像は→→によって表現する．

(ⅰ)　$V \subset \{v \mid v \subset C,\ C \in Classes(S)\}$

(ⅱ)　$u \xrightarrow{p} v \in A$ または $u \xrightarrow{p}\!\!\rightarrow v \in A \Rightarrow l(u) \in Dom(p),\ l(v) = Ran(l(u), p)$

(ⅲ)　$u \xrightarrow{p} v,\ u \xrightarrow{p} w \in A \Rightarrow v = w$

(ⅳ)　$u \in V \Rightarrow \forall p \in Props(l(u)),\ \exists u \xrightarrow{p} v \in A$ または $\exists u \xrightarrow{p}\!\!\rightarrow v \in A$

表1におけるスキーマのある解釈は，図2のようになる．

［定義12］（Id）

スキーマ S, $C \in Classes(S)$ について，$Id: C \rightarrow C$ を恒等写像（$Id(c) = c,\ \forall c \in C$）とする．

［定義13］（$Path$）

スキーマ S について，

$$Path := \{P_1 \circ P_2 \circ \cdots \circ P_n \mid P_i \in Props(C_{i+1}),\ (P_i : C_{i+1} \rightarrow C_i \text{ または } P_i : C_{i+1} \rightarrow 2^{C_i})\}$$

表1　学生スキーマ
Table 1　Student Schema

学生 {*Hobby*：趣味，学生名：*String*，登録：住所録，両親：人間，祖父母：人間}
趣味 {趣味名：*String*, *Tool*：道具}
住所録 {住所：*String*, *TEL*：*String*}
人間 {名前：*String*}
道具 {道具名：*String*}
String { }

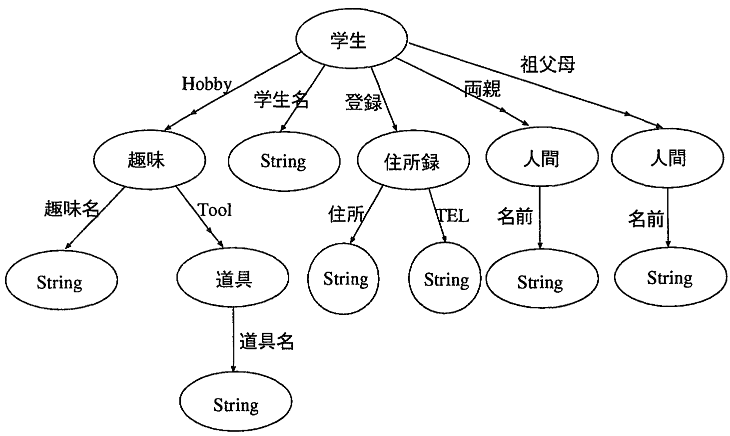

図2　学生スキーマの解釈
Fig. 2.　An interpretation of Student Schema

また，*Path* のそれぞれの元を *path function* という．

［定義 14］（*len*）

スキーマ S について，$p \in Path$ が与えられたとき，

$$len(p) := \begin{cases} 0, & p = Id \text{ のとき} \\ 1 + len(p''), & p = p' \circ p'' \text{ かつ } p' : Property \text{ のとき} \end{cases}$$

［定義 15］（*Dom*）

スキーマ S について，$p \in Path$ のとき，

$$Dom(p) := \begin{cases} Classes(S), & p = Id \text{ のとき} \\ \{C\}, & p_1, \cdots, p_n : Property, p = p_1 \circ \cdots \circ p_n, \\ & (p_n : C \to C' \text{ または } p_n : C \to 2^{C'}) \text{ のとき} \end{cases}$$

［定義 16］（*Path*(*C*)）

スキーマ S について，$C \in Classes(S)$ が与えられたとき，

$$Path(C) := \{p \in Path \mid C \in Dom(p)\}$$

［定義 17］（*Ran*）

スキーマ S について，$C \in Classes(S), p \in Path(C)$ のとき，

$$Ran(C, p) := \begin{cases} C, & p = Id \text{ のとき} \\ C', & p \in Props(C), \\ & (p : C \to C' \text{ または } p : C \to 2^{C'}) \text{ のとき} \\ Ran(Ran(C, p'), p''), & p' \in Props(C), p = p'' \circ p' \text{ のとき} \end{cases}$$

［定義 18］（*path function* の合成）

スキーマ S, $C \in Classes(S), p_1 \in Path(C), p_2 \in Path(Ran(C, p_1))$ について，

$$p_2 \circ p_1 := \begin{cases} p_2, & p_1 = Id \text{ のとき} \\ p_1, & p_2 = Id \text{ のとき} \\ p_2, \circ p_1 & \text{その他} \end{cases}$$

［定義 19］

スキーマ S, $C \in Classes(S)$, $x \in Path(C)$, $C' \in Classes(S)$, $C' = Ran(C, x)$, $Y \subset Path(C')$ について，

$$Y \circ x := \{y \circ x \mid y \in Y\}$$

3. 集合値経路関数従属性

本章では，集合値写像を取り扱える経路に関する関数従属性について述べる．

3.1 定　　義

経路関数従属性（PFD）と key PFD の定義をする．

［定議 20］（経路関数従属性（PFD））

スキーマ S, $C \in Classes(S)$, X, $Y \subset Path(C)$ について，**経路関数従属性（PFD）**$C(X \to Y)$ が存在するとは，次の条件が成立するときをいう．

$c, c' \in C$ について，

$$\forall x \in X, x(c) = x(c') \Rightarrow \forall y \in Y, y(c) = y(c')$$

［定議 21］（key PFD）

スキーマ S, $C \in Classes(S)$, X, $Y \subset Path(C)$, PFD $C(X \to Y)$ について $Y = \{Id\}$ であるとき，その PFD を **key PFD** という．

PFD $C(X \to Y)$ について，$Y \subset X$ となるとき，**自明である**という．

3.2 経路関数従属性の論理的含意

本節では，最初に経路関数従属性の論理的含意を定義する．次に，スキーマ S における経路関数従属性の有限集合 F によって論理的に含意される経路関数従属性の集合 F^+ を定義する．

［定議 22］（論理的含意）

スキーマ S, F: スキーマ S における PFD の集合が与えられたとき，$C \in Classes(S)$, X, $Y \subset Path(C)$ について，

$$F \vDash C(X \to Y) := F$$

中の *PFD* を満足する任意の解釈において $C(X \to Y)$ を満足する．

$F \mid = C(X \to Y)$ のとき，F は論理的に $C(X \to Y)$ を含意するという．

［定議 23］（F^+）

スキーマ S, F: スキーマ S における PFD の集合が与えられたとき，

$$F^+ := \{C(X \to Y) \mid F \vDash C(X \to Y)\}$$

F^+ は非常に大事な意味をもっており，これはスキーマ S が満たさなければならない PFD という一貫性制約の全てを表している．

3.3 経路関数従属性に関する推論則

経路関数従属性間の論理的含意を一般的に理解するためには，F から F^+ を計算する必要がある．あるいは，少なくとも F と $C(X \to Y)$ が与えられたとき，$C(X \to Y)$ が F^+ 中に存在するか否かを決定できなければならない．そのためには，1つあるいはそれ以上の従属性が他の従属性をいかに含意しているかを述べる推論則が必要となる．以下に，その推論則を定義する．

表2 学生スキーマにおける経路関数従属性
Table 2 Path Functional Dependencies for Student Schema

学生（{学生名，登録}→{両親}）
学生（{学生名，両親}→{*Id*}）
学生（{両親}→{祖父母}）
趣味（{趣味名}→{*Id*}）
住所録（{*TEL*}→{*Id*}）

［定義24］（PFD 推論則）
スキーマ S, F：スキーマ S における PFD の集合が与えられたとき，

(A1) (reflexivity)
$C \in Classes(S), X \subset Path(C),\ Y \subset X \Rightarrow C(X \rightarrow Y)$

(A2) (path function argumentation)
$C \in Classes(S), X, Y, Z \subset Path(C), C(X \rightarrow Y)$ かつ $C(Y \rightarrow Z) \Rightarrow C(X \rightarrow Z)$

(A3) (transitivity)
$C \in Classes(S), X, Y, Z \subset Path(C), C(X \rightarrow Y) \Rightarrow C(X \cup Z \rightarrow Y \cup Z)$

(A4) (simple attribution)
$C \in Classes(S), x \in Props(C) \Rightarrow C(\{Id\} \rightarrow \{x\})$

(A5) (simple prefix argumentation)
$(C_1 \in Classes(S), X, Y \subset Path(C_1), C_1(X \rightarrow Y))$ かつ $(C_2 \in Classes(S)$,
$\exists z \in Props(C_2), C_1 = Ran(C_2, z)) \Rightarrow C_2(X \circ z \rightarrow Y \circ z)$

［定義25］（F^{rule}）
スキーマ S, F：スキーマ S における PFD の集合が与えられたとき，

$$F^{rule} := \{f | f \text{ は } F \text{ から推論則 } (A1) \sim (A5) \text{ を使って導出できる}\}$$

以下に，経路関数従属性の導出過程を学生スキーマによって説明する．表1のスキーマと表2の経路関数従属性に推論則を適用して，学生（{学生名，*TEL o* 登録} → {*Hobby*}）が導出できるかどうかの推論過程を表3に示す．

しかし，定義24で与えた推論則を用いても，$C(X \rightarrow Y)$ が導出できるかどうかを決定するのは難しい．簡単なものであれば，推論則を用いて少ないステップで $C(X \rightarrow Y)$ を導出できるかどうか決定することはできるかもしれない．一般に，スキーマや経路関数従属性が複雑なものになってくると，いくら推論則を用いると導出できるものであっても最適なステップで決定できるとは限らない．まして導出できないものであった場合，どの時点で決定できないかを判断することも難しい．

そこで次章では，より効率よく導出できるかどうかを決定する方法について述べる．

表3 推論過程
Table 3 Reasoning Process

	PFD	正当性
1.	住所録 $(\{TEL\}\to\{Id\})$	与えられたもの
2.	学生 $(\{TEL$, 登録$\}\to\{$登録$\})$	1.と A5
3.	学生 $(\{$学生名, TEL, 登録$\}\to\{$学生名, 登録$\})$	2.と A2
4.	学生 $(\{$学生名, 登録$\}\to\{$両親$\})$	与えられたもの
5.	学生 $(\{$学生名, 登録$\}\to\{$学生名, 両親$\})$	4.と A2
6.	学生 $(\{$学生名, 両親$\}\to\{Id\})$	与えられたもの
7.	学生 $(\{$学生名, 登録$\}\to\{Id\})$	5.と6.と A3
8.	学生 $(\{Id\}\to\{Hobby\})$	A4
9.	学生 $(\{$学生名, 登録$\}\to\{Hobby\})$	7.と8.と A3
10.	学生 $(\{$学生名, TEL, 登録$\}\to\{Hobby\})$	3.と9.とA3

4. 閉 包

本章では，最初に閉包の定義を行なう．次に，基本的な閉包計算アルゴリズムについて述べる．次に，[1]において提案された効率よく閉包を計算するアルゴリズムについて述べる，最後に，我々の提案する閉包計算アルゴリズムについて述べる．

4.1 閉包の定義

［定義 26］ (X^+)

スキーマSについて，F: スキーマSにおける PFD の集合，$C\in Classes(S)$, $X\subset Path(C)$ が与えられたとき，

$$X^+ := \{p\in Y \mid C(X\to Y)\in F^{rule}\}$$

またX^+のことを，Xの閉包という．

［補題 1］ $(F^+=F^{rule})$

スキーマ S, F: スキーマSにおける PFD の集合が与えられたとき，

$$F^+=F^{rule}$$

が成立する．

【証明】

付録の補題 2, 3 より比較的容易に示される． □

補題 1 は，スキーマSと陽に宣言された PFD の集合Fから，導出可能な PFD は，Fに推論則 (A1), (A2), (A3), (A4), (A5) を可能な限り適用してゆけば出るということを言っているのである．我々は，スキーマSにおける PFD の集合FからF^+を計算することは一般的に時間がかかることがわかっている．これは仮にF自身が小さくてもF^+の集合は大

きくなりうるからである．

まず，[1]の閉包計算アルゴリズムを紹介する．

4.2　閉包計算アルゴリズム

本節では，X の閉包を求める際に F^+ を使用すると時間がかかるために，新たな PFD の集合 $F_1(C), F_2(C)$ を定義し，その集合を使用することによって，計算を効率化した閉包計算アルゴリズムを定義する．最初に，${X_1}^i$ の計算に必要な F^+ の部分集合 $F_1(C), F_2(C)$ を定義する．

［定義 27］（$F_1(C)$）

スキーマ S について，$C \in Classes(S)$ が与えられたとき，

$$F_1(C) := \{C(\{x\} \to \{y \circ x\}) \mid x, y \circ x \in Path(C), len(y) = 1\}$$

［定義 28］（$F_2(C)$）

スキーマ S について，F：スキーマ S における PFD の集合，$C \in Classes(S)$ が与えられたとき，

$$F_2(C) := \{C(X \circ z \to Y \circ z) \mid z \in Path(C), C' = Ran(C, z), C'(X \to Y) \in F\}$$

次に，$F_1(C), F_2(C)$ を使用した X^i の定義をする．

［定義 29］（X^i）

スキーマ S について，F：スキーマ S における PFD の集合，$C \in Classes(S), X \in Path(C)$ が与えられたとき，以下のアルゴリズムによって X^i を計算する．

step 1.　$i \leftarrow 0; X^0 \leftarrow X$

step 2.　$X^{i+1} \leftarrow X^i \cup \{x \in Path(C) \mid C(Y \to Z) \in F_1(C) \cup F_2(C),\ Y \subset X^i, x \in Z\}$

step 3.　$X^{i+1} \neq X^i$ のとき，$i \leftarrow i+1$；繰り返し step 2. を行なう．
　　　　$X^{i+1} = X^i$ のとき，終了する．

$F_1(C), F_2(C)$ から推論則（**A1**）～（**A3**）を使用して導出した PFD の集合は，明らかに F^+ よりも濃度が小さい集合なので F^+ より X の閉包を求める際より比較の回数は減少する．

次節では，[1]において効率化された閉包計算アルゴリズムについて述べる．

4.3　閉包計算アルゴリズム 1

本節では，第4.1節で定義した $F_1(C), F_2(C)$ について *pathfunction* の長さを制限した $F_1(C, i)$, $F_2(C, i)$ を定義する．その集合を使用することによって，$F_1(C), F_2(C)$ との比較回数を減少させる閉包計算アルゴリズム 1 を定義する．

最初に，$Path(C, i)$, $F_1(C, i)$, $F_2(C, i)$ を定義する．

［定義 30］（$Path(C, i)$

スキーマ S について，$C \in Classes(S), i \geqq 0$ が与えられたとき，

$$Path(C, i) := \{p \in Path(C) \mid len(p) \leqq i\}$$

［定義 31］ $(F_1(C, i)$
スキーマ S について，$C \in Classes(S)$, $i \geqq 0$ が与えられたとき，

$$F_1(C, i) := \{C(\{x\} \to \{y\}) \in F_1(C) \mid len(y) \leqq i\}$$

［定義 32］ $(F_2(C, i)$
スキーマ S について，F：スキーマ S における PFD の集合 $C \in Classes(S)$, $i \geqq 0$ が与えられたとき，

$$F_2(C, i) := \{C(X \to Y) \in F_2(C) \mid \forall p \in X \cup Y, len(p) \leqq i\}$$

上述した集合を用いた閉包計算アルゴリズム 1 を定義する．
［定義 33］ $(X_1{}^i)$
スキーマ S について，F：スキーマ S における PFD の集合，$C \in Classes(S)$, $X \subset Path(C)$ が与えられたとき，

$$\alpha := \max_{C'(A \to B) \in F} (\max_{p \in A \cup B} len(p) - \min_{p \in A \cup B} len(p))$$

とし，以下のアルゴリズムによって $X_1{}^i$ を計算する．

step 1. $i \leftarrow 0; X_1{}^0 \leftarrow X$
step 2. $j \leftarrow \max(1, \alpha) + \max_{p \in X_1{}^i} len(p)$
$X_1^{i+1} \leftarrow X_1{}^i \cup \{x \in Path(C, j) \mid C(Y \to Z) \in F_1(C, j) \cup F_2(C, j),\ Y \subset X_1{}^i, x \in Z\}$
step 3. $X_1^{i+1} \neq X_1{}^i$ のとき，$i \leftarrow i+1$；繰り返し step 2. 行なう．
$X_1^{i+1} = X_1{}^i$ のとき，終了する．

4.4 閉包計算アルゴリズム 2

本節では，第4.2節で定義した $F_1(C)$, $F_2(C)$ について，新たに *path function* の長さを制限した $F_1(C:i)$, $F_2(C:i)$ を定義する．次に，我々の提案する閉包計算アルゴリズムの基本となる概念について述べる．次に，それを考慮した X の閉包計算のアルゴリズム 2 を定義する．最後に閉包計算アルゴリズム 2 によって正しく X^+ を導出することを証明する．
最初に，$F_1(C:i)$, $F_2(C:i)$ の定義をする．
［定義 34］ $(F_1(C:i)$
スキーマ S について，$C \in Classes(S)$, $i \geqq 0$ が与えられたとき，

$$F_1(C:i) := \{C(\{x\} \to \{y\}) \in F_1(C) \mid len(x) = i\}$$

［定義 35］ $(F_2(C:i))$
スキーマ S について，F：スキーマ S における PFD の集合 $C \in Classes(S)$, $i \geqq 0$ が与えられ

たとき，

$$F_2(C:i):=\{C(X\to Y)\in F_2(C)\mid \min_{p\in X} len(p)=i\}$$

我々が提案するアルゴリズムは，以下の概念によるものである．

(1) ある X^i-X^{i-1} の元 x について，比較する $F_1(C)$ の元 $C(\{x'\}\to\{y\})$ は $len(x)=len(x')$ のものだけにする．

(2) 比較する $F_2(C)$ の元を第4.3節では，長さについて上側からだけ制限していたが，下側からも制限してより強い制限にする．

(3) あるステップで $F_1(C), F_2(C)$ と比較したとき，マッチしたものがあれば，それ以降のステップにおいてそのマッチした元は比較対象から削除する．

上述した概念を使用した閉包計算アルゴリズム2を以下に定義する．

［定義36］ (X_2^i)

スキーマ S について，F: スキーマ S における PFD の集合，$C\in Classes(S), X\subset Path(C)$ が与えられたとき，以下のアルゴリズムによって，X_2^i を計算する．

step 1. $i\leftarrow 0;\ X_2^{-1}\leftarrow\emptyset;\ X_2^0\leftarrow X;\ Use^0\leftarrow\emptyset$

step 2. $\alpha\leftarrow\min_{p\in X_2^i} len(p);\ \beta\leftarrow\max_{p\in X_2^i} len(p)$

$New^{i+1}(F_1)\leftarrow\{C(\{y\}\to\{z\})\in F_1(C:j)-Use^i\mid y\in X_2^i-X_2^{i-1},\ len(y)=j\}$

$New^{i+1}(F_2)\leftarrow\{C(Y\to Z)\in \bigcup_{j=\alpha}^{\beta} F_2(C:j)-Use^i\mid Y\subset X_2^i\}$

$X_2^{i+1}\leftarrow X_2^i\cup\{x\in Z\mid C(Y\to Z)\in New^{i+1}(F_1)\cup New^{i+1}(F_2)\}$

$Use^{i+1}\leftarrow Use^i\cup New^{i+1}(F_1)\cup New^{i+1}(F_2)$

step 3. $X_2^{i+1}\neq X_2^i$ のとき，$i\leftarrow i+1$；繰り返し step 2. を行なう．

$X_2^{i+1}=X_2^i$ のとき，終了する．

最後に，X_2^i が X^+ を正しく計算することを証明する．

［定理1］

スキーマ S について，F: スキーマ S における PFD の集合，$C\in Classes(S), X\subset Path(C)$ が与えられたとき，

$$X^+=\bigcup_{i\geqq 0} X_2^i$$

が成立する．

【証明】

付録の補題4, 5より比較的容易に示される．

一般的な場合について，閉包計算における比較回数を計算し，その比較回数について評価を行う．

以下の設定条件(注2)のもとで X_1^i, X_2^i での閉包の導出における比較回数を求め，その比較

を行う．

1. スキーマは完全二分木（深さ l）
2. $|F_2(C)|=m$
3. $|X|=s$
4. $\forall C(A \to B) \in F_2(C), A, B$: 1 点集合
5. $\forall C(\{a\} \to \{b\}) \in F_2(C)$, $len(a) \leq n$, $len(b)=l$
6. $\forall p \in X$, $len(p)=n+1$

上記の条件のもとでの閉包計算アルゴリズム 1 における全比較回数は次のようになる．

$$s \times \sum_{i=0}^{l-n-1} (2^{i+1}-1) \times \{2(2^l-1)+m\}$$

閉包計算アルゴリズム 2 における全比較回数は次のようになる．

$$s \times \sum_{i=0}^{l-n-2} 2^i \times 2^{n+i+2}$$

$l=10$ とし，$A: m=1, B: m=2^n, C: m=2^{n+1}-1$ の 3 パターンを考え，n を動かして計算した結果を図 3 示す．

図 3 より我々のアルゴリズムはこの場合，約 1/10 の比較回数で済むことが分かる．我々のアルゴリズムの最大の特徴は $n=9$ のとき比較を 1 回も行わない点にある．閉包計算アルゴリズム 1 はそのとき $m=1$ であっても $s \times 2046$ 回も比較を行う．

5. むすび

本論文では，意味データモデルにおいて集合値写像を扱ったした経路関数従属性に拡張を行った．集合値写像を扱った経路関数従属性に拡張することによって，データベースユーザが実世界の意味をデータベースの構造に素直に反映できる意味データモデルにおいて経路関数従属性を考えることが可能となった．

また，従来の閉包計算アルゴリズムよりも効率よく計算する閉包計算アルゴリズムを提案した．そのアルゴリズムによって，与えられた経路関数従属性が決定できるかどうかを速く計算でき，また，スキーマに与える経路関数従属性の極小（最低限必要な経路関数従属性，冗長な経路関数従属性を省いたもの）なものを速く導出することが可能となった．

文献

[1] G. E. Weddell, "Reasoning about Functional Dependencies Generalized for Semantic Data Models," ACM Trans. Database Syst., vol. 17, no 1, pp. 32–64, Mar. 1992.

[2] J. D. Ullman, "Principles of DATABASE SYSTEMS," COMPUTER SCIENCE PRESS, 1982.

（注 2）：閉包計算アルゴリズム 2 の U_{se} の利点が発揮されない例であることに注意されたい．

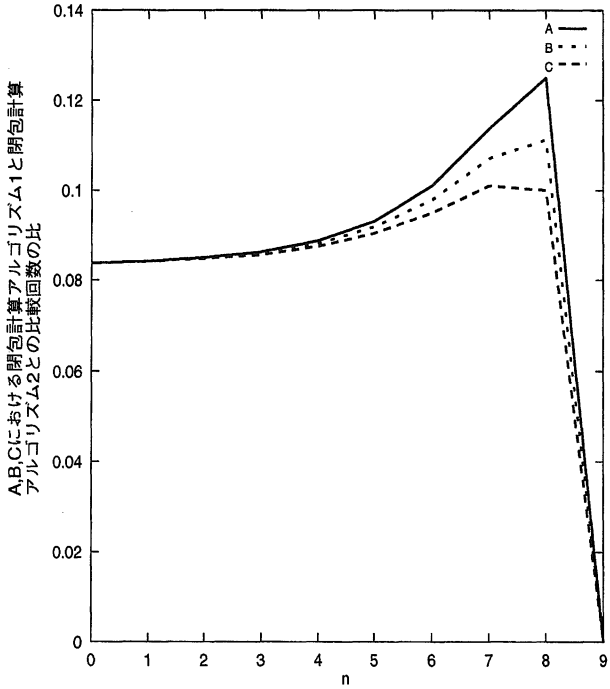

図3 計算結果
Fig. 3. Calculation Result

[3] 有澤 博, "意味データモデルにおける最近の動向," 情報処理, vol. 32, no. 9, pp. 1023–1031, Sep. 1991.
[4] 田中克己, "オブジェクト指向データベースの基礎概念," 情報処理, vol. 32, no. 5, pp. 500–513, May. 1991.
[5] 三浦孝夫, "データモデルとデータベースⅠ," サイエンス社, 1997.
[6] 三浦孝夫, "データモデルとデータベースⅡ," サイエンス社, 1997.

付 録

[補題2]
スキーマ S について, F: スキーマ S における PFD の集合が与えられたとき,

$$F^{rule} \subset F^{+}$$

が成立する.

【証明】
$f \in F^{rule}$ とすると, f は, F または定義24の推論則 (A1)〜(A5) によって導出されたもので

ある．$f \in F$ のときは明らかに成立する．(A1) によって導出されたとすると，$\exists C \in Classes(S)$, $\exists X \subset Path(C)$, $\exists Y \subset X$, $f = C(X \to Y)$．そこで，s, $t \in C$, $\forall x \in X$, $x(s) = x(t)$ とすると，明らかに $\forall y \in Y$, $y(s) = y(t)$ となる．従って，$f \in F^+$ が成立する．(A2)～(A5) については同様に証明できる．□

［補題 3］

スキーマ S について，F: スキーマ S における PFD の集合が与えられたとき，

$$F^+ \subset F^{rule}$$

が成立する．

【証明】

$f = C(X \to Y) \notin F^{rule}$ とする．

ここで次の解釈について考える．

$$
\begin{aligned}
&C = \{c_1, c_2\}\\
&\forall p \in X^+, p(c_1) = p(c_2)\\
&\forall p \in Path(C) - X^+, p(c_1) \neq p(c_2)\\
&\forall C' \in Classes(S) - \{C\}\\
&|C'| = 2\\
&\exists K \subset Path(C')\\
&s, t \in , s \neq t\\
&\forall p \in K^+, p(s) = p(t)\\
&\forall p \in Path(C') - K^+, p(s) \neq p(t)
\end{aligned}
$$

最初に F 中のすべての経路関数従属性がこの解釈において満たされることを示す．F の元 $C''(V \to W)$ が，この解釈において満たされないとする．

(ア) C'' が C のとき，

$V \not\subset X^+$ ならば常にこの解釈において満足される．また，$V \subset X^+$, $W \subset X^+$ ならば c_1, c_2 について $V \cup W$ のどんな関数に対しても値が等しくなるので常にこの解釈において満足される．従って，$V \subset X^+$, $W \not\subset X^+$ である．$C(V \to W) \in F$, $V \subset X^+$，定義26より，$W \subset X^+$ となる．これは，$W \not\subset X^+$ に矛盾する．

(イ) その他のとき，(ア) と同様に矛盾を導き出す．

(ア)，(イ) より，F の元 $C''(V \to W)$ は，この解釈において満足される．

次に $C(X \to Y)$ が，この解釈において満足されないことを示す．$C(X \to Y)$ が，この解釈において満足されるとすると，$X \subset X^+$ なので常に c_1, c_2 に対して値は等しい．したがって，この解釈において満足されるためには Y 中のどんな関数においても c_1, c_2 に対する値は等しくなければならない．従って，$X \subset X^+$, $Y \subset X^+$ となる．$Y \subset X^+$，定義26より，$f = C(X \to Y) \in F^{rule}$ となる．従って，$f = C(X \to Y) \notin F^{rule}$ に矛盾する．従って，$f = C(X \to Y)$ は，この解釈において満足されない．□

［補題 4］ $(X_2{}^i \subset X^i)$
スキーマSについて，F: スキーマ S における PFD の集合，$C \in Classes(S)$, $X \in Path(C)$ が与えられたとき，

$$\forall i \geqq 0, \quad X_2{}^i \subset X^i$$

が成立する．

【証明】
i に関する数学的帰納法によって証明する．$i=0$ のとき，$X_2{}^0 = X^0 = X$ より明らかに成立する．$i=k$ のとき，$X_2{}^k \subset X^k$ が成立すると仮定する．$i=k+1$ のとき，$t \in X_2^{k+1}$ とすると，定義36より $t \in X_2{}^k \cup \{x \in Z \mid C(Y \to Z) \in New^{k+1}(F_1) \cup New^{k+1}(F_2)\}$ となる．$t \in X_2{}^k$ のときは，帰納法の仮定と定義26より明らかに成立する．$t \in \{x \in Z \mid C(Y \to Z) \in New^{k+1}(F_1) \cup New^{k+1}(F_2)\}$ のときは，定義34, 35, 36より $t \in \{x \in Path(C) \mid C(Y \to Z) \in F_1(C) \cup F_2(C),\ Y \subset X^k,\ x \in Z\}$ となる．定義26より $t \in^{k+1}$ となり成立する． □

［補題 5］ $(X_2{}^i \supset X^i)$
スキーマSについて，F: スキーマ S における PFD の集合，$C \in Classes(S)$, $X \in Path(C)$ が与えられたとき，

$$\forall i \geqq 0, \quad X_2{}^i \supset X^i$$

が成立する．

【証明】
i に関する数学的帰納法によって証明する．$i=0$ のとき，$X_2{}^0 = X^0 = X$ より明らかに成立する．$i=k$ のとき，$X_2{}^k \supset X^k$ が成立すると仮定する．$i=k+1$ のとき，$t \in X^{k+1}$ とすると，定義29より $t \in X^k \cup \{x \in Path(C) \mid C(Y \to Z) \in F_1(C) \cup F_2(C),\ Y \subset X^k, x \in Z\}$ となる．従って，$t \in X^k$, $t \in \{x \in Path(C) \mid C(Y \to Z) \in F_1(C),\ Y \subset X^k,\ x \in Z\}$, $t \in \{x \in Path(C) \mid C(Y \to Z) \in F_2(C),\ Y \subset X^k,\ x \in Z\}$ の3つの場合にわけて考える．$t \in X^k$ のときは，帰納法の仮定と定義36より明らかに成立する．$t \in \{x \in Path(C) \mid C(Y \to Z) \in F_1(C),\ Y \subset X^k,\ x \in Z\}$ のとき，$\exists C(Y \to Z) \in F_1(C)$, $Y \subset X^k$, $t \in Z$, $t \in Path(C)$ となる．定義27，定義34より，$\exists j \geqq 0$, $\exists y, z \in Path(C)$, $Y = \{y\}$, $Z = \{z\}$, $j = len(y)$, $C(\{y\} \to \{z\}) \in F_1(C:j)$．ここで，$C(\{y\} \to \{z\}) \in F_1(C:j) - U_{se}{}^k$ または $C(\{y\} \to \{z\}) \in F_1(C:j) \cap U_{se}{}^k$ なので2つの場合にわけて考える．
（ア） $C(\{y\} \to \{z\}) \in F_1(C:j) - U_{se}{}^k$ のとき．$C(\{y\} \to \{z\}) \notin U_{se}{}^k$ なので，$y \notin X_2^{k-1}$ なる．従って，$y \in X_2{}^k - X_2^{k-1}$ となるため，$C(Y \to Z) \in New^{k+1}$ となる．従って，$t \in X_2^{k+1}$ が成立する．
（イ） $C(\{y\} \to \{z\}) \in F_1(C:j) \cap U_{se}{}^k$ のとき．$C(\{y\} \to \{z\}) \in U_{se}{}^k$ なので，定義36より，明らかに $t \in X_2^{k+1}$ が成立する．
$t \in \{x \in Path(C) \mid C(Y \to Z) \in F_2(C),\ Y \subset X^k,\ x \in Z\}$ のときも同様に証明できる． □