

データベース視覚操作の完全性

三浦孝夫
産能大学

神奈川県伊勢原市上粕屋 1573

盛屋邦彦

日立ソフトウェアエンジニアリング

横浜市中区尾上町 6-81

Abstract

本稿では図形操作を指向したデータ処理モデルを提案する。視覚操作では、スキーマに対応する図形(アイコン)を用いてデータベースへの問い合わせを行うことができる。次に、論理操作を定義し、両者が完全な対応関係を有すること(操作系の完全性)から、提案する操作系の表現力を特長付けることができる。

On the Completeness of Visual Operations for Databases

Takao MIURA† and Kunihiko MORIYA‡

† SANNŌ College Kamikasuya 1573, Isehara, Kanagawa, JAPAN

‡ HITACHI Software Engineering Onoe-Cho 6-81, Naka-KU, Yokohama, Kanagawa, JAPAN

In this paper, we propose Data Processing model for visual operations. These operations provide us with Query processing capabilities based on Iconic pictures that mean schematic information. Moreover, after introducing Logic based operations, we show these two languages have the equivalent expressive power.

1 まえがき

これまで大型計算機を中心とするデータベース処理では、情報の共通化や共有化のもと、不特定多数の利用者によって任意のタイミングで用いられていた。特に、住所録や電話番号簿などの個人情報格納されることは利用目的やコストから考えてありえなかった。しかし近年、ハードウェア処理能力の発展と、パーソナルコンピュータの普及により、個人環境でのデータベースの要望を生んでいる。専用目的の個人使用という、これまでとは逆の狙いをもつデータベースが必要となっているのである。

必ずしも専門的な知識の無い人が広く計算機を利用するため、データベース操作についても何等かの対応が必要となる。その中でも特に重要なものは、ユーザインタフェースの向上である。例えば、これまでの構文中心の処理記述に代わって、視覚化技術などの模索が重要であろう。もうひとつの重要な点は、操作言語それ自身である。これまで、データベース操作には大きく2種類の操作、すなわちアドホック操作と巡航操作が存在している。アドホック操作は、対話的に条件を与えてデータを検索する非定型操作であり、巡航操作は現在得られている情報から新たな検索を行う。伝統的に、後者を用いる処理は定型操作が多い。しかし両者は矛盾するものではない。例えば、非定型処理であっても操作内容が事前に決定出来ないとき、データベースの情報を巡航する必要がある。

これらの状況を踏まえ、本稿ではユーザインタフェースとして視覚操作を配慮した基本命令を提案し、これらを用いてデータ操作系を構築する。更に、第一階述語論理に基づく定式化を行い表現力を特長付ける。第2章では問題の整理と背景説明を行う。第3章では、本稿で扱う情報のモデル化技法とデータ操作(PIM代数)を非形式的に述べ、本稿のねらいを明らかにする。第4章、第5章ではそれぞれ代数操作、論理操作を導入し、両者が完全に対応付けられることをいう。

2 背景

本稿で想定する情報の利用については、共通していくつかの特性がある [A87]。まず対象データ量がきわめて少ない。例えば50KBの文字情報は画面で25ページにもなり、個人の理解できる範囲ではない [KM84]。第2に、定型的な利用が無く、どのようなデータの使われ方があるのか、利用者自身を含めて予測がつかない。第3には、スキーマが予め確定しない点である。むしろ利用の都度スキーマが変更され、ビューとスキーマの区別が意識されない。最後に、データベースの果たすべき信頼性保証(障害回復)や整合性保証については、洗練された機能よりも丈夫さが望まれる。

これらを大型計算機で実現するのはやっかいである。例えば、SQLを用いた関係データベースでは、文法、関係スキーマの意味、属性名を理解する必要がある。SQL非定型操作支援環境では、対話での絞り込みは「検索条件」の量に対応するが、巡航性に関する考察はない。普遍関係データベースは、属性名集合から結合演算で処理の意味を定めるものであるが、一意名仮定と極大オブジェクトの認識が必要である [U89]。

現在のワークステーション上でのデータベースシステムは、大型機環境が移動してきたものが多い。パーソナル計算機等のデータベースは関係モデルに基づくものが多いが、関係演算を完全には実現せず、整合性保証メカニズムも限定的である [A87]。

この方面では、関係モデルには基づかない研究もなされている。PDM(Personal Data Manager)は、意味データモデルに基づくブラウザ指向のシステムである [LM84]。質問は、作業型(カレントな主体型)を操作することで順に答えを得て行く。しかし、このモデルでは全ては主体であって、特にn項関連を人工的に表現せねばならない。

関係モデルの上にトークン(名詞)方式の質問を許す提案がなされている [Mo86]。列挙された各トークンを辞書を介して属性名へ変換する。この後、グラフで表されたデータベーススキーマ上で、与えられた属性名の構成木を見つけ、関係質問へと展開する。しかし、この興味ある方法では、得られた質問は一意ではなく利用者との対話が必要である。変換過程で発生する曖昧さは、たとえスキーマを知っていても避けられないことがあり、慣れてくるにつれて曖昧さがなくなるとは限らない。

オブジェクト指向データベースの提案では、操作言語として確立されたものはない。むしろ、C++などのプログラム言語を介してポインタを追跡するという手続き処理を意識しているように思える [CKW89]。我々は、データベース操作が宣言的な性質を備えるべきであると主張したい。このメカニズムの実現のために、操作言語は集合操作に基づくものでなければならない。

トークン方式で生じる曖昧さを、データ従属性や動的メッセージパッシング方式による継承情報から解消しようとする提案がある [NS88, HT89]。もちろん、この方式がデータの意味を正しく反映するか、データベースの実現手法と効率よいものであるかどうかがかれからの問題である。

ブラウザ方式の操作言語では、視覚情報をデータベースのユーザインタフェースとすることで、構文的な煩わしさを避けることができる [La86]。しかし、操作のための論理基盤がなく、思い付くままに実現される結果、統一性・有用性は経験に頼らざるを得ない。何をどこまで実現すれば良いかの基準が無いのである。現実に提案されているものでは、データベーススキーマがブラウザ対象となることが多く、視覚操作のほとんどは検索のアクセス経路(リンク)の設定に費やされる。このため、既存の経路の追跡とその途中での選択条件の設定という、きわめて単純な能力しか有さない。

これと類似したアプローチでは、ハイパーテキストが知られている [Go87]。ここでは情報はノードとリンクで表され、このリンクを追跡(ブラウズ)する操作が提供される。しかしこのモデルでは、抽象化機能が欠落している。型付けや汎化、複合オブジェクトを記述し操作する方法は、利用者の認識以外にないため質問の定式化が難しい。GraphLogは、本稿の提案に最も近い。演繹データベース上の推移閉包をパスの探索問題に変換し、ホーン節による特徴付けを行っている [CMW87]。しかし、ここでは複合オブジェクトに関する考察がなく、またハイパーテキストモデルと同様にデータ抽象化を十分には活用していない。

本稿で提案するPIM代数(視覚操作)は、スキーマ上で与えられる視覚操作であるが、他とは異なって複合オブジェクトを直接扱う事ができる。また、動的にリンクを設定することによって、スキーマとは異なった情報を生成できる。本稿では、同時にPIM論理を定義し、これとPIM代数が完全に対応することを証明する。このことで、視覚操作の論理的な表現力を特長付けることができる。

3 個人情報モデル化とデータ操作

ここでは、データ処理モデル PIM(Personal Information Manager) を述べる。これは意味データモデルをベースとし、視覚操作を指向した命令を有する。

3.1 AIS データモデル

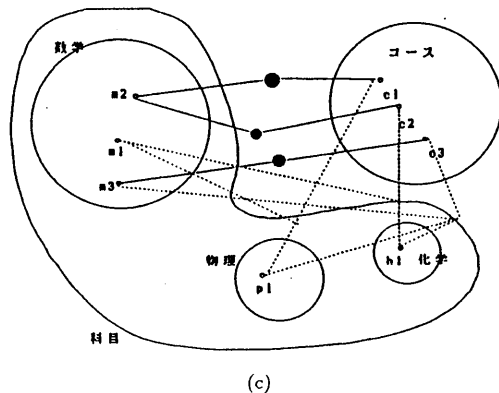
ここでは形式的な説明を避けつつ、意味データモデルの例である AIS(Associative Information Structure) の狙いと基本となる概念を簡単に述べる [AM84,AM86]。PIM では意味データモデル AIS を用いる。PIM で対象となる情報は、「もの」に対応する主体と、「もの」同士の結び付きを表す連想である。共通の特長を持つ主体はその性質を持つと言い、性質自体を主体型と呼ぶ。主体は、一般に複数の主体型を持つ。連想はひと組の主体型の各々に対応する主体組による結び付きと考える。この結び付きの性質を述語、主体型組を定義型と言う。1つの述語には同じ定義型は繰り返して表れない。連想は $p(E_1(e_1)...E_n(e_n))$ と表される。これは結び付き主体の構成 $[e_1...e_n]$ を示しており、 p は述語、 E_i は定義型で、主体 e_i は型 E_i を有することを表す。定義型が明かであれば略すことがある。例 1 (a) は主体型・主体を表し、例 1 (b) は、科目と前提コースの関係を示す。これらは図示することができる。主体は○で表現し、また主体型を集合記号で、述語を◇で表す。例 1 (c) はこのダイアグラムを示す。

[例 1]

記法	主体	型
m1	数1	科目, 数学
m2	数2	科目, 数学
m3	数3	科目, 数学
p1	物1	科目, 物理
h1	化1	科目, 化学
c1	コ1	コース
c2	コ2	コース
c3	コ3	コース

(a) 主体情報

- 前提(科目(m2) コース(c1))
- (b) 連想情報 前提(科目(m2) コース(c2))
- 前提(科目(m3) コース(c3))



AIS モデルでは、構造化された情報(複合値または複合オブジェクト)を扱う。構造化情報は大きく言って、それ自体が独立した意味を持ち主体と同格に扱って良いもの、即ち認識単位で主体と考えるべき場合と、表現を簡単にするために用いられる場合に分けられる。AIS ではこれを直接表現し、操作対象と考える [M87,MS90]。主体と構造値自体とは独立した概念であるため、主体と構造値の組を明示的に宣言する。構造値は構造化について統一した構成規則を有する場合(複合型という)を扱い、主体型と複合型の対応を型構造として宣言する。例 1(c)の点線部はこれを表す。例 2 では、数2の前提コース c1, c2 は「数1と物1」、「数1と化1」の前提科目を持つ事が記述される。

[例 2]

コース	≡	{ 科目 }
c1	≡	{m1, p1}
c2	≡	{m1, h1}
c3	≡	{m1, h1, p1}

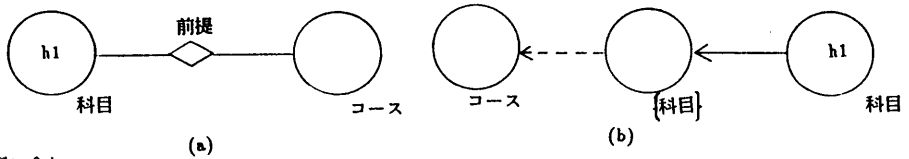
□

3.2 PIM モデルと視覚操作

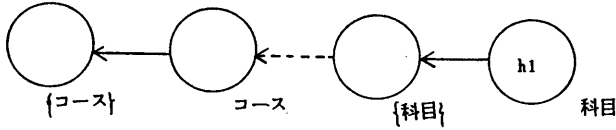
ここではデータを表現し操作するため、パッドの概念を導入する。容易に理解できるように直観的な説明を行う。パッドはスキーマの一部を表す。また、これを評価することで、このスキーマに適合する実現値(連想)集合が得られる。この意味で、パッドはデータベースに対する問合わせとなる。各パッドでは、高々1つの述語と1つ以上の主体型が連結されている。これは述語の定義、即ちひとつの述語名とひとつ以上の定義域を表すものである。前者は、菱形(◇)と名前、後者は集合(○)とその名前です。リンクは菱形と主体集合をつなぐために用いられる。例 3(a) は述語に対応するパッドを、(b) は構造に沿ったパッドを表す(有向点線は型構造を、有向実線はメンバシップを表す)。(c) では、未格納情報を意図している事に注意されたい。集合の名前のないときは、型に関する変数であることを表す。また菱形に名前が含まれないときは、この定義に一致する述語変数を表す。

[例 3]

(a) h1 の前提コース



(b) h1 を前提科目に含むコース

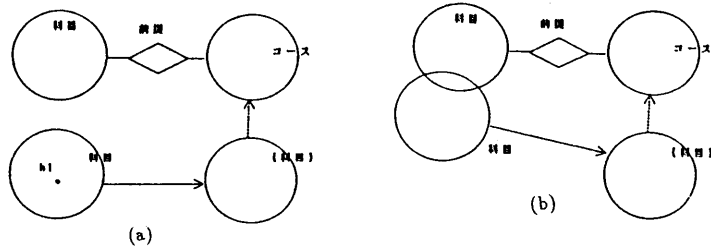


(c) h1 を前提科目に含むコースの集まり

□

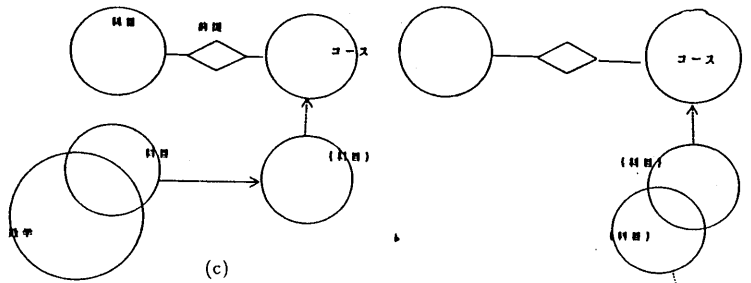
連想は、常に定義述語名・定義型を表している。この一部を変数化することで条件付けができるが、またパッドは複数個集まって条件付けを表すことができる。パッド間の関連性は、2つの値が等しいこと、特定の型構造をとること、タグ付け及び構造の構成に関する記述によって与えられる。この中では、同じ述語が複数回記述されていてもよい。例4でいくつかの場合を示す。

[例 4]



(a) h1 を前提科目に持つ科目

(b) 前提科目に自分自身を含む前提科目



(c) 数学を前提科目とするコース集合

(d) 前提科目集合が一致するコースを持つ「何か」

□

本稿で扱う PIM 論理は、筆者らが提案しているデータ論理と同じ考え方に基づく [MA90, M90]。データ論理は、第一階多類論理に複合型と構造項を導入したもので、意味データモデルの形式化を目的とする。PIM 論理は、述語変数を許す冠頭標準形の式のうち、 \forall, \exists 等の限量子が出現せず、 \neq などの否定比較は許すが、式の否定 ($\neg\psi$) を含まない。この形式は、問い合わせとしても用いられる。例5は例4の PIM 論理による記述を表している。

[例 5]

(a) h1 を前提科目に持つ科目

$$(x/\text{科目})(z/\text{コース})(w/\{\text{科目}\})(h1/\text{物理})(\text{前提}(x,z) \wedge z \equiv w \wedge h1 \in w)$$

(b) 前提科目に自分自身を含む前提科目

$$(x/\text{科目})(z/\text{コース})(w/\{\text{科目}\})(\text{前提}(x,z) \wedge z \equiv w \wedge x \in w)$$

(c) 数学を前提科目とするコース集合

$$(x/\{\text{コース}\})(y/\text{コース})(u/\{\text{科目}\})(v/\text{科目})(w/\text{数学}): (y \in x \wedge y \equiv u \wedge v \in u \wedge v = w)$$

(d) 前提科目集合が一致するコースを持つ「何か」

$$(x/\text{Entity})(y/\text{コース})(u/\text{Entity})(v/\text{コース})(z/\{\text{科目}\}): X(W(x)y) \wedge X(W(u)v) \wedge x \equiv z \wedge u \equiv z \wedge y \neq v$$

□

(a),(b) では構造情報は格納され検索される。(c) のように記述された複合型を基に動的に構造値を計算する事で、質問能力の拡大ができる。ただ「存在しない情報」を解とするには妥当な理由が必要である。例4と例5は同じ答えを得る。これらは互いに一定の規則から得られるのだろうか？本稿の目的は、この問いに Yes と答えようとするものである。

データ論理では(複合型上の)型制約を、単項述語式からなる閉集合と考える。このクラスには主要な主体型制約(包含従属, 被覆制約, 排他制約, 存在従属など)が含まれる。このクラスは充足性が決定可能であるという大きな特長を持つ [AM86,MA90]。PIM 論理でもこれを用いることで、型制約の含意, 無効な記述, 冗長性などかチェックできる。

4 視覚操作-PIM 代数

ここでは PIM モデルで視覚操作に基づく代数操作を導入し、形式化を行う。証明や詳細は [MM90] を参照されたい。

パッドは、菱形(◇)、集合およびリンクから構成される空でない形状表現で、それぞれ述語、主体集合または複合オブジェクト集合、定義型であることを表し、次の条件を満たす。

1. ただひとつの集合だけからなるパッドを主体パッド、菱形といくつかの集合がリンクでつながったものを連想パッドという。パッドとは、このいずれかである。
2. 集合はラベル付けられて良い。1つの連想パッド内の集合ラベルは相異なる。この名前は、主体型名または複合型名を表す。各集合に値が高々ひとつ記入されていてよい。
3. 菱形は述語であることを示す。菱形がラベルを持てば述語名を意味する。ラベルは p の形か $\neg p$ の形をしている。 \neg のついたラベルは複数あってよい。菱形には自然数(次数という)が与えられていて良い。但し、次数が明示されていないときはリンクの数を次数とする。リンクの数が次数に満たないときは、ラベルのない集合を補うと考える。
4. リンクは菱形と集合を結び、定義型であることを表す。1つのパッド内の構成要素は連結せねばならない。

パッド内およびパッド間に関連付けを与えることができる。即ち、構造記述(structure)、構造付け(lambda)、重ね合わせ(overlap)およびタグ付け(tag)である。

構造記述されるのは、主体パッドで複合型の場合だけで、そのデータ構造の構成を記述している。これらは条件の記述に従って、有向実線であらわす。有向実線の条件は、ラベルなし、 n, e, g, i (i は自然数)のいずれかの印によって表される。有向実線が E_i ($i = 1 \dots n$) からこの主体パッドに向かって入力しているとする。そのラベルすべてに n が付いていれば、この複合型集合は $E_1 \cap \dots \cap E_n$ であることを表す。ラベルが自然数 i であれば、直積 $E_1 \times \dots \times E_n$ を表しており、成分位置を表すため添え字 i がつく。ラベルなし、 e または g があれば、集合 E_i の有限べき集合であることを表す。このとき、 E_i はすべて同じ複合型集合である。ラベルが無いときは所属条件で、 g が付くときは非所属条件を、そして e であれば、所属するための必要十分条件を表す。なおこれらが混在することは許されない。ただし、この有向実線に関しては輪(サイクル)が無いとする¹。例3(b),(c)は有向実線がメンバシップを表している。

構造記述された複合型は、次のような名前が自動的に求められる。これを型の構成に従う名前という。ラベルがあるときでも、これに一致せねばならない：(1) 入力実線の無い集合がラベル Δ を持てば、その集合の型構成はそれを、さもなければ Entity とする。(2) 1つ以上の集合の型の構成がすべて Δ であれば、それから得られる集合型の型構成を $\{\Delta\}$ 、型構成が共通していなければ誤りである。(3) 同様に、集合の型構成がそれぞれ $\Delta_1, \dots, \Delta_k$ であれば、それから得られる順序組型の型構成を $\langle \Delta_1, \dots, \Delta_k \rangle$ とする。

主体集合と複合型集合との間の構造付けとは、2つの型が型構造実現値の関係にある事を示すもので、構造付けられる方向への有向点線で表される。この点線上に \rightarrow があってよい。構造付けに輪が存在してもよい²。例3(b),(c)では、 $\text{コース} \equiv \{\text{科目}\}$ に対応した関係が有向点線で示されている。

パッド内の重ね合わせとは、そのパッド中の集合の交わりを言う。物理的な交わりで、これを表す。2つのパッドの重ね合わせとは、集合の交わり、述語の交わりのいずれかをいう。述語の交わりを表すため、菱形の交わりを用いる。定義から、主体パッド同士や主体

¹ 例えば、主体パッド E から自分自身へラベル 1 の有向実線があるとき、これが意味する型集合は PIM ではあつかえない。

² 同じ主体パッドの基本型 E から E へ直接有向点線があるときも、同一主体の構造付けを意味するものでない。

パッドと連想パッドが重ね合わされてよい。集合の交わりに \neq や θ が記されていることがある。例 4(a) では、「科目」から {科目} への構造記述があり、コースへ {科目} から構造付けされている。(b) では科目の重ね合わせがなされ、同じ科目が条件で問われている。(c) では主体パッドとの重ね合わせがある。(d) ではラベルのない集合や菱形が生じており、型名の具体的な記述を与えていない。記号のタグ付けとは、菱形同士や集合同士に、同じ印³を付けることを言う。この印をタグという。重ね合わせと異なる点は、述語名や型名の同一性だけを主張する。

パッド集合は、データベースに対する問合わせを記述する。連想パッドに対する答えは、パッドが表すパターンに一致する連想の集合である。答えの意味を厳格にするため、適合性の概念を導入する。連想 $p(E_1(e_1) \cdots E_n(e_n))$ が 1 つの連想パッドに適合する (またはこの連想パッドの答え) とは、つぎを充すときをいう。

1. 菱形がラベルを持てば、それは p に等しい。但し $\neg q$ の形ならば q と p は同じ述語記号ではない。
2. このパッドの次数は n である。
3. 集合がラベル F を持てば、ある定義型 E_i である。その集合が値 c を持てば、 e_i に等しい。
4. 集合がラベルを持たないとき、ある定義型 E_i が存在し、 E_i は集合ラベルとして表れておらず、その集合が値 c を持てば、 e_i に等しい。

主体または複合オブジェクト e が主体パッドに適合するとは、主体パッドが型構成 T を持つとき、 e は次の条件にしたがう時を言う：

1. 入力実線のない主体パッドのラベル T ならば、 e はその主体集合の要素である。
2. 入力実線に E から n のラベルがあるならば、 e は E に対応するパッドに適合する。
3. E_i からラベル i の付いた入力実線であれば、 e は順序組オブジェクト $\langle u_1 \cdots u_n \rangle$ で、第 i 成分 u_i は E_i に適合する。
4. E からラベルなし、 ϵ または E から e の入力実線があれば、 e は集合でその要素で E に適合するものがある。リンクに ϵ が付いていれば、どの要素も E に適合しないことを示す。リンクに記号 e が付いていれば、 e の要素は E に適合するものに限られる。

次に関連付けの適合性の意味を定義する。2 つのパッドのそれぞれに適合する 2 つの連想が与えられたとする。2 つのパッドが重ね合わされるとき、それらの連想が適合するとは、次を言う。

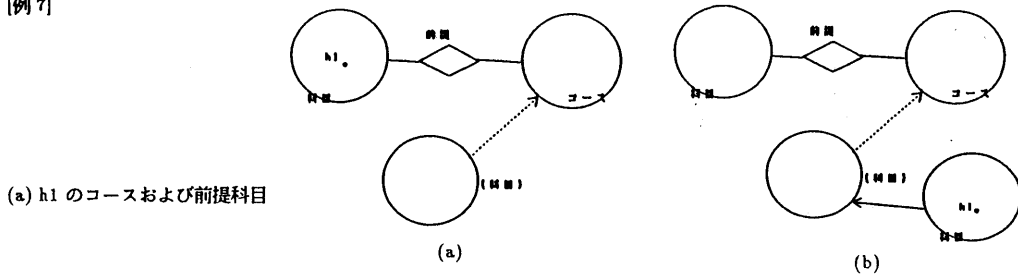
1. 集合が交わる時、当該するものは同じ主体または複合オブジェクトである。但し、条件 \neq があれば同じ主体ではないことをしめし、 θ であればこの条件を満たす。
2. 菱形が交わる時、同一の連想⁴である。

構造付けの適合性を定義する。 e は主体型 E に、 t は複合型 T に適合するとする。型構造の実現値 $\lambda(e) =_{E,T} t$ が存在するとき、 e と t とは型構成 T から主体集合 E への構造付けに適合するという。 \neg があれば、この型構造実現値は存在しないことを意味する。

タグ付けの適合性を定義する。2 つの述語記号に同じタグがあれば、同じ述語記号を意味し、集合記号に同じタグが付いていれば、同じ型名を意味する。

パッド集合のデータベース DB 上での評価結果とは、各パッドに適合する DB 中のすべての連想を言う。

[例 7]

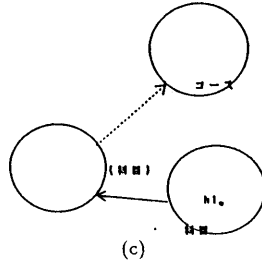


(b) h1 を前提科目に含む科目

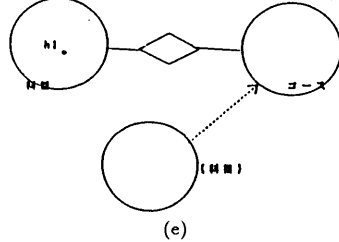
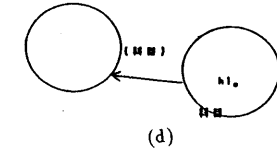
³ラベルとは記号が重ならないとする。

⁴それぞれのラベルで \sim を伴うものがあれば、どの名前にも該当しない。

(c) h1 を前提科目に含むコース



(d) h1 を含む科目集合



(e) h1 のコースと科目集合との何かの関係をもつもの

□

これまで見てきたように、連想はデータ自体であり、パッドはスキーマに関する情報(メタデータ)を表現する。述語定義 $p[E_1 \dots E_n]$ は、 p が述語であること、 $E_1 \dots E_n$ は主体型で、 p の定義域となっていることを意味する。PIM では、これらの定義を記述するためディクショナリを用いる [AM84, MM89]。この存在を仮定すると、型名や述語名を値とする変数を用いることができる。述語変数は必ず既存のものに割り当てられ、いわゆる高階操作には当たらない。

以上の基本概念を用いて PIM 代数を定義する。PIM 代数の操作には 2 種類あり、問い合わせを意識した代数操作と、それらの組立や加工を目的とする編集操作がこれである。

評価 (Eval) は、パッド集合とデータベースまたは連想集合から適合する連想を取り出す操作である。重ね合わせ (Overlap) は指定した 2 つの集合または菱形を重ね合わせる。構造付け (lambda) は、主体集合と、複合型集合パッドとの型構造対応を記述する。構造記述 (Structure) では、複合オブジェクトの条件を次々に詳細に記述するもので、例えば $\{E\}$ に対して、 E パッドをその都度生成あるいは修正する。タグ付け (Tag) とは、指定した 2 つの記号に新しいタグを与える。フレーム (Frame) は、指定された連想とその定義を同時に表示するよう指示する。即ち、次のような対応を言う；連想 $\{e_1 \dots e_n\}$ に対して述語定義 $p(E_1(e_1) \dots E_n(e_n))$ で、各 e_i が主体型 E_i を有するものを、菱形と集合を用いて表す。複合オブジェクトや型構造実現値に対しても同様である。拡大 (Zoom) は、指定された主体パッドから、それを構成する複合型のパッド集合を表示させる操作である。

これらの評価を伴う命令に対して、次のような編集操作がある。分類 (Sort) は、連想集合を述語名の等しいものに分類する。選択 (Choice) は、連想集合からひとつを選び出す。合併 (Merge) は、2 つの連想集合を 1 つにし、同じ連想を取り除く。除去 (Delete) によって、連想集合から特定の連想、または他の連想集合にあるものを取り除く。ページ (Page) は、表示方法を指示する操作である。連想内容を変更せず、連想集合を対象としている。そのほか、表示場所の移動 (Move)、サイズの変更 (Resize) 等の編集命令が用意される。

5 プログラム-PIM 論理

5.1 PIM 論理の定義-構文

ここでは、第一階述語論理を拡張した論理操作を定義する。これは既に提案されているデータ論理と同じ考え方に基づく [MA90, M90]。本来、データ論理は、第一階多類論理を拡張したもので、次の目的を持つ：(1) 複合オブジェクトを項として扱う、(2) AIS データモデルを解釈とする、(3) 述語論理に基づいて操作する。

記号系(定数、変数、述語、関数、型)の型付けは述語に対してのみなされ、項は解釈時に主体型と対応付けられる。型の記述は、型記号の組合せによる結合と順序組化・集合化を用いる。整式は基本式のブール結合に限量部分が加わったものを言う。データ論理式とは、自由変数がすべて型記述部として冠頭部に表れた整式をいう。本来、意味データモデルの形式化のために提案されたものであるが、導出原理に基づく完全な証明手続きを構築できる [M90]。この考え方は PIM でも共通している。

PIM 論理の構文を定義する。C を定数の高々可算無限集合とする。同様に、V を(項の)変数集合、P を述語集合、U を(述語の)変数集合、T を型名集合、S を型変数集合とする。これらは、互いに交わらない。以下では、定数を $e, f, \dots, e_1, \dots, f_1, \dots$ 、変数を v, \dots 、述語変数を X, X^n, \dots 、述語名を p, q, \dots 、型を t, t_1, \dots 等と表す。ここで n は自然数である。

複合オブジェクトや複合型を定義する。集合 W に対して、 $\eta(W), \Delta(W)$ を次とする。

1. W の要素は、 $\eta(W), \Delta(W)$ に含まれる。
2. $e_1 \dots e_n$ が $\eta(W)$ の要素なら $\{e_1 \dots e_n\}$ も要素である。 $\Delta(W)$ についても同じ。但し、 $n > 0$ とする。
3. $e_1 \dots e_n$ が $\eta(W)$ の要素なら $\{e_1 \dots e_n\}$ も要素である。但し、 $n \geq 0$ 。 e が $\Delta(W)$ の要素なら、 $\{e\}$ も要素である。
4. e_1, e_2 が $\Delta(W)$ の要素ならば、 $e_1 \cap e_2$ も $\Delta(W)$ の要素。

PIM 論理での項とは、 $\eta(C \cup V)$ の要素または $t.n$ の形 (t は項, n は自然数) をいう。型 (または複合型) とは、 $\Delta(T)$ の要素を言う。特に T の要素を基本型という。型構造とは、 $R \equiv R_0$ の形の有限集合を言う。ここで R は基本型, R_0 は型である。次数 n の述語名 p には相異なる n 個の基本型 $E_1 \cdots E_n$ が与えられ、これを p の定義型という。 $p(E_1 \cdots E_n)$ と表すことがある。

基本式とは、項を $s, t_1 \cdots t_n$, r を変数を含まない項, X, Y を述語変数, p を述語記号, W_i を型変数, α を型, R_i を r/α の形とすると、次のうちのいずれかをいう：

素式 $(R_1 \cdots R_m)p(E_1(t_1) \cdots E_n(t_n)), (R_1 \cdots R_m)X(E_1(t_1) \cdots E_k(t_k)W_{k+1}(t_{k+1}) \cdots W_n(t_n))$

メンバシップ $(R_1 \cdots R_m)s \in t, (R_1 \cdots R_m)s \in t$

型構造 $(R_1 \cdots R_m)t \equiv_{R, R_0} s, (R_1 \cdots R_m)t \neq_{R, R_0} s$

項等式 $(R_1 \cdots R_m)t = s, (R_1 \cdots R_m)t \neq s, (R_1 \cdots R_m)t \theta s$

述語等式/型等式 $X = Y, X \neq Y, X = p, X \neq p, W_1 = \alpha, W_1 = W_2,$

ここで θ は $=, \neq, \equiv, \neq$ 以外の比較演算子で、その意味は各型で定められているとする。素式では型変数は述語変数と共にのみ出現可能であることに注意したい。 R_i の形を基礎限量という。

式とは、 $n \geq 0$ にたいして $Q_1 \cdots Q_n : \Psi$, ここで Ψ は基本式の \wedge, \vee, \neg による結合の形を言う。限量子 Q_i は x_i/α_i の形で、 x_i は他の Q_j にはなく、 α_i は型を表す。式の有限集合をプログラムという。問い合わせ式とは、 \neg を含まない $n > 0$ の形の式を言う。

\vee を含まない問い合わせ式が安全とは、次を満たす時を言う。

1. Ψ に生じている変数はすべて限量部に生じている。
2. どの述語変数 X についても素式にも表れている。

5.2 PIM 論理の定義-解釈

主体集合と主体型集合が与えられているとする。 C の要素には主体が、 $\eta(C)$ の要素は、構造の構成方法にしたがった対応がなされる。変数は代入に相当する。 T の要素 t には主体型 t' が対応する。どの主体も主体型 $Entity$ をもつ。主体がもつ型を特定することを、主体の型付けという。型 t に対して主体集合を考えることができる。これを t の主体集合という： $\{c | c$ は型 t をもつ $\}$ 。以下ではこれを同一視して論じる。 C の要素 (定数) ではなく、その解釈 (主体) に対して型との関連を論じることに注意したい。 $\Delta(T)$ の各要素には、構成的に集合を構成する。即ち、型の順序組化にはそれぞれの集合の直積を、集合化には有限集合の集まりを対応させる。 $\eta(C)$ の要素が $\Delta(T)$ の要素 t に従うとは、その解釈が t の外延集合に含まれることを言う。

型構造 $t \equiv s$ の解釈として次の部分集合を与える： $\{(e, f) | e$ は型 t を持ち f は型 s に従う $\}$ 。ただし、 t のどの要素 e についても高々ひとつの f が対応するとする。ちょうど AIS モデルの $\lambda(e) = f$ に相当することに注意したい。

$t.n$ は、 t が順序組でその第 n 成分を示す。述語名 $p(E_1 \cdots E_n)$ には、直積 $E_1 \times \cdots \times E_n$ の部分集合が解釈値として与えられる。

式 $(R_1 \cdots R_m)p(E_1(t_1) \cdots E_n(t_n))$ が真とは、各 R_i を r/α の形とすると各 r の解釈値が α の外延集合に含まれ、 $t_1 \cdots t_n$ の解釈値が p の解釈に含まれる時を言う。 $p(E_1(e_1) \cdots E_n(e_n))$ のような記法を導入することで、位置情報を指定しなくてよいことに注意された。例えば、これは次と同じ連想を表す： $p(E_n(e_n) \cdots E_1(e_1))$

$(R_1 \cdots R_m)X(E_1(t_1) \cdots E_k(t_k)W_{k+1}(t_{k+1}) \cdots W_n(t_n))$ が X の p への代入で真とは、各 R_i を r/α の形とすると各 r の解釈値が α の外延集合に含まれ、代入で $t_1 \cdots t_n$ の解釈値が p の解釈に含まれる時を言う。基本式に表れている型名 $E_1 \cdots E_k$ は p の定義域 $E_1 \cdots E_n$ に含まれ、型変数 $W_{k+1} \cdots W_n$ は $E_{k+1} \cdots E_n$ を適当に並べ代えたものである。 W_i を略すことがある。

$s \in t$ は通常の集合のメンバシップであり、 $(R_1 \cdots R_m)s \in t$ はその否定を表す。 $t \equiv_{R, R_0} s$ が真とは、型構造 $R \equiv R_0$ の解釈に (t, s) の解釈値が含まれる時を言う。 $t \neq_{R, R_0} s$ が真とは、型構造 $R \equiv R_0$ の解釈に (t, s) の解釈値が含まれない時を言う。 $t = s$ は同じ主体である時に真となる。これは複合オブジェクトに対しても定義される：順序組ならば次数が同じで構成要素毎に同じ複合オブジェクトを、集合であれば集合として等しい (一方の要素が他方のいずれかと等しい) ことを表す。 $t \neq s$ は同じ主体ではない時に真となる。 $t \theta s$ は、2つの項が比較演算子 θ を満たす時を言う。

述語変数 $X = Y$ が真とは、同じ述語記号が代入された時とする。 $X \neq Y, X = p$ なども同様に定義される。 P にないものをとる事はなく、高階論理とは異なる。型に関する等式 $W_1 = W_2$ が真とは、同様に同じ主体型名に対するものとする。

式 $Q_1 \cdots Q_m : \Psi$ が真とは、限量子の型の任意の要素を対応する変数に一斉に代入したとき、その代入で Ψ が真のときを言う。プログラムが真とは、この解釈で全ての式が真となることを言う。この解釈をプログラム Ω のモデルという。

$e_1 \cdots e_m$ が問い合わせ式 $Q_1 \cdots Q_m : \Psi$ の解とは、 Ω のモデルの上で、各 e_i は t_i の外延集合の要素であり、述語変数や形変数があればこれを適当に代入し、かつ各 x_i を一斉に e_i に置き換えたとき、 Ψ が真となる時を言う。

第一階述語論理で成立する性質が成り立つことは、簡単に判る。

[定理 1]

次は等価、つまり真偽値に関しては同じである。ここで Q は限量部を、 R は基礎限量を表す。ただし、 Q の限量部のすべての型の外延集合は空でないと仮定する。

- (1) $Q : (\Psi_1 \wedge \Psi_2)$ と $(Q : \Psi_1) \wedge (Q : \Psi_2)$
- (2) $Q : (\Psi_1 \vee \Psi_2)$ と $(Q : \Psi_1) \vee (Q : \Psi_2)$

- (3) $(R)(\Psi_1 \wedge \Psi_2)$ と $(R)\Psi_1 \wedge (R)\Psi_2$
 (4) $(R)(\Psi_1 \vee \Psi_2)$ と $(R)\Psi_1 \vee (R)\Psi_2$
 (5) $x/T: \Psi$ と Ψ ただし, Ψ には変数 x は生じていないとする。

□

この結果, 式 $Q: \Psi_1 \vee \dots \vee \Psi_n$ の解は, 各 $Q: \Psi_i$ の解の和に等しい。

例 6 に対応して, 次の例 (a),(b),(c) は述語とそれが関わる構造値との関連を問いかけている。(d) では集合構造だけを利用しており, 科目集合で $h1$ を含む全てがアクセスされる。(e) は述語記号を問いかけている。

[例 7]

- (a) $h1$ のコースおよび前提科目

$$(y/\text{コース})(x/\{\text{科目}\}): (h1/\text{科目}) \text{前提}(h1, y) \wedge y \equiv x$$

- (b) $h1$ を前提科目に含む科目

$$(x/\text{科目})(y/\text{コース})(w/\{\text{科目}\}): \text{前提}(xy) \wedge y \equiv w \wedge (h1/\text{科目})h1 \in w$$

- (c) $h1$ を前提科目に含むコース

$$(y/\text{コース})(x/\{\text{科目}\}): y \equiv x \wedge (h1/\text{科目})h1 \in x$$

- (d) $h1$ を含む科目集合

$$(x/\{\text{科目}\}): (h1/\text{科目})h1 \in x$$

- (e) $h1$ と関係するコースと科目集合

$$(x/\{\text{科目}\})(y/\text{コース}): (h1/\text{科目})X(h1y) \wedge y \equiv x$$

□

5.3 PIM 操作の完全性

安全な PIM 論理操作と代数操作は“対応する”ことが証明できる [MM90]。言い替えると, 関係モデルと同様, PIM 代数は完全である。この結果, PIM 代数操作は安全な PIM 論理操作の表現力に等しくなり, 本稿で提案する視覚操作の特長付けが行える。

[定理 2]

与えられたバッド質問 (PIM 代数式) と等価で安全な PIM 論理式が得られる。

[定理 3(完全性)]

すべての安全な PIM 論理式に対して等価なバッド質問が存在する。

6 結び

本稿では, 個人情報管理のための PIM データモデルを AIS 意味データモデルに基づいて提案した。モデルでは, 主体連想や主体型, 述語などこれまで提案された基本概念の他に, 複合オブジェクトを直接扱うメカニズムを持つ。視覚操作に基づく PIM 代数と, データ論理の部分クラスである PIM 論理操作を提案した。前者は, データベース内をブラウズする為に用いられ, とくに述語や主体型を問いかけたり, またそれらのつながりを問うことができる。後者は形式的に定義された言語で, 問い合わせの評価や解釈の方式を厳格に定義する役割を持つ。

本稿の最大のねらいは, 視覚操作に基づくモデルが明確な論理的枠組みと対応することを示すことにあった。従って, より現実的で操作性を向上させたインタフェースが重要となるのは明かである。

謝辞 日頃ご指導頂いている小林功武教授 (産能大), 有沢博助教授 (横浜国大) に感謝します。

参考文献

- [A87] Arisawa, H.: パーソナルデータベース, 電子情報通信学会誌 Vol.70-12, 1987
 [AM84] Arisawa, H. and Miura, T.: Formal Approach to Database Description, IEEE COMPCON, 1984
 [AM86] Arisawa, H. and Miura, T.: On the Properties of Extended Inclusion Dependencies, VLDB, 1986
 [CKW89] Chen, D., Kifer, M. and Warren, D.: HiLog as a Platform for Database languages, DBPL, 1989
 [Go87] Goodman, D.: The Complete Hyper Card Handbook, Bantam Books, 1987

- [HT89] 堀尾, 津田他: アイコニックブラウザの開発, 情報処理学会データベースシステム研究会 70-4, 1989
- [KM84] Krishnamurthy,R. and Morgan,S.P.: Query Processing on Personal Computers, VLDB, 1984, pp.26-29
- [La86] Larson,J.A.: A Visual Approach to Browsing in a Database Environment, IEEE Computer, 1986-6, pp.62-71
- [LM84] Lyngbaek,P. and McLeod,D.: A Personal Data Manager, VLDB, 1984, pp.14-25
- [M87] 三浦: 非正規関係データベース理論の動向, データベースシンポジウム, 1987
- [M90] 三浦: 証明手続きとしてのデータ論理, 情報処理学会データベースシステム研究会, 1990-1
- [MA90] Miura,T. and Arisawa,H.: Logic Approach of Data Models - Data Logic, Future Database Systems, 1990
- [MM89] 盛屋、三浦: 個人情報の管理環境, 情報処理学会全国大会 1989 年度後期, 2M-2
- [MM90] Miura,T. and Moriya,K: On the Completeness of Visual Operations for Databases, in preparation, 1990
- [MS90] 三浦, 塩谷: 複合オブジェクトに基づく演繹データベース, 情報処理学会誌 Vol.31-2,1990
- [Mo86] Motro,A.: Constructing Queries from Token, SIGMOD, 1986, pp.120-131
- [NS88] Neuhold,E. and Schrefl,M.: Dynamic Derivation of Personalized Views, VLDB, 1988, pp.183-191
- [U89] Ullman,D.D.:Principles of Database Systems, Computer Science Press, 1989