

情報代数

山崎 利治

はじめに

これは 1962 年に報告された情報代数 [1] を私流に整理したものである。情報代数は事務処理の機械に依存しない非手続き的な問題定義言語を求める目的で、R. Bosak がまとめたものである。現在なら仕様記述言語というところであるが、高水準プログラム言語（スーパー COBOL）と捉えその処理系も考えていたようである。

当時は磁気テープが入出力の主要な媒体であって、その影響が機械に依存しないといいつつ情報代数の基本枠組みにみられる。そうではあるが事務処理問題の仕様記述について今日でもなお参考になると考え、あえて紹介する次第である。

1 属性空間

仮定 1 対象 e とその有限集合 E が与えられる。

対象は事象など事務処理に現れるなにかである。たとえば、給与計算の従業員、売掛金計算の顧客などである。事務処理に出現するレコードはこの対象の一側面を表現するものである。

仮定 2 集合族 $(V_q)_{q \in Q}$ が与えられる。 Q は有限集合である。その濃度を n とする。

q を属性 (property), V_q を q 種値集合 (q value set) という。各 V_q は Ω (無関係), Θ (未知) を含む。 V_q の例としては、自然数、整数、実数、真偽値、文字列などにそれぞれ Ω, Θ を付加したものなどがある。属性としては、従業員の名前、時間給、曜日、労働時間などがある。

定義 1 $P \stackrel{def}{=} \bigotimes_{q \in Q} V_q$ (直積) を属性空間 (property space) という。 P の要素 $p = (v_1, v_2, \dots, v_n)$ を点 (datum point) という。

点とは事務処理に出現するすべてのレコードがそれによって表現できるという(超)レコードである。そのために各属性値に無関係をもたせたわけである。対象 e の属性 q に対する値が $q(e) = v \in V_q$ のように定まると考える。つまり、対象は属性空間の中で表現できる ($\rho: E \mapsto P$) と考えるわけである。このとき ρ が 1 対 1 写像であることを要請する。

定義 2 P の点 p に対してある V_q の値を与える関数 $f_q^p: P \rightarrow V_q$ を点関数 (function of points) という。

定義 3 P の点の有限集合 $A \subseteq P$ を面 (area) という。面に対してある V_q の値を与える関数 $f_q^a: \mathbb{F}P \rightarrow V_q$ を面関数 (function of areas) という。 $\mathbb{F}P$ は P の有限部分集合全体をあらわす。

面とはレコードの集まり、つまり、ファイルのことである。

定義 4 k 個の面 A_1, A_2, \dots, A_k の直積 $A_1 \times A_2 \times \dots \times A_k$ の要素 $t = (p_1, p_2, \dots, p_k)$ を k 組 (k-tuple) という。組に対してある V_q の値を与える関数 $f_q^{t,k}: P^k \rightarrow V_q$ を組関数 (function of tuples) という。 P^k は P の k 個の直積をあらわす。

2 束

定義 5 面 A_1, A_2, \dots, A_k と射 $\beta_{\mathbb{B}}^{t,k}: P^k \rightarrow \mathbb{B}$ があるとき、束 (bundle) $B(\beta_{\mathbb{B}}^t; A_1, A_2, \dots, A_k)$

とは (\mathbb{B} は真偽値集合),

$$\{ t \in A_1 \times A_2 \times \cdots \times A_k \mid \beta_{\mathbb{B}}^{t,k}(t) = true \}$$

をいう. $\beta_{\mathbb{B}}^{t,k}$ を束構成関数 (bundling function) といい, また, 束に対して面を与える関数を束関数 (function of bundles) という. 束関数 $F : B \rightarrow P$ は束の構成要素である組 t に対して点 p を対応する組関数 $f_q^t : B \rightarrow V_q$ によって $F(t) = (f_{q_1}^t(t), \dots, f_{q_n}^t(t))$ と定義できる.

これはいわゆるファイルの突合せを行うための場を構成する.

3 団

定義 6 A を面, $\gamma_q^p : P \rightarrow V_q$ を P からある q の V_q への点関数とする. 団 (glump) $G = G(\gamma_q^p, A)$ とは $A_v = \gamma_q^p(v)$ ($v \in V_q$) として,

$$\{ A_v \mid v \in V_q \}$$

をいう. γ_q^p を団構成関数 (glumping function) といい, また, 団に対して面を与える関数を団関数 (function of glumps) という. これは団の要素である面上の面関数から束関数と同様に定義する.

A_v たちは A の類別になっている. すなわち, $A = \bigcup_{v \in V_q} A_v$ かつ $u \neq v \Rightarrow A_u \cap A_v = \emptyset$ が成立する. これはいわゆる多段階集計などファイル内レコードを複数個集めて計算する場を構成する.

4 補助記法

事務処理の仕様を束や団を構成し, 束関数や団関数を与えて記述する. このときの便宜のためにつぎのような記法を用意する.

記法 1 束 $B(\beta; A_1, \dots, A_k)$ に対して, その束構成面 A_i の点で B の要素組の構成要素となるもの全体を $I(B, A_i)$ と書く. $I(B, A_i) \subseteq A_i$ である.

記法 2 束 $B(\beta; A_1, \dots, A_k)$ に対して, その束構成面 A_i の点で B の要素組の構成要素とならないもの全体を $I'(B, A_i)$ と書く. $I'(B, A_i) = A_i \setminus I(B, A_i)$ である. $X \setminus Y$ は集合 X から集合 Y を引いた集合差をあらわす.

5 例題

簡単な週給計算問題を例示する. 情報代数の報告書の例をほんの少し変更したものである. 給与ファイル OP, 勤労ファイル DW, 新入従業員ファイル NE, 退職従業員ファイル LE がある. その詳細は属性空間を示した表 1 をみてほしい (x は関係ありの意味). これらのファイルから新給与ファイル NP をつくりたい. 記法については後で補足するに留めるが, 仕様記述の枠組みの概要は理解できるだろう.

$$B1 = B(OP.ManId = LE.ManId; OP, LE)$$

$$T = I'(B1, OP)$$

$$H1 = H(G(DW.ManId, DW))$$

$$H2 = H(G(DW.ManId, DW))$$

$$NP = F_1(B(H.ManId = T.ManId; H1, T))$$

$$\bigcup F_2(B(H.ManId = NE.ManId; H2, NE))$$

let $f(x : \mathbb{R}) \equiv x \leftarrow (x < 8) \rightarrow 1.5 * x - 4$

in

$$H \equiv \begin{cases} H.ManID & = DW.ManId \\ H.Hours & = \Sigma[f(DW.Hours)] \end{cases}$$

$$F_1 \equiv \begin{cases} NP.FileId & = "PF" \\ NP.Total & = T.Total + g \\ NP.Period & = T.Period + 1 \\ NP.Salary & = g \\ where & g = H.Hours * T.Rate \end{cases}$$

$$F_2 \equiv \begin{cases} NP.FileId & = "PF" \\ NP.Total & = NE.Total + g \\ NP.Period & = NE.Period + 1 \\ NP.Salary & = g \\ where & g = H.Hours * NE.Rate \end{cases}$$

表 1: 週給計算属性空間

属性	値集合	OP	DW	NE	LE	NP
FileId	{PF,DW,NE,LE}	PF	DW	NE	LE	PF
ManID	N	x	x	x	x	x
Name	CharStr	x	Ω	x	x	x
Rate	\mathbb{R}	x	Ω	x	x	x
Hours	[0,24]	Ω	x	Ω	Ω	Ω
Day	[0,6]	Ω	x	Ω	Ω	Ω
Total	\mathbb{R}	x	Ω	Ω	x	x
Period	[0,52]	x	x	x	x	x
Salary	\mathbb{R}	x	Ω	Ω	x	x

記法について一言． $x \leftarrow y \rightarrow z$ は if y then x else z を意味する．上に出てきたこの形の例は一日の勤労時間が8時間を超えるときは超えた分の時間給が5割増しになるという関数定義である．

束や団の上の束関数や団関数の定義においてそれが定義する面の点で無関係としたものはやはり無関係であり，関数定義中に陽示していない属性については定義域に存在する属性値が割り当てられるとする．

たとえば，束関数 $F_1(F_2)$ の定義において，属性 $ManID, Name, Rate$ などは明示していないが，これらの値については束に存在する値が与えられることになる．また，団関数 H がつくる面などの属性空間の表にあらわれない一時的な面の点の明示されない属性値はすべて無関係とする．

おわりに

情報代数は一括処理の全盛期に生まれた．当時は磁気テープによる逐次データファイルを分類(ソート)しては突合せ処理を行っていた．その時代にファイルを読む，ファイルに書くなどの詳細手続きを捨象し処理の本質を抽出したいとした情報代数は大容量ファイルを伴う乱処理時代の現在でも示唆に富んでいる．

また現実の課題は複雑である．たとえば，関係データベースのスキーマ図などで関係(レコード)の数が1000を超えるものがあるという．このような例を見聞すると，情報代数の属性空間の考えは非現実的とも思える．そこでは部分空間などの考えを枠組みに導入しなければならないだろう．ここでも構造化の機構が不可欠である．

他方現在の事務処理問題では対話型また分散処理などの利用形態やCORBA，データベース管理系，コンポーネントウェアなどという実現環境もあって実現が便利な反面仕様記述では面倒になっている．仕様にはそれらの側面についても書かなければならない．しかし，そのような詳細を一旦は忘れず事務対象分野のモデル構成言い換えれば理論構成をおこなうべきであろう．事務処理の仕様化の枠組み検討の出発点として，形式的な仕様記述例のひとつとして情報代数の再読を薦めたい．

参考文献

- [1] Language Structure Group of CODASYL Development Committee, *An Information Algebra Phase I Report*. Communications of the ACM, 1962(pp190-203).