

フォームの設計法

上林 弥彦

(京都大学工学部)

1. まえがき

オフィスシステムにおいて、利用者にと、て分かり易いフォームは非常に重要である。ところが、フォームの設計については直感的な手法しか用いられていない。本稿では、従属性制約や索引条件を用いたフォームの設計や、すでにあるフォームにデータベース演算をほどこした場合の結果のフォームの設計を与えられたフォームの持つ制約を利用しておこなう方法について述べる。

2. 基本的定義

フォームには種々の形式のものが考えられる。このため表現形式に独立なフォームの一般的性質と、表現形式に従属したフォームの性質の2種を分けて扱うことが非常に重要である。

本稿ではフォームに共通した性質を表わしていると考えられる抽象フォームを定義し、その性質や等価性について検討

する。抽象フォームは、正規表現を用いたフォームの表現形式で、集合を*表現に対応させ、1対1対応や直積対応等を表わすことができる。

フレームは、抽象フォームのデータをどこに置くかを示したもので、抽象フォームでは扱われていないデータ間の位置関係を表わしている。抽象フォームの各データをフレームのどの位置に対応させるかを決める対応関数と、抽象フォーム、フレームによってフォームが定義される。抽象フォームにおけるデータの位置の変更は対応関数の変更によって元へ戻せるため、抽象フォームではデータに可換性がある。

抽象フォーム F は、 $F = [X, S]$ で定義される。ここで、 X は属性集合であり S は X の要素による正規表現で構造を示すものである。正規表現を用いると次のような区別ができる。

- (1) 単純値に対応する属性と集合に対応する属性の区別
- (2) 集合間の対応として、1対1対応と直積対応の区別

属性値として空値が許されているかということや、集合に対応する属性に対して空集合が許されているかということも合わせて示すため、次のような表現を用いる。

A^* : 属性 A は集合に対応し、空集合も許される。

A^+ : 属性 A は集合に対応しているが、空集合は許されない。

A : 属性 A は単純値に対応しており、空値をとることも許されている。

A^1 : 属性 A は単純値に対応しており、空値をとることは許されていない。

各属性にはその属性データに必要なデータ型が、集合に対する属性や属性集合には順序集合かどうかといった区別が定義される。

抽象フォームは、データも表現する場合に必要な領域の大きさ等を全く考えていない表現であり、現実のフォームではこのような制限が必要である。

フォーム Q は、 $Q = [F, P, M]$ で定義される。ここで、 P は抽象フォームにおける属性とフレームの変数領域を対応付ける写像である。 M はさらに、各属性集合に対応するデータはどのように配置するかを決めるものとする。

1つの抽象フォームに対していくつものフレームが用意されることもある。たとえば、

(1) 抽象フォームではデータの大きさに対する制限が考えられていないため、オーバーフローしたデータを表わすフォームを必要とすることもある。この場合 M はオーバーフローデータ用フォームへの対応も定義する。

(2) 1つの抽象フォームに対して複数個のフレームが用意さ

れることがある。たとえば、論文を表現する抽象フォームは一種のみでよいが、論文の種類(レポート、学術誌内の論文、本の中の論文、本)によって異なるフレームが用意されることがある。

上記(2)の場合に、動的フレームの定義を一般的にすれば、一つの動的フレームで表わせるようにできるともいえる。

写像 P にも、次のような種類がある。

(1) 固定写像

(2) 可変写像

固定写像は、抽象フォーム中の構造表現の中に現われる各属性に対して、対応する変数領域を決めているものである。

可変写像は、ある属性(集合)の値によって写像が変化するものである。上記(2)の例でいえばTYPE という変数が論文の種類を示しているとするとその値によって異なるフォームへ写像されたり、異なる変数領域に写像されたりする。

一つの変数領域におけるデータの位置も、データの値によって決められることがある。ある属性がデータの順序を示しているとする、この属性値を用いてデータを並べることができる。グラフのようにAとBの値によってプロットされる点の位置の決まるようなものもある。

フォームには、識別子が付けられるのが普通である。これ

は、 S の中で*や+の付いていない属性の集合の一部に対応して定義される。文献に対するフォームでは、標題TITLEをフォーム識別子とすることができる。フォーム識別子は、フォームの先頭部分に置かれるのが普通である。いわゆる、関数従属性を考えた場合のキーとは異なるが、識別子の値が同じフォームが2つ以上存在する可能性はないものとする。

抽象フォームに対して、その値に対応させたものを抽象フォームのインスタンスと呼ぶ。

3. 抽象フォームの変換

フォームの数学的性質に注目する場合は、データのオーバフローやレイアウトの問題のない抽象フォームで扱うのがよい。以下では主として抽象フォームを扱うので混乱のない限り抽象フォームをフォームと呼ぶ。

抽象フォームには、次のような変換が可能である。

$$P1: \omega_1 (\omega_2 \cup \omega_3) = \omega_1 \omega_2 \cup \omega_1 \omega_3$$

$$P2: (\omega_1 \cup \omega_2)^* = \omega_1^* \cup \omega_2^*$$

$$P3: \omega_1 \omega_2 = \omega_2 \omega_1$$

$$P4: (\omega^*)^* = \omega^*$$

$$P5: \omega = \omega \omega$$

$$P6: \omega^+ \Rightarrow \omega^*$$

$$\left. \begin{array}{l} P7: \omega' \Rightarrow \omega \\ P8: \omega \Rightarrow \omega^* \end{array} \right\} \omega' \Rightarrow \omega^*$$

$$P9: \omega' \Rightarrow \omega^+ \quad \Rightarrow: \text{変換可}$$

$$P10: \omega_1^* \omega_2^* \Rightarrow (\omega_1 \omega_2)^* \quad =: \Rightarrow \text{か} > \Leftarrow$$

$$P11: \omega_1 (\omega_1 \omega_2)^* = \omega_1 \omega_2^*$$

$$P12: \omega_1^* (\omega_1 \omega_2)^* = (\omega_1 \omega_2)^*$$

$$P13: (\omega_1 \omega_2)^* = (\omega_1 \omega_2^*)^*$$

$$P14: \omega_1^* \omega_2^* \dots \omega_n^* = (\omega (\omega_1 - \omega)^* (\omega_2 - \omega)^* \dots (\omega_n - \omega)^*)^*$$

P15: 関数従属性 $X_1 \rightarrow X_2$ が成立する場合.

$$\omega_1 (\omega_2 \omega_3)^* = \omega_1 \omega_2 \omega_3^*$$

P16: 結合従属性 $*[X_1, X_2, \dots, X_n]$ ($X = \cup X_i$) が成立する場合.

$$\omega^* = \omega_1^* \omega_2^* \dots \omega_n^*$$

4. 制約および索引条件を用いたフォームの設計

前節で示した抽象フォームの変換を用いて、制約条件や索引条件を考慮したフォームの設計法について検討する。

【従属性を利用したフォームの設計】

- (1) 結合従属性が成立すると P16 を用いて分解する。
- (2) 関係識制子が ω であるとする ω と上記の表現を結合し、 ω 以外の部分より ω に含まれる属性を取り去る。
- (3) 関数従属性を用いて簡単化する (P15)。

(4) 結果に $P(2)$ を適用する。

(1) で得られた表現を W_1 とする。フォーム識別子が W であるので、まず $W W_1$ を得る。 W に属している属性は集合とにならないので、例えば A が W に属していると $P(1)$ により W_1 の中の A をすべて除くことができる。このことにより (2) が正しいことが判る。

適用例を次に示す。

[例1] 属性集合 $A B C D E F G$ に対し、結合従属性 $*$ [ABC, DE, ABG, EFG] と関数従属性 $A \rightarrow G$ が成立しており、 A がフォーム識別子であるとする。

(1) 結合従属性により

$$(ABC)^* (DE)^* (ABG)^* (EFG)^*$$

(2) A がフォーム識別子であるため、前に A を付けて上の表現内の A を取り、たものと連接する。

$$A(BC)^* (DE)^* (BG)^* (EFG)^*$$

(3) 関数従属性 $A \rightarrow G$ により次の表現を得る。

$$AG(BC)^* (DE)^* B^* (EF)^*$$

(4) $P(2)$ により $(BC)^* B^*$ より B^* が取れる。

$$AG(BC)^* (DE)^* (EF)^*$$

索引条件の反映は、次の3つの条件による。

a. 索引にするべき属性の順位が付けられているときそれを反映する。

b. 属性の値の種類に注目し、少ないものを優先的に索引とする。

c. 関数従属性に注目する。

d. 共通に索引とできる属性を優先する。

[索引付けの方法]

$(A_1, A_2, \dots, A_n)^*$ に対する索引付けの方法

a. 利用者によって与えられた索引付けのための属性の優先順位が $A_1, A_2, \dots$ の順とする。このときまず $(A_1, (A_2, \dots, A_n)^*)^*$ を得、次に $A_1, (A_2, (\dots (A_{n-1}, A_n^*) \dots))^*$ を得る。

b. A_i の実際に取る値の数の平均を d_i とする。 $A_1, A_2, \dots$ は d_i の小さい順に並んでいるとすると、この順に索引付けすると結果の冗長度が減る。

c. 関数従属性 $A_1 \rightarrow A_2$ があれば $d_2 \leq d_1$ が成立する。このため A_2 で索引付けした方が冗長度が減る。

d. $(A, W_1)^* (A, W_2)^*$ の場合には P 14 により $(A, W_1^* W_2^*)^*$ とできる。

これらの条件間のトレードオフを扱う必要がある。実際的と考えられる方法は次の通りである。

【索引付けのための優先順位の決定】

(1) 各属性の取る値の数の平均、関数従属性、利用者の指定する索引付けのための属性の優先順位を利用して、各属性 A_i に重み α_i を対応させる。

(2) 与えられた構造表現に対し、属性 A_i が k ヶ所で共通に索引付けできるなら α_i/k を属性 A_i の重みとする。

(3) 重みの小さいものより優先的に索引付けする。

上記 (1) の重み α_i が A_i の実際に取る値の数の平均である場合、 k 個の索引と共用できると α_i/k が冗長性に対応する。属性の取る値の数に対する情報が得られない場合は次のような方法をとる。

【構造的情報のみを用いた索引付けのための優先順位】

(1) 共通に索引付けできる属性のうち、できるだけ多くの部分で共有できるものを優先する。

(2) 関数従属性も利用する。

(3) 残った部分については利用者の選択による。

【例2】例1の結果で、 $C \rightarrow B$ と $E \rightarrow F$ という2つの関数従属性が成立していたとする。

$$AG(BC)^*(DE)^*(EF)^*$$

E が2ヶ所に表われているため E で索引付けを行なう。

$$AG(BC)^*(ED^*F^*)^*$$

$C \rightarrow B$ により、 BC の部分は B で索引付けする。

$$AG(BC^*)^*(ED^*F^*)^*$$

$E \rightarrow F$ により才3項はP15により $(EFD^*)^*$ となる。ここで
Fによる索引付けを行う。

$$AG(BC^*)^*(F(ED^*)^*)^*$$

5. フォームに対する演算

すでに存在するフォームに演算をほどこした場合に、どのようなフォームが得られるかについて検討する。

フォームに対してほどこされる演算としては関係代数演算が考えられる。関係演算として、集合演算、射影、制約、結合が基本的である。特に問題となるのは次の3つの演算である。

(1) 和集合演算

(2) 共通集合演算

(3) 結合演算

抽象フォームに対する射影は次のようになる。

【フォームに対する射影演算の適用】

(1) 与えられた構造表現 S を属性集合 X 上に射影するものとする。 S を索引部分を除いたスター・ハイト1の表現にする。

(2) X に含まれない属性で1度しか現われないものは消す。
 (3) X に含まれない属性が2ヶ所以上に表われていると、この属性を含む $*$ によるブロックをまとめた後にその属性を消去する。

(4) 上記(2)と(3)の適用により X に含まれない属性をすべて除いた後、索引操作を行なう。

上記(4)においてもともと成立していた従属性制約が成立しなくなっていることがあるため、もとのフォームと同じような形には復元できないこともあり得る。実際に属性を消去すると表現の冗長性が生じることもあり、射影で消去された属性値を符号化して利用した方がよい場合もある。

たとえば、 $(AB)^*(BC)^*$ を AC 上に射影すると $(AC)^*$ が得られる。これは A と C の1対1対応としてすべての組合せを覚える方法であるが、元の表現では B で索引付けすると冗長度が減るため、 B の実際の値を出さずに符号化した値を用いて表現の冗長度を減らしうる場合がある。

【フォームに対する制約演算の適用】

(1) 属性 A と定数 C に対し、 $A = C$ を適用した場合は、 A を与えられた表現と接続し単純化を行なう(他の A を消去する操作に対応)。

(2) 属性 A と属性 B に対し、 $A = B$ を適用した場合は、 B を

Aで置き換えた後、簡単化を行なう。

2つの抽象フォームが与えられた場合、この和集合演算の結果得られるフォームは、どちらのフォームに示された内容も表現できるようなフォームを、和集合演算($\cup$)なしに表現するものとなる。すなわち、 F_1 と F_2 という2つのフォームが与えられた場合、 $F_1 \Rightarrow F$ かつ $F_2 \Rightarrow F$ となるような最も簡単なフォーム F を求める問題となる。

【フォームに対する和集合演算の適用】

- (1) 与えられた2つの表現をスター・ハイト1の表現に変換する。
- (2) ある属性が片方の構造表現で A 、もう1つの構造表現で $(A \dots)^*$ の形で表われていると A を消去する。
- (3) ω_1^* と ω_2^* とがある、 ω_1 と ω_2 に対応する属性集合 X_1 と X_2 について、 $X_1 \supseteq X_2$ なら ω_2 を消去する。
- (4) 上記(2)(3)を適用不能になるまで適用し、その後、従属性や索引付けによる簡単化を行なう。

【例3】次の2つの構造表現が与えられたとする。

$$S_1 = GF(ABC)^*(C(AD)^*E^*)^*$$

$$S_2 = G(ABCE)^*(DA^*C^*F^*)^*$$

- (1) 上記の S_1 と S_2 は、次のように変形される。

$$S'_1 = GF(ABC)^*(CAD)^*(CE)^*$$

$$S'_2 = G(ABCE)^*(DA)^*(DC)^*(DF)^*$$

(2) Fが条件を満足するため消去する。

(3) $(ABCE)^*$, $(CAD)^*$, $(DF)^*$ および Gが残る。

(4) 結果として、次の構造表現を得る。

$$G(ABCE)^*(CAD)^*(DF)^*$$

元の S_1 や S_2 では C や D が索引付けに用いられているので、C を選ぶことにする。

$$G(C(A(BE)^*D^*)^*)(DF)^*$$

2つのフォームの内容をまとめて同時に表現するのは結合演算である。

【フォームに対する結合演算の適用】

(1) 与えられた構造表現 S_1 と S_2 に対し、 S_1, S_2 を求める。

(2) S_1, S_2 を簡単化する。

結合演算は結果に結合従属性を生成するだけであとはフォームの設計問題に帰着できる。最後に2つのフォームの共通データのみが表現できるようなフォームの論理積問題を考える。

【フォームに対する共通集合演算の適用】

(1) 与えられた構造表現を S_1, S_2 とし、この両方に共通に表

われる属性の集合を X とする。

(2) S_1 と S_2 を X に射影した結果を S'_1 , S'_2 とする。

(3) S'_1 , S'_2 に対するフォームの設計を行なう。

謝辞 日頃御指導いただき本学矢島修三教授に深謝します。

本研究に先立つ関連したデータベースに関する研究論文

[KAMBL 8111] [KAMBT 8201] [KAMBT 8311] の共著者

の方々、特に神戸大学田中克巳氏(現在南カリフォルニア大)

米国HDR社L・ヴィエト・チュン氏ならびに日本IBM武田浩

一氏に感謝します。

参考文献

- [CODD 7006] Codd, E.F. "A Relational Model of Data for Large Shared Data Banks", CACM, Vol.13, No.6, pp.377-387, June 1970
- [FAGIM8011] Fagin, R., Mendelzon, A.O. and Ullman, J.D. "A Simplified Universal Relation Assumption and its Properties", IBM Technical Report, RJ2900, Nov. 1980
- [JAESS8203] Jaeschke, G. and Schek, H.-J. "Remarks on the Algebra of Non First Normal Form Relations", Proc. of ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, pp.124-138, March 1982
- [KAMBL8111] Kambayashi, Y., Le Viet, C., Tokuda, S. and Yajima, S., "A Database Preparation System", COMPSAC, pp.335-347, Nov. 1981
- [KAMBT8201] Kambayashi, Y., Tanaka, K., Takeda, K. and Yajima, S. "Representation of Relations for Database Output Utilizing Data Dependencies", Proc. of 15th Hawaii ICSS, pp.69-78, Jan. 1982

- [KITAK8111] Kitagawa,H., Kunii,T.L. and Ishii,Y. "Design and Implementation of a Form Management System APAD Using ADABAS/INQ DBMS", COMPSAC, pp.324-334, Nov. 1981
- [KOWAL8206] Kowarski,I. and Lopez,M., "The Document Concept in a Database", Proc. ACM SIGMOD, pp.276-283, June 1982
- [LUO-Y8104] Luo,D. and Yao.S.B. "Form Operation By Example - a Language for Office Information Processing", Proc. of ACM SIGMOD Int. Conf. on Mangement of Data, pp.212-223, April 1981
- [SHU-H7510] Shu,N.C., Housel,B.C. and Lum,V.Y. "CONVERT:A High Level Translation Definition Language for Data Conversion", CACM, Vol.18, No.10, pp.557-567, Oct. 1975
- [ZLOO 8103] Zloof,M.M. "QBE/OBE: A Language for Office and Business Automation", IEEE COMPUTER, Vol.14, No.5, pp.13-22, March 1981

[KAMBT8311] 上林, 田中, 武田, 矢島, "関係データベースにおける意味制約を反映した非正規形の関係の設計問題" 情報処理論文誌, 83年11月.