

リレーショナルモデルの統合性 に関する一検証法

中野良平・長岡満夫
(日本電信電話公社・横須賀電気通信研究所)

1. はじめに

データベースは現実世界の事象をモデル化したものであるが、事象は個々の物とそれらが持つべき関係を自然にあるいは意味論的にとらえることが望ましい。これは、データモデルの基本構成要素として、少なくとも構造 (structure)、操作 (operation)、統合性 (integrity) を備えている必要があるという考え方にも現れている [1]。

従来、統合性に対する意味論はデータベース利用者側 (業務プログラム等) に委ねられてきた。しかし、データベースの外部で多元的に意味を扱う為、保守性の低下、食い違いによるデータベース全体の無矛盾性が不完全になること等 [2] により、データモデル論では意味を統合性として表現する研究が行われている [3,4]。

本稿では、データモデルとしてリレーショナルモデルを対象として、第2節以降で (i) 意味表現を強化したリレーショナルモデルの提示、(ii) 無矛盾性の定義と無矛盾性の保証アルゴリズム、(iii) 矛盾発生時の矛盾箇所検出の考え方とその方法、について述べる。

2. 意味表現を強化したリレーショナルモデル

本節では、一階述語論理を用いて意味表現を強化したリレーショナルモデルを提示する。

2.1 構造

現実世界の各事象をグルーピングすることにより、

- (i) グループ内の各メンバは同一の構造を持つ、
- (ii) グループ内の各メンバは複数当該グループ内に存在しない、
- (iii) 異なるグループの各メンバは比較可能な要素値を持つ、

を満足するようなグループをリレーションと呼び、リレーションのメンバである事象をタプル、タプルの値集合をリレーション値と呼ぶ。また、リレーションの特性を記述したものをリレーションスキーマと呼び、リレーションスキーマはリレーション構造とリレーション制約からなる。

リレーション内の各タプルは属性と呼ぶ同数の要素から成る。各属性にはその属性が取りえる値集合が対応し、この値集合のことを定義域 (domain) と呼ぶ。

2.2 操作

操作はリレーションを導出する query とリレーション値を変更する insert, delete, replace からなる。

query は大きく relational algebra と calculus ベースのものに分類される [5]。現在、後者を研究中であるが本稿とは直接関係しないため省略する。

2.3 統合性

統合性とは適用環境に於けるリレーショナルデータベース内の各タプルが満足すべき意味論的制約である。

統合性制約 (integrity constraints) は以下に述べるように一階述語論理 WFF (Well Formed Formula) で表現する。

比較演算子 θ ($=, <, >, \geq, <=, \leq$) とリレーション構造を atomic formula とし、WFF は以下のように再帰的に定義する。

1. 全ての atomic formula は WFF である。
2. F が WFF ならば、 $\sim F$ は WFF である。
3. F_1 と F_2 が WFF ならば、 $F_1 \vee F_2$, $F_1 \wedge F_2$, $F_1 \rightarrow F_2$ は WFF である。
4. F が WFF であり、 x が F 内の自由変数ならば、 $(\exists x) F$ ($\forall x) F$ は WFF である。
5. 1.~4. で定義される WFF だけが WFF である。

2.4 具体例

以下の統合性検証法で用いる リレーションと統合性制約の例を各々図 1., 図 2. に示す。

EMP (e-name, salary, d-name)
MGR (m-name, d-name, numb-of-subord)
DEPT (d-name, floor)
COMP (c-name, location)
SUPPLY (c-name, d-name, i-no, vol)
(a) リレーション構造例

図 1. リレーショナルデータベース例 (1/2)

EMP	e-name	e-salary	d-name	< — 属性
	NAME	MONEY	DNAME	< — 定義域
	SATO	510	TOY	< — タプル
	NAGASAKI	280	BOOK	
	NISHI	650	DRESS	
	HOSHI	430	DRESS	
	TANAKA	370	BOOK	
	SHRAI	400	TOY	

(b) EMP リレーション

MGR	m-name	d-name	numb-of-subord
	NAME	DNAME	NUMBER
	SATO	TOY	1
	TANAKA	BOOK	1
	NISHI	DRESS	1

(c) MGR リレーション

図 1. リレーショナルデータベース例 (2/2)

① $INTEGRITY (\forall x)EQ (COUNT(y/DEPT(x,y),1))$.
部門は必ず 1つの階になくなくてはならない。

② $INTEGRITY (\forall x)(\sim SUPPLY(x, @, GUN, @) \vee SUPPLY(x, @, BULLET, @))$.
銃を提供している会社は必ず弾丸を提供している。

③ $INTEGRITY (EQ (COUNT(x/(\forall y)(SUPPLY(y, @, x, @) \wedge ITEM(x,A))),1))$.
どの会社も提供する,Aというタイプの品目は唯一つである。

④ $INTEGRITY ((\forall x)(\exists y)(\exists z)(MGR(x,y,@) \wedge EMP(x,@,z) \wedge EQ(y,z)))$.
管理者の管理する部門は自分が従業員として属する部門と等しい。

⑤ $INTEGRITY (GT (AVG(x/EMP(@,x,DRESS)), AVG(y/EMP(@,y,TOY))))$.
洋服部門の従業員の給与の平均は、玩具部門の従業員の給与の平均より多い。

⑥ $INTEGRITY ((\forall x)(\exists y)(\sim MGR(x,@,@) \vee (EMP(y,@,@) \wedge EQ(x,y))))$.
管理者は従業員でなくてはならない。

図 2. 統合性制約の例

3. 検証メカニズム [6]

3.1 無矛盾性の定義

リレーショナルデータベースを WFF の集合と解釈すると、統合性制約の評価に際して Closed World Assumption あるいは Open World Assumption のいずれかを選択する必要がある [7].

リレーショナルデータベース中に存在しないタプルに対応する WFF の真偽値は、Closed World Assumption では偽となり Open World Assumption では unknown の扱いとなる。

通常の適用領域では Closed World Assumption が一般的であるため、以下では Closed World Assumption を採用し、無矛盾性を以下のように定義する。

(定義) リレーショナルデータベースと統合性制約が無矛盾である必要十分条件は、リレーショナルデータベースと Closed World Assumption の両方にもとずく解釈 (interpretation) に対して、統合性制約が真として評価されることである。

3.2 検証の分類

3.1 で述べた定義にもとずいた無矛盾性の検証をここでは単に統合性検証と呼ぶ。

統合性検証は検証の適用に対応して以下に述べる全検証と逐次検証の2種類に分類される。

(1) 全検証

全検証とは、リレーショナルデータベースと統合性制約の両方が全体として無矛盾であるか否かを検証することであり、データベースの初期設定時、2つ以上のデータベース統合時、周期的な検査時等に行われる。

(2) 逐次検証

逐次検証とは、既に存在するリレーショナルデータベースと統合性制約は無矛盾であると仮定し、あるトランザクションを実行した後依然としてリレーショナルデータベースと統合性制約が無矛盾となるか否かを検証することである。

逐次検証は全検証の部分作業であるので、以後は一般的な全検証について統合性検証アルゴリズムを述べる。

3.3 検証アルゴリズム

全検証は個々の統合性制約の統合性検証に分割できる。それは、統合性制約の全体集合が真である必要十分条件は個々の統合性制約が真であることに依る。

従って以下では個々の統合性制約の検証アルゴリズムを述べる。

step 1 評価空間 (evaluation space) の作成

1.1 WFF を prenex normal form に変換する。
prenex normal form は、quantifier と quantified variable から成る prefix と WFF の残りの部分である matrix からなる。aggregate function 内の quantifier と quantified variable はその aggregate function の外側に出ることはなく、matrix 内に含まれる。

1.2 評価空間の作成

評価空間は prefix 内の変数 i に対応する定義域を D_i とするとき、それらの cartesian product

$$D_1 \otimes D_2 \otimes D_3 \otimes \dots \otimes D_n$$

で表現される。

step 2 評価空間内の各要素の評価

評価空間内の要素に対し、step 2を繰り返す。

2.1 aggregate function の評価

aggregate function 内に aggregate function は含まないものと仮定する。(注1)

2.1-1 リレーションの属性が定数かあるいは評価空間内の要素である場合は、リレーションの属性値を指定した selection を行う。

2.1-2 2つ以上のリレーション内に同一の変数が指定され、かつこれらのリレーションが '∧' で結ばれている場合は、natural-join を行う。

2.1-3 2つ以上のリレーション内の変数が比較演算子 θ で結ばれ、かつこれらのリレーションが '∧' で結ばれている場合は、theta-join を行う。

2.1-4 同一のリレーションに2つの変数が比較演算子 θ で結ばれている場合は、theta-selection を行う。

2.1-5 prefix内の変数のquantifierの '∨' (又は '∃') に対応し、リレーションの division (又は projection) を行う。

2.1-6 aggregate function が統計関数の場合には直ちに step 2.1-8へ、そうでない場合は求めたい変数に対応する属性に関してリレーションの projection を行う。

2.1-7 リレーション間の結合記号 (∨, ∧等) に対応し以下の操作を行う。

∨ (or) → ∪ (集合のunion)

∧ (and) → ∩ (集合のintersection)

∧~ (and not) → − (集合のdifference)

~ (not) → $\bar{\Phi}$ ~

最後のマッピングは $\wedge \sim \rightarrow -$ のマッピングが適用できない時のみ行う。但し、 $\bar{\Phi}$ は求めたい変数の定義域を表す。

2.1-8 リレーション又は集合の値から関数値を求める。

(注1) aggregate function のネストが存在する場合には、ネストの内部から本節と同様の手順で処理すれば良い。

2.2 命題論理の評価

prenex normal form のmatrixを命題論理 (propositional logic) に従って真 (又は偽) の評価を行う。

step 3 評価の統合

3.1 評価表 (evaluation table) の作成

表の各行を評価空間の要素とし、また列にはprefixの変数を順に割当て、最右列にはstep 2で得た評価空間の要素の真偽値の列を付け加えた評価表を作成する。

3.2 評価の統合

評価表の右側から左側へと評価していく過程において、

(i) 評価表の列に対応するquantifierが \exists ならば、直ぐ右列の評価値の中に少なくとも 'T' (真) が1 つ見つければ、 'T' を評価列の該当位置に設定する。そうでなければ 'F' (偽) を設定する。

(ii) 評価表の列に対応するquantifierが \forall ならば、直ぐ右列の評価値の中に少なくとも 'F' (偽) が1 つ見つければ、 'F' を評価列の該当位置に設定する。そうでなければ 'T' (真) を設定する。

(i), (ii) に於いて、留意すべきことは、評価空間の全ての要素に対して評価を行う必要はない点で、その点を考慮すれば評価作業の削減が可能となる。

x	y	z
SATO	BOOK	BOOK
SATO	BOOK	DRESS
SATO	BOOK	TOY
SATO	DRESS	BOOK
SATO	DRESS	DRESS
SATO	DRESS	TOY
SATO	TOY	BOOK
SATO	TOY	DRESS
SATO	TOY	TOY
TANAKA	BOOK	BOOK
TANAKA	BOOK	DRESS
~ ~ ~ ~		
NISHI	TOY	DRESS
NISHI	TOY	TOY

図 3. 評価空間

図 1. のリレーショナルデータベースと図 2. の統合性制約④に上記のアルゴリズムを適用する。

quantified variable x, y, z の各定義域は、

$$D_x = \{ \text{SATO, TANAKA, NISHI} \}$$

$$D_y = D_z = \{ \text{TOY, BOOK, DRESS} \}$$

となり、評価空間は図 3. のようになる。

評価表は図 4. のようになり、この例では図 1. のリレーショナルデータベースと図 2. ④の統合性制約は無矛盾であることが分かる。

なお、図 4. の中でハッチングを入れた要素に対する評価結果は統合に使用しておらず、評価作業が省力化できることを示している。

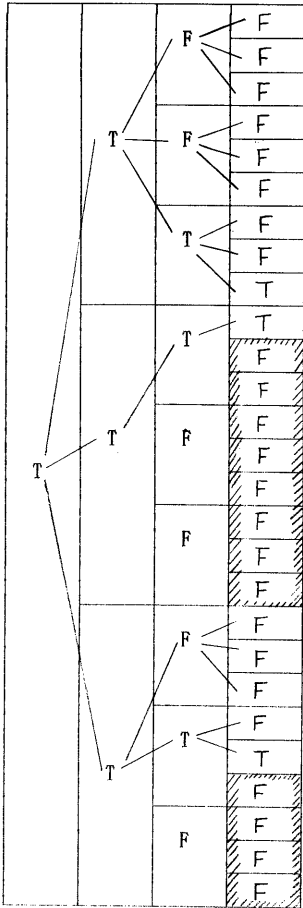


図 4. 評価表

4. インパクト分析 [8]

インパクト分析とはリレーショナルデータベースと各統合性制約に対し3節で述べた検証を行った結果、統合性評価結果が偽となった場合に偽に至らしめたリレーショナルデータベースのタプル或いはaggregate function を求めることである。

なおここで統合性制約そのものは常に正しいものと仮定する。

4.1 インパクト分析のための基本概念

(1) E 空間とF 空間

インパクト分析で使用する評価表の部分集合を E 空間と呼び、リレーショナルデータベース内のタプルあるいはaggregate function を対象とした空間を F 空間と呼ぶ。

図 1. のリレーショナルデータベースは図 2 ④の統合性制約に対して真と評価されたが、ここでは矛盾を生じさせるためリレーショナルデータベースを図 5 のように変え、図 2 ④の統合性制約を評価する。

統合性検証結果は図 6. に示すようになるが、図中に、統合性制約を偽に至らしめる E 空間の要素を丸印で示す。

EMP		
SATO	510	TOY
NAGASAKI	280	BOOK
NISHI	650	DRESS
HOSHI	430	DRESS
TANAKA	370	BOOK
SHRAI	400	TOY
TACHI	530	DRESS
SUZUKI	390	BOOK

(a) EMP リレーション

MGR		
SATO	TOY	1
TANAKA	DRESS	2
NISHI	BOOK	2

(b) MGR リレーション

図5. リレーショナルデータベース例

(2) AND 責任 と OR 責任

統合性制約の検証結果を偽に至らしめる E (又は F) 空間の要素 e (又は f) を抽出していく過程において、ある e (又は f) の評価結果を真に至らしめるためには、その直前の e (又は f) の要素を全て修正する必要がある場合、その直前の e (又は f) は AND責任にあると呼ぶ。即ち、図 6. に於いて要素 2, 3は要素 1 に対しAND 責任にあると言う。

また、直前の e (又は f) のいずれかを修正すれば良い場合、その e (又は f) はOR責任にあると呼ぶ。

図 6. に於いて要素 4, 5, 6 は要素 2 に対しOR責任にあると言う。

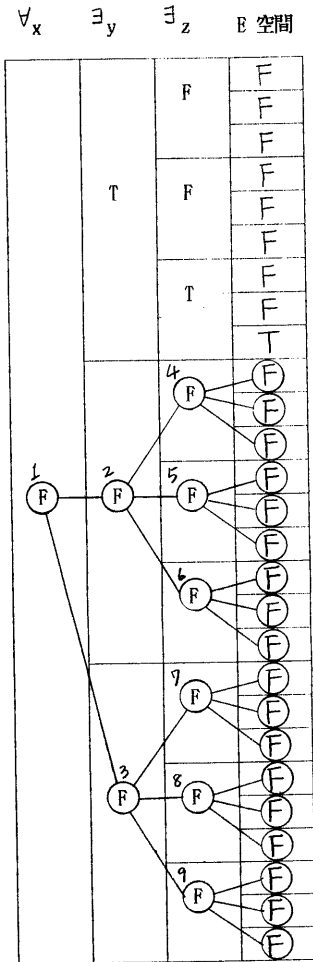


図 6. 図5.のリレーショナルデータベースに対する統合性制約④の検証結果

(3) 責任の重み (w)

AND 責任, OR 責任の定量的表現として, AND責任, OR 責任にある E (又は F) 空間の要素に以下のように重みを定義する。

- (i) AND 責任のとき, $w = 1$
- (ii) OR責任のとき, $w = 1/n$

但し, nはOR責任にあるE (又はF) 空間の要素数である。

図 6. に示す例に於いて, OR責任にある要素 4, 5, 6 の重みは各々 1/3 となり, AND責任にある要素 2, 3 の重みは各々 1 となる。

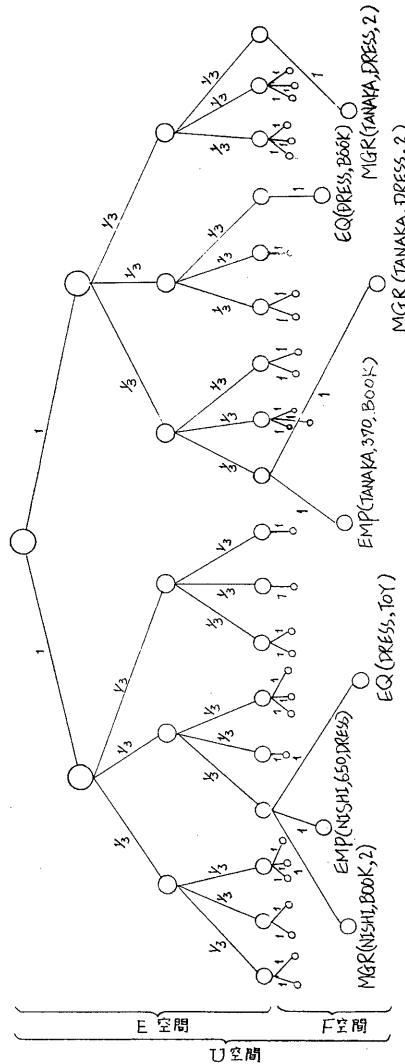


図 7. 責任グラフ

(4) 責任 グラフ (liability graph)

統合性制約を偽に至らしめる評価表の各要素をノードに対応させ、ノード間を結ぶ線分を辺に対応させ、更に各辺に責任の重み (w) を付与したグラフを責任グラフと呼ぶ。

図 6. に示す例の責任グラフは図 7. の E 空間に対応したグラフになる。

(5) U 空間

責任 (AND 責任, OR 責任), 責任の重み (w) 及び責任グラフの基本概念は, 表 1. に示すように E 空間, F 空間に於いて統一的に扱うことができる。

即ち, E 空間 と F 空間を統合した統合空間 (U 空間と呼ぶ) に於いてインパクト分析を考察することができる。

表 1. E 空間と F 空間の類似性

空間種別 基本概念		E 空間	F 空間
		AND 責任	'∨'
責任	OR 責任	'∃'	'∀'
責任の重み	AND 責任	1	1
	OR 責任	$\frac{1}{n}$ (nは責任グラフのノードから出る辺の数)	$\frac{1}{n}$ (nは責任グラフのノードから出る辺の数)
責任グラフ		評価表の最左端の F をrootとし、それに関連するE空間の各要素をleafとするグラフ	評価表の最右端の各要素をrootとし、リレーションのタプル, aggregate function, 比較演算子をleafとするグラフ

(6) E 空間 (又は U 空間) のインパクト指標

修正対象のE空間 (又はU空間) の要素は実際には全てを修正する必要はなく, 例えば図 7. に於いて E 空間のいずれかの要素を 2つ修正すれば良いことが分かる。

従って修正対象の E 空間 (又は F 空間) の要素数 (I_d) と実際に最小限修正するべき E 空間 (又は F 空間) の要素数 (I_e) が存在する。I_dを被疑インパクト指標, I_eを実効インパクト指標と呼び, これらによりインパクト分析の定量的評価が可能となる。

I_e は責任グラフにおいてleaf (i = 1, ..., k) 毎に rootから各leafに至る辺の重み (w) の積 $\prod_{j=1}^i w_{i,j}$ を求め, 各leafの総和により表すことができる。

即ち,

$$I_e = \sum_{i=1}^k \left(\prod_{j=1}^i w_{i,j} \right)$$

となる。

図 7. における E 空間 (又は U 空間) の I_e は以下のようなになる。

$$I_{e(e)} = 1/9 * 18 = 2$$

$$I_{e(w)} = 1/9 * (1+3+3) + 1/9 * (1+1+2)$$

$$1/9 * (2+3+2) + 1/9 * (1+1+2)$$

$$1/9 * (3+1+3) + 1/9 * (3+2+2)$$

$$= 4$$

また, 図 7. における E 空間 (又は U 空間) の I_d は I_{d(e)} = 18, I_{d(w)} = 36 である。}}

4.2 インパクト分析手法

図 7. に示した責任グラフからも明らかなように統合性制約を偽に至らしめるタプルあるいはaggregate function は一意に決定されるものではない。AND条件あるいはOR条件で結ばれたタプルあるいはaggregate function の CP (Correction Perscription) として提示される。

CP は4.1 で述べた責任グラフを変換することにより得られる。

(責任グラフからCPへの変換)

責任グラフの任意のノードにおいて, そのノードに至る辺の重み (w) は E 空間においてはquantifier の種類, U 空間においては∧, ∨ の付与の仕方により全て等しいことに留意すると, 以下の変換規則が成立する。

即ち, 責任グラフの任意のノードn_iに至るノードをu₁, u₂, ..., u_mとすると,

$$W = 1 \text{ の場合のCP表現は, } CP_{n_i} = u_1 \wedge u_2 \wedge \dots \wedge u_m$$

$$W = 1/m \text{ の場合のCP表現は, } CP_{n_i} = u_1 \vee u_2 \vee \dots \vee u_m$$

となる。

n_iを責任グラフのrootとすることにより全体のCPを得ることができる。

例えば, 図 6. に示した統合性制約の検証を偽に至らしめた修正対象のCPは,

$$CP = (H \vee I \vee J) \wedge (K \vee L \vee M)$$

となる。

但し、ここで

- A = MGR (TANAKA, DRESS, 2)
- B = EMP (TANAKA, 370, BOOK)
- C = MGR (NISHI, BOOK, 2)
- D = EMP (NISHI, 650, DRESS)
- E = EQ (BOOK, DRESS)
- F = EQ (BOOK, TOY)
- G = EQ (DRESS, TOY)

- H = A \vee (A \wedge B \wedge E) \vee (A \wedge B \wedge F)
- I = E \vee B \vee (B \wedge G)
- J = (A \wedge F) \vee (A \wedge B \wedge G) \vee (A \wedge B)
- K = D \vee E \vee (D \wedge F)
- L = (C \wedge D \wedge E) \vee C \vee (C \wedge D \wedge G)
- M = (C \wedge D \wedge F) \vee (C \wedge G) \vee (C \wedge D)

とする

5. おわりに

本稿では、意味表現を強化したリレーショナルモデルを提示し、無矛盾性の定義と無矛盾性の保証アルゴリズム、及び矛盾発生時の矛盾箇所検出の考え方とその方法について具体的な例を用いて述べた。

今後は本稿で示した考え方にもとづいた試作・評価、及びより詳細な検出アルゴリズムの研究（特に F 空間について）が必要である。

(参考文献)

- [1] Codd, E.F., "Data Models in Data Base Management," ACM SIGMOD Record, 11, 2, Feb. 1981.
- [2] Fernandez, E.B., Summers, R.C., and Wood, C., "Data Base Security and Integrity," Addison-Wesley, 1981.
- [3] Hammer, M.M. and McLeod, D.J., "Semantic Integrity in a Relational Data Base System," Proc. 1st VLDB, Framingham, 1975.

- [4] Stonebraker, M., "Implementation of Integrity Constraints and Views by Query Modification," ACM SIGMOD Int. Conf. on Management of Data, San Jose, May, 1975.
- [5] Codd, E.F., "Relational Completeness of Data Base Sublanguage," in Data Base Systems, Courant Computer Science Symposium 6, Prentice-Hall, 1972.
- [6] Nakano, R., "Integrity Checking In A Logic-Oriented E-R Model," Proc. 3rd E-R Int. Conf., Oct. 1983.
- [7] Codd, E.F., "Extending the Data Base Relational Model to Capture More Meaning," ACM TODS, 4, 4, Dec. 1979.
- [8] 中野, 長岡., "推論を用いたデータデイクショナリインパクト分析に関する一考察," 昭和58年度信学全大, 1584.