データベースシステム 109－7
（1996. 7. 24）

関数制約の文脈的意味

三浦孝夫
産能大学

Abstract

どのようなデータモデルに基づいたデータベース設計であっても、しばしば関数対応に関する一貫性制約が検出される。これまで多くの研究を通じて様々な性質が知られており、特に、関係データモデルでは完全な推論規則が知られている。しかし、この議論には重要な制限がある。つまりよく知られた推論規則では、どの実体型集合上でも高々一つの関連しか表わせない。この条件を充足するために設計の変更が必要となり、一貫性制約として得られた性質をも考察し直さねばならない。

本稿では、データモデリングの立場から、情報の関連を文脈に持つ関数制約を扱う方法を延べ、検出された関数対応のための設計手法を提案する。

On Semantics of Context in Functional Constraints

Takao MIURA
SANNO College

Abstract

Functional constraints are detected very often whatever database models are discussed. These have been widely investigated and analyzed, especially in theory of relational databases,

However, the framework has one important restriction, that is, there can be at most one relationship among entity (or object) types and eventually the scheme should be re-designed for correctly enforcing constraints.

In this work, we introduce the semantics of *context* for functional constraints and we develop a new class of constraints to capture functional properties in real-world.

1 前書き

データベーススキーマ(概念スキーマ)は、情報の構造記述という側面のみならず、一貫性制約記述のための言語である。一貫性制約とは、データベースのとり得る値を制限することで利用者に知識を提供する条件である [5]。代表的に、データベース設計過程では数多くの関数的な対応をとる一貫性制約が論じられてきた。例えば、実体関連 (ER) モデルなどの意味データモデルでは、多対 1 対応などの濃度制約として、関数データモデルでは対応自体がモデル基本機能として、また関係モデルでは関数従属として研究され、設計方法への指針を提供するものとなっている。

関数的な対応は、特に関係モデルで関数従属として深く考察されており、多くの重要な結果が知られている。特に、推論に関する Armstrong の理論的な分析は中心的な役割を果たしている。

ところが、ここでは実体型集合上では高々ひとつの関連しか許されないという強い制限があり、他のデータモデルへの単純な適用を制限する主要な理由になっている。

本稿では、関係モデルでの経緯をふまえ、論理的な観点から、現実世界から抽出された意味をそのまま一貫性制約の考察に利用する枠組みを論じる。第 2 節では、関数従属の性質述べる。また、様々なデータモデルにおける位置付けを要約する。第 3 では Armstrong 推論の限界を示す。第 4 節で、新たな概念、文脈 (context) と、それを用いた関数制約を論じる。

2 関数従属とデータモデリング

関数従属 (functional dependency, FD) は、2 つのデータ (実体) の間の関数対応 (多対 1 対応) を意味し、特に関係モデルを中心として発展した [11]。それ以外にも、実体関連モデルにおける実体型属性は、一貫性制約の代わりにモデルの基本機構としたものとして知られる。

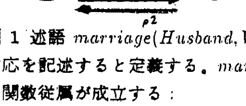
関数従属に関しては多くの重要な結果が知られている。特に、関数従属の推論 (inference) に関する Armstrong の理論的な分析は重要で、効率よい導出方法がデータベーススキーマを独立にする鍵となる [11]。スキーマ R上で関数従属集合 F と関数従属 σが与えらたとき、Fの全てを満たす連想集合 rが必ず σ をも満たすならば、σは Fから含意 (imply) されるという。このことを $F \models \sigma$とかく。関数従属集合 Fから新しい関数従属σを計算する方法は推論則 (inference rule) と呼ばれ、σ は F から導出される、または証明されるという[1]。これを $F \vdash \sigma$ と表す。F から導出された関数従属 σ は論理的に正しい、つまり Fに含まれる関数従属がすべて満たされるような関係では必ずσが成立するとする。この条件を満たすような推論則を、健全 (sound) な推論則という。つまり $F \vdash \sigma$ならば $F \models \sigma$が成り立つときに健全という。これにたいして、Fから得られる論理的に正しい関数従属は常に推論則を用いて得られるとき、その推論則は完全 (complete) であるという。つまり $F \models \sigma$ ならば $F \vdash \sigma$ であるとき完全な推論則という。

Armstrong の推論則が健全で完全であることは良く知られているが、これはデータベース設計で検出された関数従属という一貫性制約が無矛盾性 (inconsistency) であり[2]、しかも独立 (independent) している[3]ことを保証するためである。

Armstrong 推論則は関係モデルだけでなくて他のデータモデルにとっても有用である。例えば、ER モデル [4] では、モデルは実体、関連および属性から構成される。実体型 E に対して属性 A が定義されているとき、A 上の値は E の実体に対して関数的に対応する、直観的に表せば $E \rightarrow A$ が成り立つ。オブジェクト指向データモデルの形式化 [1, 3] では、クラス C のオブジェクトが型 E を持つとき、E は各特性 A からなる順序組として表される。ここでも $E \rightarrow A$ が得られる。AIS モデルでは、型 E の属性 A は実体型で、E との間で全域関数対応する述語 f が存在するものと定義される。

3 Armstrong 推論の限界

これまで関係モデルでは、どのような実体型集合であっても高々ひとつの関数従属しか存在しないとする仮定を明示的または暗黙の内にとってきた。述語定義 p 上の関数従属では、定義型間の関数従属は p の持つ条件と見なされ、関数従属がどうであれ p の意味を越えるもので無い。即ち、述語 $p(E_1, \ldots E_n)$ 上では、任意の定義型集合 $XY \subseteq \{E_1, \ldots, E_n\}$ に対しては高々ひとつの関数従属しか存在しない。このため、関数従属 $\sigma : X \rightarrow Y$ を $p(X, Y)$ と表すこともある。



例題 1 述語 $marriage(Husband, Wife)$ は夫婦対応を記述すると定義する。$marriage$ では次の関数従属が成立する：

$$\sigma_1 : Husband \rightarrow Wife$$
$$\sigma_2 : Wife \rightarrow Husband$$

[1]証明は有限時間で終了するものでないといけない。

[2]すべて充足するデータベースが少なくとも一つ存在する。

[3]一貫性制約の一部が他から計算できるものであってはならないことを表す。

σ_1 は夫に対する妻を、σ_2 は妻に対応する夫を表す。このとき、$\sigma_1 \cdot \sigma_2$ も $\sigma_2 \cdot \sigma_1$ も自己射像 $id(x) = x$ である。

次の関数従属を考える：

$$\rho_1 : Husband \rightarrow Wife$$
$$\rho_2 : Wife \rightarrow Husband$$

ρ_1 は夫に対する母親を、ρ_2 は妻に対応する父親を表すとする。このとき、$\rho_1 \cdot \rho_2$ も $\rho_2 \cdot \rho_1$ も自己射像にならない。これは、*marriage* が夫婦を記述する述語であり親を表すものではないことによる。ρ_1, ρ_2 を表現するには、*marriage* とは別の新たな述語を導入せねばならない。

つまり、関数従属 σ_1, σ_2 は、述語 *marriage* の表す記述を用いて与えられる一貫性制約であるのに対し、ρ_1, ρ_2 は *marriage* と別個に論じるべき述語をも表す。□

ISA の場合、例えば E_1 ISA E_2 のとき、$E_1 \rightarrow E_2$ を得る。E_1, E_2 がそれぞれ属性 A_1, A_2 を持つとき、ISA 特性より A_2 は E_1 の属性と見なせる。実際、$E_1 \rightarrow E_2$, $E_2 \rightarrow A_2$ から推移則 F5 によって $E_1 \rightarrow A_2$ を得る。これは属性継承 (attribute inheritance) と呼ばれる機構である。これを一般化すれば、実体型 E_1, E_2 がそれぞれ属性 A_1, A_2 を持つとき、述語 $p(E_1 E_2)$ で $E_1 \rightarrow E_2$ が与えられていれば、同様の議論によって A_2 を E_1 の属性と見なすことができる。これを述語 p を介した属性継承という。

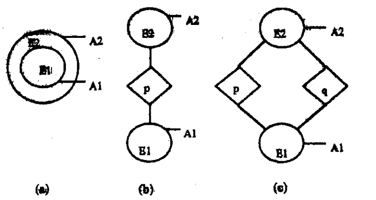


Figure2 Inheritance

ところが、(c) のように、E_1, E_2 上の述語が複数存在し、どれもが $E_1 \rightarrow E_2$ という性質を持てば、$E_1 \rightarrow A_2$ の意味が確定しない。言い替えると、A_2 は E_1 に対して複数の意味対応関係にあり、属性の継承の方法も同時に考えねばならない。この状況を属性の多重継承 (multiple inheritance) という。Armstrong 推論則は多重継承問題を解決しない。なぜなら、E_1, E_2 の間に複数の述語が存在しており考慮の対象外としている。

4　文脈を持つ関数制約

Armstrong 推論は主要な部分を関係モデルに依存している。このため、以下では筆者らの提案するデータモデルを用いて、新たな展開を行う。

4.1　データベーススキーマ

モデル構築のような高度の応用を支援するデータベースを得るためには、情報記述を如何に素直でしかも簡単に行うかがポイントであり、この枠組みはデータモデルと呼ばれる。以下では、本稿を理解するのに必要となる AIS データモデルを簡単な例を用いて、非形式的に述べる。詳細は関連文献を参照されたい [6, 7, 8, 9]。

図 3 では、marriage(夫婦), parent(親子), PreferredMusic(音楽) 情報に関するスキーマが示されている。この図で、円と菱形はそれぞれ実体型、述語を表す。これらの間の辺は述語の構成を意味する。ISA 関連は型記号の間の包含として示される。例えば Man(男性), Woman(女性) は Human(人間) に含まれている。各記号に付けられた名前はその意味を表す。

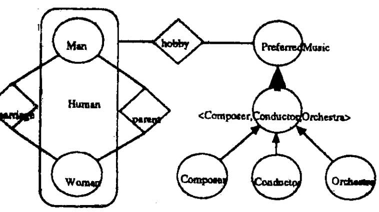


Figure3 AIS diagram

多くの場合、情報は集まって単独の意味を保持し、データ構造がその構成の方法を記述する。この情報の集まりはふたつの意味を持つ。一つはどのような要素から構成されているか、もう一つはどのようにして独自の意味を表しているかである。例えば、図 3 において、PreferredMusic は詳細な構造記述と実体としての認識を同時に表す。後者は、実世界の情報を表す実体として、前者は $\langle Composer, Conductor, Orchestra \rangle$ という複合値 (complex value) として表される。AIS では、複合オブジェクト (complex object) を、実体と複合値の組として定義し、それぞれの側面を同時に記述する基本概念としている。

モデル構築のような高度の応用を支援するには、スキーマやデータを宣言的に記述し、データ操作のみならずスキーマ計算に関する基礎が提供され、しかもデータベース設計をデータモデルの枠組み内で記述する機構が必要である。このためには、データベース記述・操作の宣言性とも対応して、情報を論理的は手法で記述することが自然である。筆者らはデータ論理 (Data Logic) を提案しており、宣言的な操作機構や推

論機構を用いた計算能力を導入し、複合オブジェクトを扱う枠組みを設定する。

例題 2 述語 $marriage$ は Man, Woman で定義されており、$marriage(Man, Woman)$ と表現されている。各連想を $marriage$-連想という。複合オブジェクト PreferredMusic を複合値を強調して記述するならば、次のように表す: $PreferredMusic \equiv \langle Composer, Conductor, Orchestra \rangle$ 。あるいは、述語 $hobby$ において直接記述しても良い: $hobby\ (Human,\ PreferredMusic : \langle Composer, Conductor, Orchestra \rangle)$ □

4.2 文脈

Armstrong 推論が想定する関係モデルでは普遍述語 p を仮定し、その内で成立している関数従属を論じている。特に、与えられた定義型集合間に存在する述語は高々ひとつという制限がある。しかし、複数の述語定義が存在すれば、導出される関数制約は定義型集合の上で何通りも定義される。関数制約集合に'閉路'が含まれていれば、その閉包は一般には無限集合となり計算できない。とくに、'関数従属'を正しく定義できない。以下では、文脈を考察したときの関数対応を考え、これを関数制約と呼ぶ。

例題 3 次の 3 つの関数制約を考える。

(1) $wife$: 男 → 女
(2) $mother$: 男 → 女
(3) $cloth$: 女 → サイズ

この 3 つから '導出' できる関数制約 男 → サイズ は、$wife$ の $cloth$ ((1) と (3)) 及び $mother$ の $cloth$ ((2) と (3)) のふたつの意味がある。

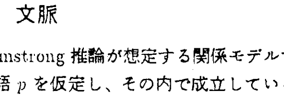


Figure 4 context

一般に、複数の述語に渡る関数制約は、述語を列挙する (導出過程を示す) ことによって初めて意味が確定する。□

例題 4 次の 4 つの関数制約の集合 F を考える。

(1) $wife$: 男 → 女
(2) $mother$: 男 → 女
(3) $husband$: 女 → 男
(4) $father$: 女 → 男

これらから '導出' できる関数制約 男 → 男 は無数の意味がある。つまり、自分自身、妻の父、父、妻の母の父、妻の父の父、父の父、母の父、… 。このような対応は述語名を列挙することで区別できるが、同じ実体型の間の関数制約でありながら、導出結果が無数存在することになる。

関数制約の閉包 F^+ は、単なる定義型の組合わせでなく、述語名列の集合 (一般に無限集合) として与えられる : { 自分自身、妻・父、母・夫、妻・父・妻・父、…}

同様に実体型集合 X の閉包 X^+ も述語名列の付いた型名の集合として表される : 男$^+$ = { 自分自身・男、妻・父・男、母・夫・男、妻・父・妻・父・男、…}

この場合、途中の '経路' は、得られた実体の由来 (文脈) を表している。□

我々が扱うべき問題は、複数の述語が存在する環境で、制約従属の定義を行うことである。この環境を文脈と呼ぶ。文脈 (関数制約が意味を持つ由来) と、その文脈で関数制約が成立することの意味を定める必要がある。

実体型集合 X, Y が各々何かの述語定義に含まれるとすれば、関数制約 $\sigma : X \rightarrow Y$ の X から Y への文脈 (context) α とは次で定義される。X, Y を明示するときは、$\alpha(X, Y)$ とも表し、また必要に応じてカッコを用いる。

述語文脈: 述語 p の XY への射影 $p(X, Y)$ は文脈 p をもつ。X, Y は共に述語 p の定義型集合に含まれる。 $Range$ を p とする。

合併文脈: $\alpha_1(X, Y_1), \alpha_2(X, Y_2)$ が文脈ならばこれらの合併 $[\alpha_1, \alpha_2]$ も X から Y_1Y_2 への文脈である。ただし Y_1Y_2 は $Y_1 \cup Y_2, Y_1 \cap Y_2 = \phi$ を表し、Y_1Y_2 を含む述語定義 p が存在する。$Range$ を p とする。

推移文脈: $\alpha_1(X, Z), \alpha_2(Z, Y)$ が文脈ならば $\alpha_1\alpha_2$ は X から Y への文脈である。$Range$ は $Range(\alpha_2)$ に等しいとする。

文脈 $\alpha(X, Y)$ の定義では、X, Y の各々に対して、X, Y の意味を規定する述語定義が予め与えられていることに注意したい。文脈は、関数制約を直列または並列に並べて定義されるが、'端点' では必ず述語定義で意味を確定する。終点で規定される述語を $Range$ と呼ぶ。

更に、$\alpha(X_1, Y_1)$ が $\beta(X, Y)$ の部分文脈 (partial context)、$\alpha \prec \beta$、とは次で定義される。

(1) α, β は同一で、$X = X_1, Y = Y_1$
(2) β が合併文脈 $[\beta_1, \beta_2]$ ならば、α は β_1 または β_2 の部分文脈である。
(3) β が推移文脈 $\beta_1\beta_2$ ならば、α は β_1 または β_2 の部分文脈である。 または γ_1 が β_1 の接尾部 (つまり β_1 は γ_1 で終了している) で γ_2 が β_2 の接頭部 (つまり β_2 は γ_2 で始まる) のとき推移文脈 $\gamma_1\gamma_2$ も部分文脈である。

例題 5 例 3 で考える。

(1) $wife$: 男 → 女
(2) $mother$: 男 → 女
(3) $cloth$: 女 → サイズ

この 3 つから含意される関数制約　男 → サイズ は、$wife \cdot cloth$(男、サイズ) および $mother \cdot cloth$(男、サイズ) のふたつの文脈で定義できる。
例 4 では 男 → 男 は無数の文脈を有する：

(1) $wife$: 男 → 女
(2) $mother$: 男 → 女
(3) $husband$: 女 → 男
(4) $father$: 女 → 男

これを組み合わせると、id(男、男)、$mother \cdot husband$(男、男)、$wife \cdot father$(男、男)、$mother \cdot husband \cdot mother \cdot husband$(男、男)、$mother \cdot father$(男、男)、$wife \cdot father \cdot mother \cdot husband$(男、男)、$wife \cdot father \cdot wife \cdot father$(男、男)、$\cdots$ と言った文脈が存在する。□

例題 6 航空日誌を考える。各記載に対して、フライト、フライト日時、などが関数的に対応する : $record : Report \rightarrow Flight$
各フライトは機体のメーカとそれに関する情報を対応させている : $manage : Flight \rightarrow Maker$
またフライトにはパイロットに関する情報を記述する述語がある : $drive : Flight \rightarrow Pilot$
機体メーカとパイロットが決定すると、そのメーカの機種の経験年数が対応する : $know : \{Pilot, Maker\} \rightarrow Experience$ 。このとき、航空日誌からフライトの経験年数は次の文脈で決定される : $(record \cdot [manage, drive] \cdot know) : Report \rightarrow Experience$ □

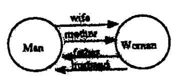


Figure 5 cyclic context

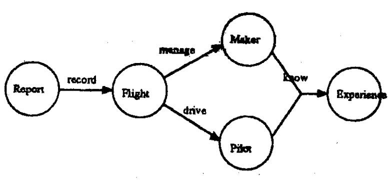


Figure 6 flight record context

4.3 文脈を持つ関数制約

文脈 $\alpha(X,Y)$ 上の連想列 (association sequence) $\overline{a}(x,y)$ を次とする：

(1) $\alpha(X,Y)$ が述語 p 上の関数制約 $p(X,Y)$ のとき、$\overline{a}(x,y) = \langle a_1 \rangle(x,y)$、ここで a_1 は $p-$連想で、(x,y) は定義型 X,Y 上での a_1 の実体組 $a_1(XY)$ である。

(2) $\alpha(X,Y)$ が $\alpha_1(X,Y_1)q(Y_1,Y)$、但し $q(Y_1,Y)$ は述語 q の Y_1Y への射影とするとき、$\alpha_1(X,Y_1)$ の連想列 $\overline{b}(x,y_1)$ に対して、$q-$連想 a_1 で $a_1[Y_1] = y_1$ となるものを選び、$\overline{a}(x,y) = \langle \overline{b}, a_1 \rangle(x,y), y = a_1[Y]$ とする。

(3) $\alpha(X,Y)$ が $\alpha_0(X,Y_0)[\alpha_1(Y_0,Y_1),\alpha_2(Y_0,Y_2)]$ のとき、$\alpha_0(X,Y_0)$ 上の連想列を $\overline{b}(x,y_0)$、$\alpha_1(Y_0,Y_1), \alpha_2(Y_0,Y_2)$ の連想列をそれぞれ $\overline{c_1}(y_0,y_1), \overline{c_2}(y_0,y_2)$ とすれば、$\overline{a}(x,y)$ を $\langle \overline{b}, [\overline{c_1}, \overline{c_2}] \rangle(x,y_1y_2)$ とする。y は y_1y_2 である。

文脈 $\alpha(X,Y)$ の関数制約とは、$\sigma_\alpha : X \rightarrow Y$ の形の式を言う。データベース d が文脈 $\alpha(X,Y)$ の関数制約$\sigma_\alpha(X,Y)$ を満たすとは、任意の部分文脈 $\beta(X_1,Y_1) \prec \alpha(X,Y)$ に対して、$\overline{a}(x_1,y_1), \overline{b}(x_1,y_2)$ が文脈 $\beta(X_1,Y_1)$ 上の任意の連想列ならば、$y_1 = y_2$ となるときを言う。この定義から明らかなように、文脈 α の関数制約$\sigma_\alpha : X \rightarrow Y$ とは、α 上の任意の部分文脈 $\alpha'(X_1,Y_1)$ に対して X_1 から Y_1 へ関数的に実体が対応することを意味する。この関数制約を σ_α の $\alpha'(X_1,Y_1)$ 上への射影という。

文脈をもつ関数制約の集合を F とするとき、データベース d が文脈 $\alpha(X,Y)$ の下で F を満たすとは、F に含まれる文脈 $\alpha(X,Y)$ の任意の関数制約 σ_α に対し、d が$\sigma_\alpha(X,Y)$ を満たすときを言う。F が σ_α を含意するとは、F を満たすどのようなデータベースも必ず σ_α を満たすときを言い、$F \models \sigma_\alpha$ と表す。

5 関数制約の推論

複数述語上の文脈を持つ関数制約の推論規則を考えることができる。先に述べたように、文脈が無数に存在し得るため、文脈を持つ関数制約すべては、一般には有限時間で計算できない。しかし、各文脈に対しては導出過程を計算することができる。

C1:反射則 $\pi : X \rightarrow Y$、ただし ある述語 $p(E_1...E_n)$ があって $Y \subseteq X \subseteq \{E_1,...,E_n\}$ を満たす。π は射影操作を表し、特に $X = Y$ のとき id とも表す。

C2:合併則 $\sigma_1(X,Y_1)$、$\sigma_2(X,Y_2)$ がそれぞれ文脈 α,β をもつ関数制約で、Y_1,Y_2 を含む述語 p があり $Y_1 \cap Y_1 = \phi$ ならば、$[\sigma_1,\sigma_2](X,Y_1Y_2)$ は文脈 $[\alpha,\beta]$ の関数制約である。

C3:推移則 $\sigma_\alpha(X,Y)$, $\sigma_\beta(Y,Z)$ が文脈を持つ関数制約ならば、$\sigma_\alpha \cdot \sigma_\beta(X,Z)$ は文脈 $\alpha \cdot \beta(X,Z)$ の関数制約である。

C4:射影則 文脈 $\alpha(X,Y)$ の関数制約 $\sigma(X,Y)$ に対し、$\beta(X_1,Y_1) \prec \alpha(X,Y)$ ならば

$\sigma(X,Y)$ の β への射影 $\sigma_\beta(X_1,Y_1)$ は文脈 β の関数制約である。

関数制約集合 F から $C1, C2, C3, C4$ を用いて σ を得るとき、σ は F から導出されると言い $F \vdash \sigma$ と表す。文脈 α を示す必要のあるときは、$F \vdash \sigma_\alpha$ などと表す。

例題 7 例 5 で論じた文脈を持つ関数制約を導出する。

(1) $wife$: 男 → 女
(2) $mother$: 男 → 女
(3) $cloth$: 女 → サイズ

$\sigma_{wife \cdot cloth}$(男、サイズ) は (1),(3) を推移則 $C3$ によって得られる。$\sigma_{mother \cdot cloth}$(男、サイズ) も同様である。

例 6 では次の 4 つの関数制約が与えられている。

(1) $record : Report \rightarrow Flight$
(2) $manage : Flight \rightarrow Maker$
(3) $drive : Flight \rightarrow Pilot$
(4) $know$: $\{Pilot, Maker\} \rightarrow Experience$

(2), (3) を用いて $C2$ から、$\sigma_{manage} \cdot \sigma_{drive}$ を合併して文脈 $[manage, drive](Flight, Maker Pilot)$ をもつ関数制約が得られる。$C3$ から (1) σ_{record} および (4) σ_{know} と結合することで、$\sigma_{record \cdot [manage,drive] \cdot know}(Report, Experience)$ を得る。□

述語定義が 1 つしか表れない導出では、文脈を論じる理由はない。実際、次の性質が成り立つ。

定理 1 述語定義がひとつしか生じないとき、推論規則 $C1, C2, C3, C4$ から Armstrong 推論則が導出できる[4]。

(証明) 幾つかの規則を導出する。
$F1, F3, F5$ はそれぞれ $C1, C2, C3$ 自体である。
$F2 : X \rightarrow Y$ ならば $XZ \rightarrow Y$ 。実際、$C1$ より、$\pi_X : XZ \rightarrow X$ であり、$\beta : X \rightarrow Y$ とすれば、$C3$ から $\pi_X \cdot \beta : XZ \rightarrow Y$ を得る。
$F4 : X \rightarrow YZ$ から $X \rightarrow Y$。実際、$C1$ を用いれば、$\sigma : X \rightarrow YZ$ に対して、$\sigma \cdot \pi : X \rightarrow Y$ を得る。
$F6 : X \rightarrow Y, YZ \rightarrow W$ ならば $XZ \rightarrow W$。実際、$C1$ から $\pi_X : XZ \rightarrow X, \pi_Z : XZ \rightarrow Z$ を得る。$C2$ より $[\pi_X \cdot p(XZ,Y), \pi_Z(XZ,Z)]$ を得る。$C3$ より、$[\pi_X \cdot p(XZ,Y), \pi_Z(XZ,Z)] \cdot p : XZ \rightarrow W$ を得る。□

文脈を持つ関数制約集合 F の、文脈 α での閉包 F_α^+、および定義型集合 X の F の文脈 α での閉包 X_α^+ を定義する。F_α^+ の要素には全て文脈を添字に付ける。$F(p)$ を述語 p に関する関数制約 ($p(X,Y)$ の形のもの) で F に含まれるものの集合とし、$F(p)^+$ を Armstrong 推論則で得られるその閉包とする。X_α^+ は、F_α^+ によって X から決定される ($Range$ を添えた) 実体型の集合を表す。

$$F_\alpha^+ = \{\sigma_{\alpha'} \mid F \vdash \sigma_{\alpha'}、\alpha' \prec \alpha\} \cup \bigcup_{p\,\text{は}\,\alpha\,\text{に生じる述語}} F(p)^+$$

$$X_\alpha^+ = \{A_p \mid F_\alpha^+ \vdash \sigma_\beta : X \rightarrow A_p、p\ \text{は}\ Range(\beta)\}$$

例題 8 一夫多妻制度では、母、父、夫は関数的に対応するが、妻はそうでない。この状況を記述する関数制約集合 F は次である：

(1) $mother$: 男 → 女
(2) $father$: 女 → 男
(3) $husband$: 女 → 男

但し、妻 ($wife$) という述語は存在している: $wife$(男、女) 。 このとき、文脈 $\alpha = father \cdot wife \cdot husband$ に対する閉包 F_α^+, 女$_\alpha^+$ を計算する。

$$F_\alpha^+ = \{id_{father}, id_{wife}, id_{husband}, father, husband\}$$

$$\text{女}_\alpha^+ = \{\text{女}_{wife}, \text{女}_{father}, \text{女}_{husband}, \text{男}_{husband}, \text{男}_{father}\}$$

□

同様に、この推論則を用いれば、$F \vdash \sigma_\alpha \Leftrightarrow F_\alpha^+ \vdash \sigma_\alpha$ 、$F_\alpha^+ \vdash \sigma \Leftrightarrow \sigma \in F_\alpha^+$ 、$\sigma_\beta : X \rightarrow A \in F_\alpha^+ \Leftrightarrow A_p \in X\alpha^+$、但し $p = Range(\beta)$ などの性質が証明できる。

定理 2 文脈を持つ関数制約の推論則 $C1, C2, C3, C4$ は健全かつ完全である。

(証明) 個々の関数制約が、それぞれ当該する文脈で定義できることに注意する。個々の規則の健全性は易しい。完全性を示す。

関数制約 $\sigma_\alpha : X \rightarrow Y$ に対して、$F \models \sigma_\alpha$ ならば $F \vdash \sigma_\alpha$ を示すため、$F_\alpha^+ \nvdash \sigma_\alpha$ として、$F_\alpha^+ \wedge \neg\sigma_\alpha$ を充たすデータベースを作れば良い。

F, α に生じる述語を $p_1, ..., p_n$、$R_i = p_i(E_1^i ... E_{n_i}^i)$ とする。p_i − 連想集合 $\{t_1^i, t_2^i\}$ を次のように作り、これ以外の連想集合を空集合とする：

t_1^i は、p_i の定義型上で 1 をとる連想
: $t_1^i = 1...1$
t_2^i は、$A_{p_i} \in X_\alpha^+$ 上で 1、それ以外で
0 とする連想

t_2^i が到るところ 1 となれば t_1^i, t_2^i は同じとなる。

この割当 (データベース) d によって σ_α は充足されない。これを証明するため、実際に $\alpha(X,Y)$ 上の連想列$\overline{a}(x,y_1), \overline{b}(x,y_2)$ を作ろう。F_α^+ から σ_α が導出できないとしているから、

[4]次の 6 つの規則を言う：
F1:反射則 $X \rightarrow X$
F2:増加則 $X \rightarrow Y$ならば $XZ \rightarrow Y$
F3:合併則 $X \rightarrow Y, X \rightarrow Z$ならば $X \rightarrow YZ$
F4:射影則 $X \rightarrow YZ$ならば $X \rightarrow Z$
F5:推移則 $X \rightarrow Y, Y \rightarrow Z$ならば $X \rightarrow Z$
F6:準推移則 $X \rightarrow Y, YZ \rightarrow W$ならば $XZ \rightarrow W$

$\alpha'(X,Z) \prec \alpha(X,Y)$ となる接頭文脈 α' で、射影関数属 $\sigma_{\alpha'}(X,Z)$ が F から導出できず、しかもその真接頭部分文脈がこの性質を持たないとする。X からα' に沿って、$\overline{a}$ 上の p_i − 連想として全て t_1^i を選ぶ。$q = Range(\sigma_{\alpha'})$ とし、Z を Z_q と改めて表せば、$t_2^q(Z_q)$ は 1...1 の形ではない。というのは、仮定から Z_q は完全には X_α^+ に含まれないので、少なくともひとつ 0 を含むためである。$\overline{b}$ は q 以前は $\overline{a}$ と同じで、q − 連想 として $y_2 = t_2^q(Z)$ とすれば、この連想列組は反例となる。

F_α^+ に含まれる任意の関数制約 $\rho(U,V)$ で文脈がβ とする。β 上の任意の連想列 $\overline{a}(u,v_1), \overline{b}(u,v_2)$ が存在するとき、$v_1 = v_2$ を証明する。

$\overline{a}, \overline{b}$ を先頭から比較し、初めて異なる連想が p_i − 連想を g, g' とする。$g[X_i] = g'[X_i]$ かつ $g[Y_i] \neq g'[Y_i]$ であるから、$t_1^i \neq t_2^i$ で、例えば $g = t_1^i, g' = t_2^i$ のはず。$g[X_i] = g'[X_i] = t_1^i(X_i) = 1..1$ であるから、X_i のどの要素 A_{p_i} も X_α^+ に含まれる。関数制約 $p_i : X_i \to Y_i$ は $C4$ より F_α^+ に含まれ、しかも p_i は α に生じる $Range$ のひとつだから $Y_i \subseteq X_\alpha^+$ となる。従って t_2^i の作り方から Y_i 上で 1..1 となる。$g[Y_i] \neq g'[Y_i]$ であるから矛盾。□

例題 9 述語定義を $p(AB), q(ACD), r(BC)$ とするとき、関数制約集合 $F = \{p : A \to B, q : A \to C, q : C \to D, q : A \to D, r : C \to B\}$ を考える。このとき、関数制約$\sigma_{p(A,B)r(B,C)q(C,D)} : A \to D$ は F から導出できない。証明の手順に従った反例データベースは次のようになる:

p	A	B
	1	1

q	A	C	D
	1	1	1

r	B	C
	1	1
	1	0

□

関数制約の含意問題は多項式時間で決定可能である。実際、関数制約集合 F と関数制約 $\sigma_\alpha : X \to Y$ に対して、$F \vdash \sigma_\alpha$ を判定する手間を考えると、F の長さ N, σ_α の長さを M とすると、高々 $O(N \times M + N^2)$ 、つまり $O(N^2)$ を越えることはない。

6 結び

本稿では文脈を持つ関数制約を論じ、データモデリングにおいて、一組の実体型集合に複数の関数対応述語が定義されるときに、形式的に推論する基礎を与えた。特に、関係モデルにおける関数従属と同様、効率よい推論機構が存在し、無矛盾で独立したスキーマを得ることが可能である事を示した。

この種の問題の難しさ・複雑さは、多重継承として知られる問題に起因している。本稿の結果は、このクラスが形式化でき推論機構を装備できることを示している。実際、推論の曖昧性を除去するためには、継承をすべて導出するような文脈を与えれば十分であり、文脈が個々の意味を分離する方法になる。

ここで示した枠組みは、一貫性制約だけではなくて質問機構にたいしても重要である。著者らは、メタレベルの制御を用いれば、極めて高水準のデータ独立性が達成できることを論じている [10]。この本質は、利用者の質問の意図を動的に計算し、'戦略' によって曖昧さを除去することにある。本稿で提案した文脈は、戦略を記述するためのツールとみなすことができ、簡潔で強力な質問機構の基礎となる。

参考文献

[1] Abiteboul, S., Hull, R. and Vianu, V.: Foundation of Databases, Addison-Wesley, 1995

[2] Armstrong, W.: Dependency Structures of Database Relationships, *IFIP*, 1974, 580-584

[3] Beeri, C.: Formal Models for Object-Oriented Databases, *DOOD*, 1990, 405-430

[4] Chen, P.P.: The Entity Relationship Model, *ACM TODS* 1-1 (1976), 9-36

[5] 小林功武: 古典データベースから演繹データベースへ. (特集:演繹データベース). 情報処理学会誌 31-1, 1990, 189-197

[6] Miura, T., Arisawa, H.: Logic Approach of Data Models – *Data Logic*, *Future Database Systems*, 1990, 145-159

[7] Miura, T.:A Visual Data Manipulation Language for A Semantic Data Model, *IEEE COMPSAC*, 1991, 212-218

[8] Miura, T.:A logical Framework for Deductive Objects, *Info.Syst.* 17-5 , 1992, 395-414

[9] Miura, T.: Database Paradigms toward Model Building, *ORM*, 1994, 228-258

[10] Miura, T. Shioya, I.: Strategy Control for Database Queries. *DEXA*, 1995, 231-240

[11] Ullman, J.D.: Principles of Database and Knowledge Base Systems (Vol.1,2), Computer Science Press, 1988