証明手続きとしてのデータ論理

三浦孝夫
産能大学経営情報学部

伊勢原市上粕屋１５７３
電話（０４６３）９２－２２１１
電子メイル　miura@sanno.junet

本稿では、データ論理によるデータモデルの形式化を提案し、また導出原理による完全な証明手続きを有することを言う。データ論理は第一階述語論理を型概念を用いて拡張したもので、データモデルをその解釈とする。定数を主体に、構造を含む項を複合値に対応させ、両者を明示的な宣言により結び付る。構造項を値とする変数を導入することで、動的に定義された構造項を考えることが出来る。質問は、データベースに格納された情報を取り出すだけでなく、型付けによって計算できるものも対象とする。

データ論理は形式化のみならず、証明手順としても有用であることを示す。つまり、データ論理による演繹操作のうち、ここでは導出原理による定理証明が完全であることを証明する。構造項を扱うように拡張されたエルブラン領域を用いて、スコーレム定理、エルブラン定理を証明し、単一化を拡張して導出原理が健全且つ完全であることを示す。

Data Logic As Proof Procedures

Takao Miura
Department of Management and Informatics
SANNO College

Kamikasuya 1573, Isehara, Kanagawa, Japan
Tel (0492)96-2211
E-Mail miura@sanno.ac.jp

In this paper, we present Data Logic to formalize data models with complex objects, and we show that the Resolution Principle is sound and complete. Data Logic is an extention of the typed first order theory whose interpretation is a semantic model. Constants are considered as entities (objects), structure terms as complex objects. Entities and complex objects are related by explicit declaration. All structures are not assumed to be objects but dynamically defined complex values can be defined by structure-valued variables.

The Completeness of the general Resolution Principle on Data Logic presents the solid foundation of Logic Programming and Deductive Databases. To clarify the usefulness, several fundamental properties are given such as Skolemization, Herbrand theorem as well as other basic operations like typed unifiers.

１　前書き

過去２０年もの間に関係データベースの研究が精力的になされ、商用に到る成果をもたらした。反面、ＣＡＤ・ＣＡＭなどの新しい応用分野でこれまでのデータベースに基づいたものが不十分との指摘がある。この動機から、構造情報の自然な取扱をねらった非正規関係データモデル、格納された事実と規則から導出される情報を扱う演繹データベース、プログラム言語で盛んに研究されているオブジェクト指向との融和とデータベース操作の意味ギャップ（インピーダンスミスマッチ）を解消するオブジェクト指向データベースなどの、新しいデータモデル化を求める研究が開始されている[1-3]。

データモデルはこれまで活発に提案され、基本概念、操作、質問機能、制約条件記述など様々な観点から比較されている[4,5]。このうち、意味データモデルは重要な役割を果たしており、主体と型（タイプ）の区別、関連（述語）と主体の区別等が特長的である。意図の明確な表現、ビュー上の更新、情報の関連性の明示的な把握等は、関係データモデルにない利点である[6]。反面、形式化の数学的基盤に欠けるため、曖昧な解釈を含むことや操作系による記述能力を問えない等、緻密さを必要としている。本稿は、複合オブジェクトと意味データモデル機能を有するデータモデルを、論理によって形式化する試みである。ここでは、代表的にＡＩＳデータモデルを中心に考察する[7,8]。

まず「データ論理」と呼ぶ論理クラスを提案する[10,31]。データ論理は述語論理を構造記述で拡張した構文を有し、意味データモデルをその解釈とする。意味データモデルの記述はすべてデータ論理で表現され、データ操作は解釈（データベース）と質問式から評価することが出来る。又、型制約を導入しこの充足性を論じることが出来る。データ論理では定数の他に構造情報も構文単位（項）となり、複合オブジェクトは構造項として表される。

データ論理に導出原理を定義し、これに基づく証明手続きは完全となる。すなわち、データ論理で含意される事実にはそれを証明する方法が存在する。この結果、データ論理は証明方法としても捉えることができ、論理プログラムの動作原理や演繹データベースの理論的な基礎を提供する。

第２章では、データモデルの位置づけと、基本となる概念を述べる。又、関連研究を評価する。第３章では形式化が行われる。構文と解釈を導入し、型制約を定義する。第４章では、導出原理を導入するための諸定義を行い、いくつかの重要な性質、特にスコーレム化とエルブラン定理を照明する。第５章では、型を伴う代入と単一化を導入して、導出原理を定義し、これがある制限のもとで健全且つ完全であることを言う。第６章ではこれらを要約する。

２　データモデルとデータ論理

２．１　主体と連想

ここでは、以下の議論に必要なデータモデルの考察を行う。データベース設計におけるデータモデルの役割は次の４つに整理できる[10,31]。

・データの認識・利用に共通の「場」を与える。
・多様な理解認識を整合化・一般化する。
・仕様や検証に数学的基盤を与える。
・実現に有用な情報を提供する。

意味データモデルによるデータ設計では、「もの」やその「関連性」が一定の「制限」のもとで情報を表わすと考える。「もの」を表す認識単位として主体（entity）を、「関連性」を表す認識単位として連想（association）を用いる。「制限」は「制約条件」として表される。主体は直接「もの」を捉える概念であり、表現方法や性質の記述には左右されない。オブジェクト同一性はまさしくこの点を問題とする[11,12]。同様に各結び付きごとにひとつの連想が対応するが、同じ主体の組で異なる意味を表す。

主体記述では、認識しやすく取扱い易くするため、主体型に着目する。主体型は主体の持つ共通の性質（カテゴリ）を意味する内包概念ある。主体は幾つもの性質を有することから複数の型を持っていてよい。これと同様に連想の共通の性質が存在し、この内包性を述語(predicate)と呼ぶ。述語によって結びつく主体の型は、共通の意図を表し、述語の定義型という。

実世界の「関連性」は通常述語としてうつしとられる。しかし注意してみると、集合やグラフと言った抽象性の高い情報も数多く出現する。これまでデータ構造には計算機の実現手法として構文的な性質や効率よいアルゴリズムが数多く知られている。もし構造情報をデータ構造で表現できれば、記述の簡潔性や構造の強要による意図表現が容易に達成できよう。本稿ではこの対応に着目し、データ構造としての意味を解釈に強要するモデルを検討する。構造情報を直接扱う処理が頻繁に出現する分野（ＣＡＤ，ＭＩＳなど）での、非正規関係や複合オブジェクトへの期待はここにある[13-15,32,33]。

意味データモデルの立場からは、データ構造で表される情報は主体ではない。本稿では、構造情報も型表現に従うと考える。この型表現は、主体型から再帰的または非再帰的に得られるもので複合型という。集合主体と構造情報の対応は、主体（e）と複合値（v）の陽な関数対応（λ(e)=vと書く）によって記述される。構造を表す主体とその構造情報は必ずこの対応を有する。

形式的な枠組みからは、述語表現と構造項表現の差は小さい。事実、構造を表す公理を加えれば対応は易しい[16-19]。しかし、解釈及び操作を考慮すれば構造項は次の３つの点で有利である。

・構造情報の扱いのため構造値変数を用いる。
・再帰的な型を持つ事が出来る。
・集合構造を扱える。

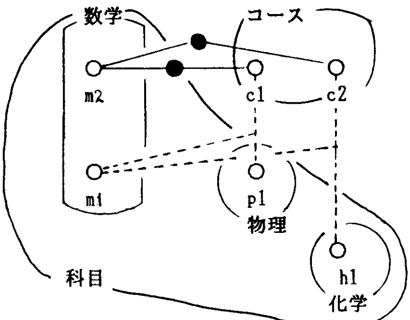
２．２　データ論理の非形式的な例

ここでは、次章以降で定義される概念を例を用いて非形式的に述べる。

データ論理では論理式の解釈に意味データモデルを用いる。この代表としてＡＩＳデータモデルを用いる[7,8]。ＡＩＳでは主体、連想（関連）、主体型及び連想型（述語）を基本概念とする。各主体は１つ以上の型を有する。例２．１では主体型・主体の例で、型とその解釈を示している。

【例２－１】

記法	主体	型
m1	数１	科目、数学
m2	数２	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



■

ＡＩＳでは汎化や専化は主体型上の制約と考える。対象とする主体型上の制約では、主体集合の和や交差からなる式の間の包含性や排他性を記述する。例２．２では主体集合（型）の制約を示している。

【例２．２】

＃｛数学、化学、物理｝

数学・化学・物理には共通の主体が無い

数学＜科目、化学＜科目、物理＜科目

科目の部分型になっている

■

ＡＩＳでは順序組・集合による構造化された情報（複合値）を扱う。構造情報は、それ自体が独立した意味を持ち、値指向アプローチをとる[33]。例２．３はＡＩＳデータベースの例であるが、更に、数２は「数１と物１」および「数１と化１」の複数の前提科目を複数要求する事が記述される。

【例２．３】

・前提コース（ｐ）

（科目	コース）
m2	c1
m2	c2

・構造項対応

コース	｛科目｝
c1 〜	{m1,p1}
c2 〜	{m1,h1}

■

ＡＩＳはこれまで実世界のモデル化を中心に考察されてきたが、データ論理は論理操作言語として位置づけられる。データ論理式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。例２．４、例２．５の２つは、例２．３の情報（データベース）に対して適用される。構造情報は格納されていないが検索されていることに注意されたい。動的に構造値を計算する事で、質問能力の拡大が出来る。

【例２．４】

(x／科目)：(∃y／科目)

(∃z／コース) (∃w／｛科目｝)

前提 (y,z) ∧ z〜w ∧ w∋x

前提科目に生じる科目を求める。■

【例２．５】

(x／｛科目｝)：(∃y／科目)

(∃z／コース) (∃w／｛科目｝)

前提 (y,z) ∧ z〜x ∧ x∋y

前提科目に自分自身を含む前提科目集合を求める。■

２．３　データ論理と導出原理

データ論理は、型の取扱い易さから、第一階多類述語論理を基本とする。しかし、複数の型を有する主体を許さず、型に関する制約記述ができず、構造情報を扱えないなど、直接応用することが出来ない。データ論理は、データベース（データモデルに基づく具体値集合）を解釈とし、型及び連想についての制約条件を有する論理モデルを意味とする。型、主体、連想及び型制約、連想制約のそれぞれについて、構文論と意味論が対応した形式論理である。多類論理を拡張し、冠頭標準形の存在や式の安全性など、データ操作を指向した性質がある[31]。

データ論理で扱う複合項のうち、集合の扱いは本質的に第一階論理で対応できる、というのは有限集合のみを考察対象としているからである。また、代入や単一化の意味がこれまでとは異なる。例えば後述するように、ｍｇｕは一意ではないが、最汎単一形は確定するため、導出原理は単一形を基にして定義される。また型付けされた項を用いるため、単一化されたあとの型の関連が重要な意味を有する。導出原理を用いた証明は、この代入や単一化の拡張により再構築されねばならない。本稿では、データ論理上の導出原理が健全且つ完全な証明手続きであることを述べる。

２．４　関連研究

関係は制約条件のモデルであり、論理と自然な対応がある。しかし、論理は個体を特別視せず（このことを値システムとよぶ[6]）、主体と属性の区別が出来ない。古典的な関係論理が第一階論理によって意味付けされているためである。ＲＭ／Ｔは、主体と属性を認識する操作系を有している[20]。関係代数や関係論理は拡張された意味をもつが、その数理論理的な意味はかえって曖昧なものとなる。

構造情報を述語を用いて表すアプローチ、例えばメンバシップや順序組構成の関数表現を考え、集約関数を用いた構造操作の形式化が提案されている[16,17]。オブジェクト論理は、順序組と（無限）集合を項に加え、メンバシップと等号だけを扱う操作系である[18,19]。高階操作を多類論理によって解釈している。

オブジェクトモデルは近年活発に議論されている[2]。意味データモデルとの決定的な違いは「手続き」にある。しかし、チューリング計算可能なデータベース操作言語は、（体系だてたデータアクセス機能、整合性や信頼性、同時制御と言ったＤＢＭＳ機能を利用した）１レベル上位の考え方であり、データベースの問題でなくどの様に活用すべきかを目的とする[6]。

O_2のオブジェクトは操作単位であり、主体と値の組からなる[21,22]。この形式化は型システムに基づいており集合の包含性によって意味付けされる[23]。データ論理とは２つの点で違う：（ア）O_2では同じ構造を持つ項は同じ意味を持つため、同じ値で異なる型を持つ２つの主体は表現できず、値とは独立な主体が、値によって主体型を規定されてしまう。（イ）O_2で扱われる複合値は、定められた制約を満すときだけ有効で、主体と常に一体で考えねばならず区別しにくい。

ＩＦＯの複合オブジェクトへの拡張は、構造操作のある基準での完全性を示したものとして興味深い[24]。しかし、ここでは構造情報自体が主体であり、２つの構造情報が同じ主体を表すケースを扱えない。複合オブジェクトアプローチのように各オブジェクトはひとつの表現

しか持てない（値システム！）。

Ψ項論理[25]に基づくモデル化や、O論理[12,26]では、オブジェクト（主体）を陽に扱うメカニズムを持たず、変数（タグ）を導入して間接的に行う。構造と主体との対応は意味データモデルで言うそれとは異なる。

F論理は、ここで提案するデータ論理と同様、完全な証明手続きを有し、計算手法や証明手法としても議論することが出来る[27,28]。ただ、F論理は本質的に2項関係に基づくモデルであり、構造情報を直接扱うものでない。また、述語と項の区別を行わないため、解釈（データベース）では全てが主体として扱われる。本稿では、単なる集まりと、認識単位としての集まりを区別しており、異なる立場を取る。

3　データ論理

3．1　データ論理の構文

ここでは以下の議論で必要な構文の定義を行う。詳細は文献を参照されたい[31]。

データ論理のアルファベットは、次の記号の5つの可付番集合と論理記号から構成される；E（定数集合）、V（変数集合）、T（型名集合）、P（述語名集合。eq、∈、≡を含む）、F（関数名集合、Pの部分集合）、結合子（→、←、∩、∪、¬）、限量子（∃、∀）、句読点（（、））。また、述語（関数）型付け関数ξがつぎのようにあたえられている； $\xi: P \to \cup T^n$　$n=1, \ldots$。ただしT^nはTのn次直積を表す。

複合型(complex type)集合δ（T）とは、t，t_iがδ（T）の元のとき、Tの元、｛t｝、$\langle t_1 .. t_n \rangle$、$t_1 \cap t_2$、$t_1 \cup t_2$　の形の集合を言う。同様に、複合値(complex value)集合ζ（E）とは、e_iがζ（E）の元のとき、Eの元、$\{e_1 .. e_n\}$、$\langle e_1 .. e_n \rangle$の形の集合を言う（nは有限）。δ（T）とζ（E）とは可付番である。δ、ζはオペレータと考えることが出来る。

型構造(type structure)Σは、t≡expの形の元の有限集合である。但し、tは型名（定義名）、expは複合型（定義構造）であり、どの定義名も一度しか表れず、expに生じる名前は、定義名だけである。expはtを含んでいてよい。expがt自身のとき、tは基本型、集合｛..｝のとき集合型、順序組<..>のとき、順序組型という。順序組の構成要素e_iを成分という。

【例3．1：型構造】

第2．2節の例では、次が自明ではない型構造として与えられている：

コース≡｛科目｝

非公式に言えば、型構造は定義される型を持つ定数が、定義構造を型とみたとき、その構造値と対応していることを主張している。■

定数c、変数x，Tの元tとする。型式(type formula)とは、t（x）、t（c）と、それらの結合子や限量子による表現をいう。型式hが閉(closed)とは、どの変数xも限定されている時を言う。閉型式を型条件とも言う。型条件の有限集合を型制約とする。この性質については[10,31]を参照されたい。本論文では次の形の型制約しか扱わない：限量子を含まない型式Wにたいして、（$\exists x_1$）..（$\exists x_n$）（$\forall y_1$）..（$\forall y_m$）W　。型制約は、関数記号を含まない単項述語記号だけからなる閉式である。

【例3．2：型制約】

（∀x）数学（x）→科目（x）

　　数学は科目である

（∃x）¬（数学（x）∧物理（x））

　　物理かつ数学という科目はない　■

［定義3．1．1（項と整式）］

データ論理の項(term)とは、変数、定数、それらから構成される複合値、項を引数とする関数形式f（$t_1 .. t_n$）、t．nのいずれかを言う。項t、自然数nとする。変数x、y、｛x｝，｛x，y｝，<xy>等はすべて項である。

整式(formula)は次のように定義される；

・次は整式で、特に原子式(atom)と言う

Pの元p、t_iを項とすると、p（$t_1 .. t_n$）

t，sを項とすると、t∈s、eq（t、s），t≡s

・W_1，W_2 が整式なら、結合子による表現（$W_1 \to W_2$など）も整式である。

・Wが整式でt_iが項なら、（$Q_1 t_1 / T_1$）..（$Q_n t_n / T_n$）Wは整式である。但し、T_iは複合型、Q_iは∀，∃または何もないのいずれかで、∀または∃のときは、tは変数でなければならない。

∀，∃，何もなしで限量されている変数をそれぞれA限量，E限量、自由限量という。tが変数を含まない限量を基礎限量と言う。変数を含まない整式を基礎整式、どの変数もA－またはE－限量されているとき閉整式という。限量されていない変数または自由限量されている変数を自由という。閉整式を制約条件と言う。

データ論理(Data Logic)とは、型制約Δを持つ型構造Σ上で、制約条件集合Sを公理として有する整式クラス（言語）である。データ論理式(DataLogic Formula)とは、Wを整式とするとき、（x_1 / H_1）..（x_m / H_m）W　の形の整式で、m≧0であり他に自由限量変数が無いものを言う。■

【例3．3：データ論理式】

第2．2節での例で示される式は、データ論理式である。■

3．2　データ論理の解釈

データ論理の解釈とは、構文と外延（データベース）との対応付けをいう。

［定義　　データベース］

データベース(database)とは、<E' T' A' P' W' αβλ>で次を満たすものを言う；

・E'は主体の高々可付番無限集合、T'は主体型の高々可付番無限集合。

・α（e）はΩ（T'）の元、Ω（T'）はT'の（全ての有限集合からなる）べき集合を表す。主体eが持つ主体型全てをα（e）で表す。

・連想はp（$e_1 .. e_n$）と表す。A'は連想の高々可付番無限集合、P'は述語の高々可付番無限集合で、A'に生じる述語はP'の元で、A'に生じる主体はすべてE'の元である。

・述語pにたいしβ（p）＝$t_1 .. t_n$はP'から$\cup T^n$への関数で、pの述語定義型を表す。但しどのiについてもt_iはT'の元とする。

・（型整合性）任意の連想p（$e_1 .. e_n$）にたいし、β（p）＝$t_1 .. t_n$とすれば、どのiについてもα（e_i）はt_iを含む。

・E'から有限回の集合化・順序組化で得られる複合値の集合をW'とする。

・主体型tの要素のうち、複合値集合W'と関数対応するものがある。これはλ対応と呼ばれ、外延的に与えられる。■

データ論理の解釈は2つの部分に分かれ、制約を満たす型解釈と整式解釈からなる。以下ではデータベースDB=<E' T' A' P' W' αβλ>を固定する。

［定義　解釈］

データベースDBに対して、アルファベットE，T，F，Pとの自然な対応を考える。述語pについて、次の条件を満たすとする： ξ（p）$=t_1..t_n$ならば、β（p'）$=t'_1..t'_n$。つまり、ξにβを対応させる。

型名tに対し、Γ_0（t）=｛E' の元e' で、α（e'）がt' を含む｝とする。ただしΓ_0（t）は空ではないとする。

複合型exの解釈 Γ_1（ex）を次のように定義する；

・exがTの元tならば、Γ_1（ex）=Γ_0（t）

・exが集合型｛ex_1｝ならば、
Γ_1（ex）=Ω（Γ_1（ex_1））

・exが順序組型<$ex_1..ex_n$>ならば、
Γ_1（ex）=Γ_1（ex_1）×..×Γ_1（ex_n）

・exが連言$ex_1 \cap ex_2$ならば、
Γ_1（ex）=Γ_1（ex_1）∩Γ_1（ex_2）

・選言（∪）の場合も同様

複合値vが複合型exを持つとは、解釈値v' がΓ_1（ex）の要素のときをいう。

型構造Σの要素 t≡exの解釈Γ_2（t≡ex）は次で与えられる：｛Γ_1（t）の要素e' で、λ_{ex}（e'）が定義されΓ_1（ex）の要素｝。ただし、基本型t≡tを正しく扱うため、λは自明な等式（e'、e'）を含むものとする。これはまたΓ_2（t）とかく。

DBにたいする型解釈Γとは、tがTの元のときΓ_2（t）、δ（T）の元でTの元ではないときΓ_1（t）とするものをいう。■

型解釈が型制約Δを満たすとは、主体集合とその型付けが、Δの各型条件を満たす事をいう。形式的に述べれば、次のようになる。

［定義　型制約の解釈］

型解釈Γが型制約Δを満たすとは、各型条件が真となる時を言う。ここで型条件が真とは、次をいう：

・定数e，Tの元tの時、t（e）が真なのは、e' がΓ_2（t）の要素の時をいう。

・h_1，h_2が型条件のとき、連言$h_1 \cap h_2$が真なのは、h_1、h_2共に真の時をいう。　選言$h_1 \cup h_2$、含意$h_1 \to h_2$、否定 ¬h_1も同様

・（∀x）h_1が真なのは、全ての主体eに対してh_1の自由な出現xにeを代入したとき、真となる時を言う。（∃x）h_1も同様。

型解釈が型制約Δを満たすとき、Δに関して無矛盾な型解釈、またはΔ型解釈という。Γ（Δ）とも表す。

■

このクラスはこれまでの制約条件の多くを含んでいる[9]。また証明を略すが、単項述語だけからなる式の充足性判定は決定可能である[29]。

［定義　項の解釈］

項の解釈は次のような対応を言う：

・Vの元（変数）に対し、変数から主体（E' の元）及び複合値（W' の元）への代入を対応させる。

・Eの元（定数）に対して、E' の要素を対応させる。

・K=V∪Eとするとき、ζ（K）の元にたいして、その構成要素（項）から得る複合値を対応させる。

・項$t_1..t_n$に対応した解釈が$t_1'..t_n'$で、関数fの解釈値がf' ならば、f（$t_1..t_n$）の解釈値をf'（$t_1'..t_n'$）とする

・関数fに対して、次のような関係を対応させる：
fの解釈=｛A' の元でf' を述語に持つ連想f'（$t_1'..t_n'$ t'）｝

fの定義型をξ（f）=$t_1..t_n$tとすると、型整合性よりfの解釈は、Γ（t_1）×..×Γ（t_n）×Γ（t）の部分集合と同一視できる。更に、fの定義域$t_1..t_n$上で一意でなければならない。

・項t．nの解釈とは、tの解釈t' が順序組でそのn番目の成分を言う。

■

例えば、x、yを変数とするとき、項｛｝は空集合、｛x｝は（代入後は）単一要素集合、｛x，y｝は高々要素数2の集合を対応させることが出来る。集合記法によって要素数の上限が限定される。

［定義　基礎原子式の真偽値］

述語pに対して、次のような関係をpの解釈という：
｛A' の元でp' を述語に持つ連想p'（$a_1..a_n$）｝。pの定義型をξ（p）=$t_1..t_n$とすると、型整合性よりpの解釈は、Γ（t_1）×..×Γ（t_n）の部分集合と同一視出来る。

・述語p、$t_1..t_n$を変数のない項とすると、p（$t_1..t_n$）が真とは、$t_1..t_n$の項解釈$t_1'..t_n'$がpの解釈に含まれるときをいう。

・t∈sが真なのは、t，sの解釈値t'，s' に対し、s' が集合でt' がs' の要素となる時を言う。

・eq（t、s）が真なのは、項解釈値t'，s' が等しい時を言う。但し、複合値ならば構成要素について eq が成り立つときをいう。

・t≡sが真なのは、項解釈値t'，s' に対して、λ（t'）=s' またはλ（s'）=t' が成り立つ時を言う。■

［定義　閉整式の解釈］

・整式が基礎原子式ならば、その真偽値を解釈とする

・W_1，W_2 を閉整式とするとき、$W_1 \cap W_2$、$W_1 \cup W_2$、$W_1 \to W_2$、¬W_1の4つの整式の解釈は、それぞれ連言、選言、含意、否定とする。

・Wを整式、t_1を項、（Qt_1／T_1）Wを閉とする。Qが限量子∀のとき、全てのΓ（T_1）の要素xに対して、W（x）が真となるときだけ、真とする。限量子∃も同様。自由限量でt_1が変数を含まないとき、t_1の解釈値t_1'がΓ（T_1）の要素でかつW（t_1）が真の時だけ、真とする。■

第3の型表現T_1はデータベースの型構造で与えられていなくてもよい。型解釈Γは、基本型からΓ_1解釈を基にT_1の外延集合を生成する。これはデータベースに格納されている値から演繹したことに相当し、構造情報を計算するメカニズムとして働く。例えば、（x／｛A｝）W（x）は、要素型をAとするような有限集合xでWが真となるものを答えとする。集合項｛$y_1..y_n$｝をWの引数とした時と比べ、要素数nを限定しない任意の有限集合となる。

［定義　モデル］

型制約Δを満たす解釈をΔ解釈と言う。閉整式の有限集合Sの全てを満たす任意のΔ解釈にたいして、閉整式φが必ず真となるとき、φはSのΔ論理帰結（S⊨φとかく）と言う。データベースDBが閉整式の有限集合Sのモデル(model)とは、DBがΔを満たし、Sのどの要素もDBによって真となるときを言う。これを明示して、SのΔモデルとも言う。型解釈Γ（Δ）にたいして、閉

整式θが妥当(valid)とは、Γ (Δ) 上の任意の解釈について、θが真となるときを言う。有限な閉整式集合SがΔ充足可能(satisfiable)とは、あるΔ型解釈でSのモデルとなるものが存在するときをいう。Δ充足不可能とは、Δ解釈でSのモデルがないときをいう。■

［定義　質問］
Wが開整式のとき、DBによるWの解釈とは、代入集合から真偽値集合への写像で、すべての自由な変数$x_1 .. x_n$にたいし、代入ωの結果を$a_1 .. a_n$とすると、Wにあるすべての自由なx_iをa_iと置き換えた結果真となるものをいう。

開データ論理式はまた質問(query)といい、解釈DBによる質問Wの解釈で真となるような代入の集合をこの質問の解(answer)と言う。■

3．4　標準形

［定義　同値性］
2つの整式W_1，W_2が共に閉または共に開とする。このとき、2つが同値(equivalent)または等価とは、次のように定義される。

共に閉ならば、どの様なΔ解釈DBであっても次が恒真のときをいう：$(W_1 \rightarrow W_2) \cap (W_2 \rightarrow W_1)$

共に開ならば、解集合が等しい時を言う。■

［定義　変換規則］

・A (x) … A (y)
　A (y) はAの全てのxを一斉にyに置き換えた結果、yはAには生じていない

・$(\forall x/T)$ W … W
　$(\exists x/T)$ W … W
　Wには自由なxはない

・$\neg(B \cup C)$ … $\neg B \cap \neg C$
　$\neg(B \cap C)$ … $\neg B \cup \neg C$
　$\neg(\neg B)$ … B
　$\neg(\forall x/T)$ W … $(\exists x/T) \neg W$
　$\neg(\exists x/T)$ W … $(\forall x/T) \neg W$

・$(\forall x/T)$ $B \rightarrow C$ … $(\exists x/T)$ $(B \rightarrow C)$
　$(\exists x/T)$ $B \rightarrow C$ … $(\forall x/T)$ $(B \rightarrow C)$
　$B \rightarrow (\forall x/T)$ C … $(\forall x/T)$ $(B \rightarrow C)$
　$B \rightarrow (\exists x/T)$ C … $(\exists x/T)$ $(B \rightarrow C)$
　Cには自由なxはない

定数しか含まない限量子c／Tはそのまま残す。変数とは別扱いとする。同じ変数は、高々一度しかAまたはE限量されないとしてよい。■

［定義　冠頭標準形］
整式で次の形を持つものを冠頭標準形という：$(x_1/T_1) .. (x_n/T_n)$ $(Q_1 y_1/G_1) .. (Q_m y_m/G_m)$ M。　ただし、$T_1 .. G_1 ..$はδ (T) の元で、$x_1 .. x_n$は相異なる変数。Qjは$\forall \exists$のいずれかである。また、Mは選言の連言形式で、変数を含む限量子はなく否定記号は最も内側にある。■

冠頭標準形は、自由変数の値域が最も小さいという望ましい性質を有する[31]。

［定理3．1］
全てのデータ論理式は等価な冠頭標準形に変換できる。

4　スコーレム化とエルブラン定理

4．1　スコーレム化

本節で述べる型制約Δは次の形の有限集合に限る；
$(\exists x_1) .. (\exists x_n)$ $(\forall y_1) .. (\forall y_m)$ R

ただし、Rは限量子を含まない型式である。

スコーレム化は、データ論理式が節形式に変換できることを保証する論拠を与える点で、きわめて重要な働きを持つ。まず、スコーレム化の定義を行う。

［定義　スコーレム化］
ϕを次の形のデータ論理式とする；　$Q_1 .. Q_n W$
ここで、Q_iは (a_i/t_i) 、 $(\forall x_i/t_i)$ または $(\exists x_i/t_i)$ のどれかとする。ここでx_iは変数、a_iは基礎項、t_iは複合型とする。このとき、次の操作を$i = 1 .. n$について行う；
・Q_iが (a_i/t_i) のとき、これを削除する。
・Q_iが $(\forall x_i/t_i)$ のとき、何もしない。
・Q_iが $(\exists x_i/t_i)$ のとき、これを削除し、Wに生じる全てのx_iを一斉に$f(y_1 .. y_k)$で置き換える。ここで、fは新しい関数記号、$y_1 .. y_k$はQ_1からQ_{i-1}までに生じる全てのA限量変数である。$k = 0$ならば、$f(y_1 .. y_k)$の代わりに新しい定数cで置き換える。この関数（定数）をスコーレム関数（定数）と言う。fは複合型上の関数である。こうして得られたデータ論理式Ψを、ϕのスコーレム化 (Skolemization) と言う。■

［定理4．1（スコーレム定理）］
ϕをデータ論理式、そのスコーレム化をΨとする。このとき、ΨがΔ充足不可能ならばϕもΔ不可能で、逆も成立する。

一般のスコーレム化と同様に、定理4．1は充足可能性について何も保証しない。

【例4．1】
ϕとして次を考える；　$(\exists x/t)$ p (x) 。このスコーレム化Ψは次である；　p (c) 。　このとき、次の解釈は充足性を保存しない；
　$\Gamma(t) = \{c, d\}$、$A = \{p(d)\}$
　$\xi(p) = E$　■

以下、ここではスコーレム化がなされたとする。型制約Δは、通常の第一階述語論理式（の部分クラス）であるため充足不可能性を保存したままスコーレム化が可能である。本稿では、スコーレム関数は生じ得ないがスコーレム定数は発生する。以下では、Δもスコーレム化がなされたと考える。

4．2　エルブラン解釈

定理4．1より、A限量変数だけを含み、自由変数を含まないデータ論理式だけを考えれば、充足不可能性を論じるのに十分である。このとき、エルブラン領域と呼ばれる特殊な主体集合上で論理帰結を考察できることを証明する。

［定義　節集合］
データ論理式ϕが、次の形であるとする；$(\forall x_1/t_1) .. (\forall x_n/t_n)$ $(C_1 \wedge \cdot \cdot \wedge C_m)$ 。　ただし、各C_iはリテラルの和　$L_1 \vee \cdot \cdot \vee L_k$　である。この時ϕの節集合Sとは、$\{C_1、\cdot \cdot、C_m\}$をいい、この先頭には限量子の並びQがあると考える事が出来る。Sから限量子Qを各C_iの頭部に直接付け、$\{QC_1、\cdot \cdot、QC_m\}$とする。　これを、Sの分配 (distributed) 節集合という。■

［定義　限量子による型制約］
節集合S，その限量子Qを $(\forall x/t)$ の形の並びとする。ここで、tは複合型とする。Q限量による型制約 (type

constraints induced by Q) Δ_Qとは、次を言う：Sに生じる　eq,∈,≡　以外の各述語記号pの出現毎に、その各位値の項uがQで限量されているとき、

(∀x/t) の形の限量に対し、t（y）

(u/t) の基礎限量に対して、t（u）

これ以外は何も定めない

これらをt_1（a_1）．．t_m（a_m）とすれば、次をpのこの出現に関する制約とする；

$\exists y\, t_1(a_1) \wedge .. \wedge t_m(a_m)$

Δ_1．．Δ_nを全ての制約とする。この時、Δ_Qを$\Delta_1 \wedge .. \wedge \Delta_n$とする。■

［補題４．２］
節集合ＳとＳの分配節集合Ｓ’のΔ充足不可能性及びΔ充足可能性は、任意の$\Delta \cup \Delta_Q$－解釈Ｉに関して等価である。

定理４．１より、ϕのΔ充足不可能性とＳのΔ充足不可能性は等価である。更に、補題４．２より、ＳとＳ’の充足性は、$\Delta \cup \Delta_Q$を充足する任意の解釈Ｉに関して等価である。本稿では、断わらない限り与えられたデータ論理式ϕに関する限量子Ｑについて、$\Delta \cup \Delta_Q$を改めて型制約Δと考え、Δは充足可能であるとする。

［定義　　エルブラン領域］
節集合Ｓが与えられているとする。Γ_S、H_T、H_S、H_Bを次のように定義する；

H_P＝｛Ｓに生じる　eq,∈,≡以外の全ての述語記号｝

H_T＝｛Ｓ及びΔに生じる全ての型名｝∪｛H_P内の述語の定義型｝

$H_S{}^0$＝｛Ｓ及びΔに生じる定数とＰの関数から構成される項｝

Γ_S＝｛t(s)　ただし　$t \in H_T$、$s \in H_S{}^0$｝

H_S＝｛$H_S{}^0$ から、複合型関数、順序組及び射影、有限集合化で構成される項｝

$H_B{}^0$＝｛p（$t_1..t_n$）、t_iはH_Sの元、ｐはH_Pの元｝

H_B＝$H_B{}^0$∪｛t1∈t2、t_iはH_Sの元｝∪
｛eq(t_1,t_2)、t_iはH_Sの元｝∪
｛t1≡t2、t_iはH_Sの元｝∪
｛eq(t_1.n,t_2)、t_iはH_Sの元、n は自然数｝

H_S、H_Bは可付番無限である。■

［定義　　エルブラン解釈］
Ｓを節集合、Δを型制約とする。ＩをＳのΔ解釈とする。Ｓのエルブラン型付け、エルブラン領域、エルブラン基底を、各々Γ_S、H_S、H_B とする。

Ｉに関するΔ－Ｈ解釈I_S（またはＨ解釈）とは次を言う；

（ア）Ｉ=＜ＥＴＡＰＷαβλ＞とすれば、I_S=＜$H_S{}^0 H_T H_A H_P H_W$ α’β’λ’＞（イ）Ｓの基礎項はそのままH_Sに対応させる。

（ウ）H_A＝｛$H_B{}^0$の元ｐ（$t_1..t_n$）、Ｉにおいてｐ（$t_1..t_n$）は真｝

（エ）$H_S{}^0$の元ｅに対してα’（ｅ）＝｛$t_1..t_n$｝、但し、α（ｅ’）＝｛t_1’..t_n’｝

H_Pの元ｐに対してβ’（ｐ）＝$t_1..t_n$、但しβ（ｐ’）＝t_1’..t_n’

λ’（ｔ）＝ｓ、但しλ（ｔ’）＝ｓ’

この結果H_Wが定義される。

（オ）eq(t,s)が真、ただしＩにおいて eq(t,s)が真。ｔ≡ｓ等も同様。

ＩがΔを充足するから、I_Sも型制約Δを充足する。ある解釈ＩについてのＨ解釈を単にＨ解釈という。$H_B{}^0$またはＨ解釈が整合している（consistent)とは、型整合性を満たす時を言う。■

［定義　　エルブランモデル］
整合するＨ解釈I_SがＳをΔ充足するとき、I_SをＳの（Δ－）Ｈモデル（エルブランモデル）という。Ｈモデルを持つときＳは（Δ－）Ｈ充足可能という。　■

［定理４．３］
節集合ＳがΔ充足可能ならば、ＳはΔ－Ｈ充足可能である。

［定理４．４］
節集合ＳのΔ充足不可能性とΔ－Ｈ充足不可能性は等価である。

４．３　意味木とエルブラン定理

節集合ＳがΔ充足不可能であることを証明するには、整合した任意のΔ－Ｈ解釈について、ＳをΔ充足しないことを言えばよい。つまり、Ｈ解釈を尽くす表現方法が望まれる。以下に述べる意味木は、Δ－Ｈ充足不可能性と有限の閉じた木を対応させることができる。

［定義　　意味木］
基礎原子式ϕに対して、｛ϕ、¬ϕ｝の組を相補組という。

Ｓを節集合、Δを満たす型解釈Γ（Δ）が与えられており、Ｓのエルブラン基底H_Bのうち整合しているリテラルをＢとする。この時、Γ（Δ）に関するＳの意味木（Semantic Tree）ＳＴとは、次で定義される；

・ルート節点をT_0 とする。
・ｉ＝０．．として順にT_iを定義する。
　Ｂからϕを取り除く。
　T_i の全ての葉に２つの節点とその間の枝を加えラベルをϕ、¬ϕとする。
　ｉをｉ＋１として繰り返す。

ＳＴのルートから節点ｎまでの正リテラルラベルの（有限）集合H_nを、ＳＴのｎでの部分Δ－Ｈ解釈という。■

［定義　　失敗節点と推論節点］
節集合をＳ、Ｓの型解釈Γ（Δ）に関する意味木をＳＴ、ｎをＳＴの節点とする。ｎが失敗節点とは、（ア）H_n がＳを充足せず、（イ）ｎより上位（ルート方向）の任意の節点ｍにたいしてH_m はＳのΔ充足性を決定しない、つまり、Δは充足するがＳの充足性については決定できない時を言う。ＳＴが閉じている（closed）とは、すべての葉が失敗節点の時を言う。ｎが推論節点とは、ｎの全ての子節点が失敗節点の時を言う。■

［定理４．５］
型制約をΔとする。節集合ＳがΔ充足不可能であることと、勝手なΓ（Δ）に関するＳの任意の意味木ＳＴが閉じた部分木を含むこととは等価である。

［定理４．６（エルブラン定理）］
節集合ＳがΔ充足不可能であることと、任意の型解釈Γ（Δ）にたいし、Ｓの整合する有限基礎例集合ＵでΔ充足不可能なものが存在することとは等価である。

５　導出原理

５．１　型付代入と導出原理

この節では、データ論理にたいして導出原理を拡張し、その証明方法が健全且つ完全であることを述べる。キー

となるアイデアは、代入や単一化が型操作を含むように拡張され、型付代入と型付単一化が定義されることにある。

前節で証明したように、節集合SのΔ充足不可能性はΔ－H充足不可能性と等価である。与えられたデータ論理式をϕ、Sの限量子の並びをQとすると、ϕがSの$\Delta \cup \Delta_0$論理帰結であることを証明するには、$S \cup \{\neg\phi\}$が$\Delta \cup \Delta_0$－H充足不可能であることを言えばよい。この考え方を反駁法 (Refutation) という。以下では、反駁法による証明を想定した諸定義を行う。

［定義　基礎導出形］
2つの基礎節C_1，C_2 が各々基礎リテラルL、¬Lを含むとする；
$C_1 : Q_1\ (L \vee D_1)$　　$C_2 : Q_2\ (\neg L \vee D_2)$
この時、$Q_1 Q_2\ (D_1 \vee D_2)$ を $\{C_1, C_2\}$ の型付導出形 (typed resolvent) と言う。■

【例5．1】
$S = \{\ (a/t_1)\ p(a) \vee q(a),\ \ (a/t_2)\ \neg p(a)\}$
導出形は $(a/t_1)\ \ (a/t_2)\ q(a)$で、簡約化できる；$(a/t_1 \cap t_2)\ q(a)$ ■
【例5．2】
$S = \{\ (a/t_1)\ p(a) \vee q(a),\ \neg p(a)\}$
例5．1との違いは第2節が積極的に型情報を有さないことである。この時でも、導出形は $(a/t_1)\ q(a)$である。■
［補題5．1］
2つの基礎節C_1，C_2に対して、Dを $\{C_1, C_2\}$ の導出形$Q_1 Q_2\ (D_1 \vee D_2)$ とすると、Dは $\{C_1, C_2\}$ の論理帰結である。

スコーレム化した時には、定義での形の基礎限量は生じない。しかし一般に変数を含む2つの節の導出形は、「共通部」を生成するための方法（代入）を必要とする。一般の第一階述語論理と違い、データ論理では型付けを強く意識する点が大きく異なる。
［定義　型付代入 (typed substitution) ］
σが（型付）代入とは、次のどちらかの形の有限集合を言う：　$x \rightarrow s / t$　　$x \rightarrow s$
ここで、xは変数、sは項、tは複合型である。恒等代入とは空集合代入を言う。改名代入とは、sが相異なる変数である$x \rightarrow s$の形だけを含む代入を言う。

節QCを考える、QはA限量または基礎限量の並びとする。QCのσによる代入例（$QC\sigma$）とは、次を言う：$x \rightarrow s / t$を代入の要素とすると、
・QのA限量変数x / uに対して、これを$s / u \cap t$で置き換える。
・Cの変数xを一斉にsに置き換える。
代入は節でなくても適用できる。以下では項についても代入を考えるものとする。2つの代入ρ、σを次の形の集合とする：
$\rho : x_i \rightarrow s_i / t_i \quad i=1..n$
$\sigma : y_i \rightarrow u_i / v_i \quad i=1..m$
この時その合成$\rho\sigma$とは、次の代入を言う；
$\{x_i \rightarrow s_i\sigma / t_i \quad i=1..n,\ y_i \rightarrow u_i / v_i \quad i=1..m\}$
但し、$s_i\sigma / t_i$がx_iに等しければ取り去り、またy_iでx_iに等しければこれも取り去る。■

【例5．3】
$C_1 : x/t_1\, p(x)$　$\sigma_1 : \{x \rightarrow y/t_2\}$
$C_1\sigma_1 : (y/t_1 \cap t_2)\ p(y)$
$C_2 : (x/t_1)\ y \equiv \{a,x,z\}$　$\sigma_2 : \{x \rightarrow z/t_2\}$
$C_2\sigma_2 : (z/t_1 \cap t_2)\ y \equiv \{a,z\}$
最後の例では集合が簡約化された事に注意されたい。■

［定義　型付単一化］
節集合Sを $\{Q_i C_i,\ i=1..n\}$ とする。Q_iはA限量または基礎限量である。この時、代入ρが（型付）単一化 (unifier) とは、$Q_i C_i \rho$がすべて等しい時を言う。またSの要素は単一化可能 (unifiable) という。単一化は節に限らず、2つの項についても定義できる。ρがSの最汎単一化代入（mgu）であるとは、任意の単一化代入σに対してある代入があり　$\sigma = \rho\theta$　となるときをいう。■

【例5．4】
次のいずれも$S_1 = \{\ (x/t)\ p(x),\ (y/s)\ p(y)\}$ の単一化代入、実際にはmguである；　$\sigma_1 : \{x \rightarrow y/s\}$
$\sigma_2 : \{y \rightarrow x/t\}$　■
【例5．5】
（ア）　$S : \{x,\ \{y\}\}$
$\sigma : \{x \rightarrow \{y\}\}$ これは、mguである。
（イ）　$S : \{\{x,\ a\},\ \{a\}\}$
$\sigma : \{x \rightarrow a\}$　これもmguである。
（ウ）　$S : \{\{x,\ a\},\ \{y\}\}$
$\sigma : \{x \rightarrow a,\ y \rightarrow a\}$　これもmguである。
（エ）　$S : \{\{a,\ b,\ x\},\ \{a,\ z\}\}$
$\sigma_1 : \{z \rightarrow b,\ x \rightarrow a\}$
$\sigma_2 : \{z \rightarrow b,\ x \rightarrow b\}$　ともにmguである。
（オ）　$S : \{\{a,\ x,\ y\},\ \{a,\ b\}\}$
$\sigma_1 : \{x \rightarrow a,\ y \rightarrow b\}$
$\sigma_2 : \{x \rightarrow b,\ y \rightarrow a\}$　ともにmguである。
（カ）　$S : \{\{a,\ x,\ y\},\ \{a,\ z\}\}$
$\sigma_1 : \{x \rightarrow z,\ y \rightarrow z\}$
$\sigma_2 : \{x \rightarrow z,\ y \rightarrow a\}$
$\sigma_3 : \{y \rightarrow z,\ x \rightarrow z\}$
$\sigma_4 : \{y \rightarrow z,\ x \rightarrow a\}$　この4つはmguである。
$\sigma_5 : \{x \rightarrow a,\ y \rightarrow a,\ z \rightarrow a\}$　mguではない。
（キ）　$S : \{\{a,\ x,\ y\},\ \{a\}\}$
$\sigma : \{x \rightarrow a,\ y \rightarrow a\}$　mguである。
（ク）　$S : \{\{a,\ b,\ x\},\ \{y\}\}$
単一化不可能。■

第一階述語論理では、mguが存在するならば、改名代入を除いて一意であった[30]。しかし、データ論理では例5．4、5．5（エ）、（カ）などで示されるようにmguは一意に定まらない。
［定義　最汎単一形］
節集合Sを $\{Q_i C_i,\ i=1..n\}$ とする。Q_iはA限量または基礎限量である。この時、単一化代入ρによる結果$Q_i C_i \rho$を単一形 (unified form) と言う。これらは定義よりすべて等しい。　2つの単一形P，Qに対してPがQより一般的 (general) とは、ある代入σに対して$Q = P\sigma$となるときをいう。Sの単一形で最も一般的なものを、Sの最汎単一形（mgf）と言う。■

［補題5．2］
（1）単一形があればmgfは存在する。
（2）mgfは存在すれば改名代入を除いて一意である。

（3）ρがmguならば、その単一形はmgfであり、逆も成立する。

データ論理でのmgu／mgfの計算はそれほど簡単ではない。直観的に言えば、複合項のうち順序組は成分毎に単一化すればよく、問題は集合項どうしの単一化にある。2つの集合$S_1=C_1 \cup V_1 \cup U_1$，$S_2=C_2 \cup V_2 \cup U_2$を考える。ここで$C_i$は基礎項集合、$V_i$は変数集合、$U_i$は変数を含む複合項集合とする。簡単のため、まず$U_i$が空とする（$i=1, 2$）。

基礎項集合が等しいならば変数集合を対応させる。即ちV_1、V_2に含まれる変数を1対1に対応させ、対応しない変数はどれかの変数にまとめて対応させる。C_1がC_2に含まれる（$C_1 \subseteq C_2$）ならば、V_1によってC_2-C_1に対応つける。対応しない変数はV_2に対応させればよい。対応しなかった変数は、どれかにまとめて対応させる。C_1がC_2と共通部を有さないとき、V_1によってC_2に、V_2によってC_1に対応つける。対応しなかった変数は、どれかにまとめて対応させる。

U_iが空でないとする。U_iの元は、「一部だけが基礎化」された項であるから、変数や基礎項の対応に失敗したとき、対応関係を解いてやり直せばよい。

［定義　一般導出形］

C_1，C_2を次の形の2つの節とする；

$C_1 : Q_1 \ (L \vee D_1)$　$C_2 : Q_2 \ (\neg L \vee D_2)$

このとき、L_1とL_2が単一化可能ならば、そのmgfを与えるmguをρとするとき、$Q_1 Q_2 \ (D_1 \vee D_2) \ \rho$を$C_1$，$C_2$の型付導出形（typed resolvent）という。型付導出形を得ることを型付導出原理（typed resolution）と言う。■

【例5．6】

$S = \{ \ (x/t_1) \ p(a) \vee q(x),$

$(y/t_2) \ \neg p(y), \ \neg q \ (f(a)) \ \}$

第一節と、第二節から　yにa／t_2を代入することで、次のような節を得る；　(x/t_1)　(a/t_2) $q(x)$

この結果と第三節から、xにf(a)／t_1を代入することで、空節を得る：　$(f(a)/t_1)$　(a/t_2) □

このことが意味することは、定数a及びf（a）が所定の型を有するならば充足不能となることであり、型条件を有した反駁法に相当する。■

5．2　導出原理の健全性と完全性

［定義　反駁］

節集合Sから導出原理を繰り返して型付空節　Q_0□　を得たとする。このとき、SはQ_0反駁（Q_0 Refutation）を持つ、または単に反駁を持つと言う。■

［定理5．3（型付導出原理の健全性）］

型制約Δは、節集合の限量子列Qによる型制約Δ_Qを含むとする。このとき、節集合Sが反駁を持てば、SはΔ充足不可能である。

次に、定理5．3の逆、即ちΔ充足不可能ならばQ_0反駁を持つことを言う。

［補題5．4（持ち上げ補題）］

2つの節C_1，C_2に対する代入θの結果をC_1'，C_2'、それらの導出形をC'とする。このとき、C_1，C_2の導出形Cと、$C' = C\rho$となる代入がある。

［定理5．5（型付導出の完全性）］

節集合SがΔ充足不可能ならば、SからのQ_0反駁が存在する。

6　結び

本稿では、データ論理によるデータモデルの形式化を行い、述語論に基づく操作系の提案を行った。データ論理は、多類論理を拡張し複合型を用いた構造情報の記述や操作を容易にする第一階言語である。他方、解釈（データベース）は主体と連想概念に基づく意味データモデルをベースにしており、主体は基礎項に、連想は基礎原子式の解釈に対応する。主体が複合値と対応するときは解釈の間で明示的な対応（λ対応）がなされている。一般の複合値は質問で生じるが、この解釈はデータベースから計算される。型に関する制約条件が与えられ、また連想に関する制約条件はデータ論理の閉式集合で表され、データベースはこれらのモデルとなる。操作系は、複合型に基づいた質問で記述される。データベースとの対応を単純にするため、正則形が定義される。任意の質問はいくつかの規則を用いて正則形に変換できる事を述べた。これらを通じて、べき集合操作がデータ論理での最も本質的な拡張機能を与える命令であることを理解することが出来る。

データ論理は、記述のための形式化だけでなく、導出原理に基づく形式論理として、健全且つ完全な証明手続きを有する。スコーレム化が充足不可能性を保存する変換であることや、エルブラン定理が成立する事を証明し、更に代入や単一化が拡張されて、型を扱うものとなった。この結果、導出原理が定義できた。

様々な拡張や応用や考えられる。演繹機能による拡張は複合述語で示したが、論理プログラムの観点から論じるべき事が多くある[32]。質問処理の最適化は、特に型制約や一般制約条件を活用したアプローチが考えられる。

謝辞

本稿構想時から、貴重な意見を数多く頂いた小林功武教授（産能大学）、有沢博助教授（横浜国立大学）に感謝します。新世代コンピュータ技術開発機構・演繹及びオブジェクト指向作業グループでの有益な議論にも強い影響を受けました。

参考文献

[1] Bancilhon,F.,Koshafian,S.:A Calculus for Complex Objects, PODS, 1985

[2] Bancilhon,F.:Object Oriented DataBase Systems, PODS,1988,152-162

[3]勝野裕文：演繹データベースの動向、データベースシンポジウム、１９８７

[4] Brodie,M.:On the Development of Data Models, in On Concuptual Modelling (ed.Brodie), 1984, Springer, 19-47

[5] Hull,R.,King,R.:Semantic Data Modelling, ACM Comp.Survey 19-3, 1987,201-260

[6] Ullman, J.D.: Database Theory, Past and Future, PODS, 1987,1-10

[7] Arisawa,H.,Miura,T.:Formal Approach to database Description, IEEE COMPCOM, 1984

[8] Arisawa,H.,Miura,T.: On the properties of extended Inclusion Dependencies, VLDB, 1986, 449-456

[9] Miura,T. and Arisawa,H.: Data Logic - Formalizing Data Models, maniscript

[10] Miura,T. and Arisawa,H: Logic Approach of Semantic Data Models(abstract),appears in PARBASE-90
[11] Koshafian, S.,Copeland,G.: Object Identity, OOPSLA, 1986
[12] Maier,D.: A Logic for Objects, Found.Deductive databases and Logic Programming, 1986
[13] Kim,W.,Ballou,N. et al:Integrating An Object Oriented Programming System with a Database System, OOPSLA, 1988, 142-152
[14] Kim,W.,Chou,H.-T.et al:Operations and Implementation of Complex Objects, IEEE Trans.On.Software Eng. 14-7, 1988, 985-996
[15] Batory,D.S.,Kim,W.:Modeling Concepts for VLSI CAD Object, ACM TODS 10-3, 1985, 322-346
[16] Kobayashi,I.: Classification and Transformations of Binary Relationship Relational Schemata, Info.Sys.11-2, 1987, 109-122
[17] Kobayashi,I.: Losslessness and Semantic Correctness of database Schema Transformation - Another Look of Schema Equivalence, Info.Sys.11-1, 1986, 41-59
[18] Beeri, C.: Data Models and Languages for Databases,ICDT, 1988
[19] Beeri, C.: Formal Models for Object Oriented Databases,DOOD,1989
[20] Codd, E.F.: Extending the database relational model to capture more meaning, TODS 4-4, 1979, 397-434
[21] Lecluse,C.,Richard,P.:Modelling Inheritance and Genericity in Object Oriented Databases, ICDT, 1988, 223-238
[22] Lecluse,C.,Richard,P.,Velez,V.:02 - An Object Oriented Data Model, SIGMOD, 1988, 424-433
[23] Cardelli,L,Wegner,P.:On Understanding Types, Data Abstraction and Polymorphism, ACM Comp.Survey 17-4, 1985, 471-522
[24] Abiteboul,S.,Hull,R.: Restructuring Hierarichical Database Objects, ICDT, 1986
[25] AitKaci,H.,Nauer,:LOGIN-A Logic programming language with built-in inheritance, J.Logic Prog.,1986
[26] Kifer,M.,Warren,D.S.: A Logic for Object Oriented Logic Programs, PODS, 1989
[27] Kifer,M.,Lausen,G.: F-Logic, SIGMOD, 1989
[28] Chen,W.,Warren,D.S.: C-Logic for Complex Objects, PODS,1989
[29] Bernays,P.,Schoenfinkel,M.:Zum Entscheidungs Problem der Mathematischen Logik, Mathematische Annalen,Vol.99,1928
[30] Chang,C-.L.,Lee,R.C-T.: Symbolic Loghic and Mechanical Theorem Proving, 1973, Academic Press
[31] 三浦孝夫、有沢博:意味データモデルの論理アプローチ、データ工学研究会、１９８９
[32] 三浦孝夫:演繹情報の非正規表現、情報処理学会論文誌、３０－９、１９８９
[33] 三浦孝夫: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