演繹情報の非正規表現†

三浦 孝夫††

本稿では，非正規関係技法を用いて，演繹情報を書き換え，コンパクトな表現を得るための手続きを考察する．事実 (EDB) がアプリオリに書き換えられたという仮定のもと，この情報を手がかりに規則集合 (IDB) の変更を行い，等価な仮想（演繹）情報を生成するメカニズムを論じる．IDB の非正規化のため，集合型論理プログラムを導入し，元のプログラムとの関係，特に最小（極小）モデルの対応があることを示す．また，拡張された不動点オペレータにより，非正規化された IDB の最小モデルが計算できることを述べる．

1. まえがき

近年演繹データベースの研究では，データベース的な側面特に関係データベース理論との関連を探ろうとすることが盛んである[1),2)]．再帰質問，否定を含む検索言語能力の拡大や質問処理の最適化などは，論理プログラミングおよび関係データベース理論の双方の観点から表現および計算量について考察されており，今後の発展が期待される．このうち演繹処理の性能向上については，性能の良い導出原理の採用という理論的な観点からのもの[3)]と，質問や（論理）プログラムのコンパイル，バックトラックの回避，構造共有，制約条件の活用など手続き的な観点からの提案がある[4),5)]．質問の解集合を等価に保ったまま探索空間を小さくするため，単一化の効率向上[6)]，事前に探索空間を求め明示的に指定するマジック集合技法やその拡張[7)]，変数の実体化の伝播[8)]など多数提案されている．本稿では，データ構造の活用という観点から集合型ホーン節プログラムを用い，直接探索空間のサイズ（事実の総数）を小さくすることを考える．

集合型論理プログラミングは，モデリング能力に加え，これまでの方法より細かな意図を記法上の操作で表し，表現能力の拡大と分かりやすさを目的としている．多層論理 (MLL) は，多類論理を拡張して，各類にべき集合化を行う特殊なオペレータを導入し，データベース探索を行わせる[9)]．LPS (Logic Program with Sets) では要素型と集合型という多類論理上に（集合の性質や集合間の関連性を記述できる）特殊な形のホーン式記述を与え，健全かつ完全な演繹処理が表現される[10)]．LDL (Logical Database Language) では内包型の集合記述が可能であり，否定の存在を含めての宣言的意味の研究が進んでいる[11)]．COL (Complex Object Language) は関数定義をホーン式自体で行い，さらに集合操作可能とした言語である[12)]．後の二つは集合生成規則を含むが，第2章で述べるようにLDL アプローチのほうが表現能力が大きく本稿でもこれに基づく．

ただ集合の構成規則を与えこれに基づく意味論を用いるならば，集合型論理プログラムではモデル交差性が成立せず，宣言的意味を一意に確定できない．また，内包型集合の構成は証明木が有限の深さ，幅のときだけ意味を持つため，演算的意味を与えることがきわめて難しい．

基礎原子式（事実）集合は同じ情報量を持つ集合化情報に変換されるが，その集合自体は意味を持たず，コンパクト化技法と考える（どのように効果的に変換するかは興味ある問題である）．コンパクト化された事実集合をそのまま活用するため，規則集合の書換えを行う．一般に，二つの論理プログラムの等価性は決定不能である[13)]が，ここで述べる書換え規則では特殊な部分クラスに限られ，同じ意味を有することが保証される．論理プログラムによる演繹情報は，正のホーン式プログラムならば最小モデルと考えられ，宣言的意味と呼ばれる．本稿で提案する変換規則の結果得られる集合型論理プログラムでも，この性質は成立する．最小モデルは拡張された不動点オペレータによって計算できる．前提部に否定を許す論理プログラムの場合でも，集合型論理プログラムの極小モデルは，元のプログラムの極小モデルと対応する．ただ，この逆は成立しない．

本稿と関連した研究が，非正規関係データベースで代数的な側面からなされている．非正規関係データベースを記憶域モデルと考え，射影による分解（正規化）と逆に外部結合によって一つのテーブルにまとめる（逆正規化）．このとき生じる冗長性は関係代数（ネ

† Unnormalizing Deductive Information by TAKAO MIURA (Department of Management and Informatics, Sanno College).
†† 産能大学経営情報学部

スト操作）で減少させ，また，空値を用いて部分情報を表現する[14)~17)]. 概念スキーマと記憶域スキーマが同じモデル上で展開され，概念変換が不要である．検索効率も明らかによい．反面，階層的な情報表現しか扱えず，ネスト操作でカバーできない場合（多値従属性や，キー破壊など）があったり，空値の導入が必要となる．また，データ従属性は逆正規化で捕らえやすくなったが，これを保存するメカニズムが考えにくく，さらにこの議論は演繹処理への展開に用いることができない．

本稿の構成は次のものである．第2章では，動機と非公式な説明を述べ，第3章で形式化を行う．第4章は，対応する二つのプログラムに成立するモデル間の対応付けを論じる．特に，正ならば最小モデルが対応し，第5章ではその計算方法を示す．第6章は要約である．

2. 論理プログラムの変換：動機と非公式な説明

この章では，動機となる例をあげ，非形式的プログラムを書き換えることで，基本となる考え方を述べる．形式化およびこれに基づく性質の解析は次章で行う．

2.1 外延情報の変換

非正規関係の利点のひとつは，それを記憶域モデルと考えることができることで，タプルに集合項を導入し複数のタプルをひとつに表すコンパクト化と見てよい[18),19)]．この考え方を論理プログラムにも適用する．ここでは集合化の対象となる基礎項は集合型に変換したと考える．つまり，述語の各項は要素型か集合型のいずれかとなる．集合化の方法は問わない．

［例 2.1.1］

E_1= {parent (adam, cain), parent (adam, abel), parent (cain, enoch), parent (adam, seth)}

E_2= {PARENT(adam, {cain, abel, seth}), PARENT(cain, {enoch})}

E_3= {PARENT(adam, {cain, abel}),
/＊初期の息子 ＊/
PARENT(adam, {abel, seth}),
/＊ 素直な息子 ＊/
PARENT(cain, {enoch})}

個体述語 parent に対し，対応する集合述語を大文字 PARENT と表している．

E_1 を，可能な限り同じ親で重ね合わせれば（いわゆるネスト操作で）E_2 を得る．E_3 はむしろ現実の意味から得たもので，集約関数を外延的に与えたものと考えられ，また重複した基礎原子式を含む．代数的には取扱上問題だが宣言的意味としては扱いやすい．■

外延情報をこのように変換するとき，次の規則を加え元の個体述語を仮想化してしまったと考える（復元ルールと呼ぶ）；

$$p(x_1\cdots x_n)\leftarrow P(X_1\cdots X_n),\ x_1\in X_1, \cdots, x_n\in X_n$$

ここでは，p の第1項から第 n 項までが集合化されたとする．このルールによって E_2, E_3 から E_1 に復元可能である．なお述語 p に関するすべての事実は集合型述語 P により集合化し，基礎例としては残さないとする．

2.2 演繹規則の変換

復元ルールを用いて明示的な事実（個体項のみを含むため，個体事実とも言う）を再生成すれば，これまでの演繹処理が行えるが，ここでは集合を用いて変換された事実をそのまま演繹処理で扱えるように，規則集合を書き換えることを考える．

個体述語 p が事実内に生じ，集合化されて P に変換されたとする．述語 p は規則に生じていてもよい．まず，p が規則の前提部に生じる場合を考える．前提部にある $p(x_1\cdots x_n)$ を，次のように変更する；

$$P(X_1\cdots X_n),\ x_1\in X_1, \cdots, x_n\in X_n$$

ただし，p の第1項から第 n 項までが集合化されたとしている．この規則をボディルールとする．

［例 2.2.1（ボディルール)］

述語 p の第1，第2項が集合化されたとする．

/＊結合＊/
$r(xz) \leftarrow p(xy), q(yz)$
$r(xz) \leftarrow P(XY), x\in X, y\in Y, q(yz)$
/＊選択＊/
$q(x) \leftarrow p(xa)$
$q(x) \leftarrow P(XY), x\in X, a\in Y$
/＊交差＊/
$r(xy) \leftarrow p(xy), q(xy)$
$r(xy) \leftarrow P(XY), x\in X, y\in Y, q(xy)$
/＊差＊/
$r(xy) \leftarrow p(xy), \neg q(xy)$
$r(xy) \leftarrow P(XY), x\in X, y\in Y, \neg q(xy)$ ■

最後の例で，否定を含んだリテラルが集合化されていない．否定を含む項を集合化すればこの変換は等価にならない．

［例 2.2.2（否定）］

述語 r が集合型述語 R に変換されたとする.

$H=\{p(x)\leftarrow q(x), \neg r(x)\}$

$G=\{p(x)\leftarrow q(x), \neg R(X), x\in X\}$

$M=\{q(a), q(b), r(a), p(b)\}$

このとき，r を集合化して変換結果 N を得る.

$N=\{q(a), q(b), p(b), R(\{a\})\}$

H の個体述語 r が集合化されて R になったとし，またそのモデル M を N に変換する．G において，$R(X)$ を偽とする代入 $X=\{a, b\}, x=a$ は正解代入となり，$p(a)$ が得られるが N には含まれず，モデルでない. ■

［例 2.2.3（関数項）］

述語 p の第1，第2項が集合化されたとする.

$q(xy)\leftarrow p(f(x)y)$

$q(xy)\leftarrow P(XY), f(x)\in X, y\in Y$ ■

変換対象となる個体述語 p を結論部に持つ規則が存在したとする．この規則の役割は，明示的な事実ではないが論理帰結として得られる事実を生成することにある．これを仮想事実と呼ぶ．事実集合の変換は仮想事実まで及ばないため，この規則で独自の（仮想事実の）集合化を行うと考える．ボディルールを結論部に適用すると生成される仮想事実は異なったものとなって対応しない.

［例 2.2.4（正しくないボディルール）］

述語 p の第1，第2項が集合化されたとする.

$p(xy)\leftarrow q(x), r(y)$

$P(XY), x\in X, y\in Y\leftarrow q(x), r(y)$

この意味は次と考えるべきである；

$(\forall x)(\forall y)(\forall X)(\forall Y)P(XY),$

$x\in X, y\in Y\leftarrow q(x), r(y)$

いわば X は x, y の（集合値）関数で，$f(xy)$ と陽に表してもよい．同様に Y は $g(xy)$ に対応する.

$P(f(xy)g(xy))\leftarrow q(x), r(y)$ ■

$f(xy), g(xy)$ の定義はアプリオリに与えられているのではなく，構成ルールを特定することができる．例 2.2.4 では，前提部を満たすすべての (x, y) を集めて $f(xy), g(xy)$ とすればよい．COL[12]は，関数定義を規則で与えることを提案している．上の例では，f, g は次のようになる；

$x\in f(xy)\leftarrow q(x), r(y).$

$y\in g(xy)\leftarrow q(x), r(y).$

関数記号および述語記号に関して，次のように順序を導入する；結論部にある述語記号および定義されている関数が，前提部で負として生じる述語および $t\in f(x_1\cdots x_n)$ 以外の形で出現する関数より大きいとする．このようにして定義されたプログラムは関数および否定に関して層状化されている（すなわち，すべての関数記号および述語記号がこの順序で矛盾なく整列化できる）ときに層に関する順序に依存した最小（極小）モデルを持つことが知られており，本稿には適用できない．次の例は層状でないため COL で意味を与えることができないことを表している.

［例 2.2.5（COL の限界）］

parent の第2項が集合化されたとする.

ans$(xy)\leftarrow$ parent(xy).

ans$(xy)\leftarrow$ parent(xz), ans(zy).

$\mathrm{ANS}(xf(y))\leftarrow \mathrm{PARENT}(xY), y\in Y.$

$\mathrm{ANS}(xg(y))\leftarrow \mathrm{PARENT}(xZ), z\in Z,$

$\mathrm{ANS}(zY), y\in Y.$

このとき，述語 ANS と関数 g は層状でない．実際，次のサイクルがある；$\mathrm{ANS}>g$, $g\geq \mathrm{PARENT}$, $g\geq \mathrm{ANS}$ ■

本稿では，LDL と同様，規則の形式を定め前提部を満たす個体要素の集合を求めるという，グループ化による解釈だけを考える．特に，$f(x)$ とは表さず $\langle x\rangle$ の形の集合変数を考える．前提部を充足する x への代入を集めて $\langle x\rangle$ に代入するという規則を，ヘッドルールという．もし結論部に二つ以上の集合変数があるときは，LDL と同様左から順に集合化可能な組合せを構成すると考える．このため議論は結論部にはただひとつの集合変数しか生じないとしてよい.

［例 2.2.6（ヘッドルール）］

ans$(xy)\leftarrow$ parent(xy).

ans$(xy)\leftarrow$ parent(xz), ans(zy).

$\mathrm{ANS}(x\langle y\rangle)\leftarrow \mathrm{PARENT}(xY), y\in Y.$

$\mathrm{ANS}(x\langle y\rangle)\leftarrow \mathrm{PARENT}(xZ), z\in Z,$

$\mathrm{ANS}(zY), y\in Y.$

前提部を満たす x, y に対してこのような y だけを集めて集合化する. ■

復元ルールは集合化の効果をなくすものであるから，可能な限りこれを回避するような工夫が望まれる．個体述語が集合化されるべき項を共有すれば復元ルールの適応を行わざるを得ないから，逆にこの述語を集合化することを考えればよい．これを集合化の伝播ルールと呼ぶ.

［例 2.2.7（伝播ルール）］

/*結合*/

$r(xz) \leftarrow p(xy), q(yz)$

$R(\langle x \rangle z) \leftarrow P(X_1 Y_1), x \in X_1, y \in Y_1,$

$Q(Y_2 z), y \in Y_2.$ ■

ところが，例2.2.6の第4式は，前提部にANSを含み，第1項で集合化できる．この結果，第3式でANSは，第1項に関して（ヘッドルールにより）集合化せねばならず，この手間が増える．伝播ルールはこの意味で無条件に適用するわけにはいかず，プログラム設計上の考慮が必要となる．

3. 形 式 化

3.1 諸 定 義

以下では，集合論理プログラムの形式的な定義を行う．なお論理プログラムの一般的な知識は仮定する[3]．

アルファベットとは個体定数 $a, b, \cdots$，個体変数 $x, y, \cdots$，個体関数 $f, g, \cdots$ および集合変数 $X, Y, \langle x \rangle, \langle y \rangle, \cdots$ 述語記号 $p, P, \cdots$ および結合子からなる．特別な述語∈および特別な関数 $\{_n, n=0, 1, \cdots$ が用意されている．述語の引数は要素型，集合型のいずれかで型付けされている．項は，個体項 $t_1, \cdots$ および集合項 $T_1, \cdots$ からなる．個体項とは，個体変数，個体定数および個体関数 f による $f(t_1 \cdots t_n)$，ただし t_i は個体項，の形をいう．集合項とは，変数を含まない個体項 $a_1, \cdots, a_n$ に対して $\{_n(a_1 \cdots a_n)$ または集合変数をいう．$\{_n(a_1 \cdots a_n)$ は，$\{a_1, \cdots, a_n\}$ とも表す．原子式は，$p(t_1 \cdots t_n)$ または $P(T_1 \cdots t_n \cdots)$ の形の式をいう．変数を含まない原子式を基礎原子式，または事実という．リテラルとは原子式 A に対し，A または$\neg A$ のかたちをいう．整式は，リテラルまたは整式のブール結合子（$\vee, \wedge, \neg$）および限量子（$\exists, \forall$）による表現をいう．規則は，$A \leftarrow B_1, \cdots, B_m$ の形の式（これは，$A \vee \neg B_1 \vee \cdots \vee \neg B_m$ の省略形）で A は原子式，B_i はリテラルを表し，すべての変数は全称限定されているとする．A を結論部（ヘッド），残りを前提部（ボディ）と呼ぶ．プログラムとは規則の有限集合で，特にどの B_i も原子式のとき，確定的または正という．どの述語の引数も個体型で集合項が生じていないプログラムを，個体（型）プログラム，そうでないとき集合プログラムという．

解釈 $\langle D, M \rangle$ とは，空でない集合 D と次のような写像 M からなる（M とかく）;

個体定数 c に対して $M[c]$ は D の要素

関数 f に対し $M[f]$ は $D^n \rightarrow D$ なる写像

個体述語 p に対し $M[p]$ は $D^n \rightarrow \{0, 1\}$ なる写像

集合述語 P に対し $M[P]$ は $E^n \rightarrow \{0, 1\}$ なる写像，ただし E は引き数の型に応じて D または 2^D と解釈される．

$M[\in]$ は，メンバシップ述語で値1（真）または0（偽）となるものを対応させる．$M[\{_n]$は，引数すべてを集合とする関数 $D^n \rightarrow 2^D$ とする．

個体変数および集合変数に対する代入とは，それぞれ D の要素，D の部分集合への割当をいう．

変数代入と解釈 M に関する評価 θ とは次で定義される；

変数に対しては（普通の）代入となる

$\theta(c)=M[c]$

$\theta(f(t_1 \cdots t_n))=$
$M[f](\theta(t_1) \cdots \theta(t_n))$

$\theta(\{a \cdots\})=\{\theta(a) \cdots\}$

$\theta(p(t_1 \cdots t_n))=$
$M[p](\theta(t_1) \cdots \theta(t_n))$

$\theta(P(\{a \cdots\} \cdots))=$
$M[P](\theta(\{a \cdots\}) \cdots)$

$\theta(t \in T)=1$ もし $\theta(t)$ が $\theta(T)$ の要素，さもないとき0となる．

$\theta(\neg A)=\neg\theta(A)$

$\theta(A \vee B)=\theta(A) \vee \theta(B)$ $\wedge$も同様

解釈 M に対して，整式の充足性は次で定義される；

（i） $p(a_1 \cdots) \dashv M \Leftrightarrow M[p](M[a_1] \cdots)=1$

$P(\{a \cdots\} \cdots) \dashv M \Leftrightarrow M[P](M[\{a \cdots\}] \cdots)=1$

（ii） $t \in T \dashv M \Leftrightarrow M[T]$ が集合でかつ $M[t]$ が $M[T]$ の要素となる

（iii） $A \leftarrow B_1 \cdots B_n \dashv M \Leftrightarrow$ 二つの場合に分ける；

A が $\langle t \rangle$ の形の変数を含まないとき

すべての評価 θ に対して，どの i についても $\theta(B_i)=1$ ならば $\theta(A)=1$

A が $\langle t \rangle$ の形の項を含む（t が変数 x を含む）とき

すべての評価 θ に対して，どの i についても $\theta(B_i)=1$ ならば，次を満たすように θ を拡張して $\theta(A)=1$ とできる；高々 t でだけ θ と異なり，どの k についても $\eta(B_k)=1$ となる任意の代入 η に対して，$\eta(t)$ が $\theta(\langle t \rangle)$ の要素となる，ただしこのような η が無限個存在すれば常に成立するとする．

（iv） $\neg r \dashv M \Leftrightarrow r \dashv M$ でない

（v） $H \dashv M \Leftrightarrow H$ のすべての元 r に対して $r \dashv M$

（vi） M が H のモデル $\Leftrightarrow H \dashv M$

LDL と同様に，従来のエルブラン領域に加えて有限集合を項に持つように拡張する．これは，可付番無

限領域であり U とかく．またエルブラン基底を B とかく．Mod(H) を集合プログラム H のすべてのモデルの集合とする．H が充足可能とは Mod(H) が空でないことをいう．Mod(H) の元 M が極小（最小）とは，M の真部分集合で H のモデルとなるものはない（極小でただひとつしかない）ときをいう．Mod(H) $\wedge B$ の元をエルブランモデルと呼ぶ．

整式 W が H の論理帰結とは，H のすべてのモデル M に対し $W \dashv M$ のときをいう．

次の性質から，Mod(H) はエルブランモデルの集合に限定してよい．

［定理1］（集合）プログラム H が充足可能ならば，H は（集合型）エルブランモデルを持つ．

（証明） H の任意のモデル M を考える．このとき，エルブラン解釈 M_0 を次のように与える．

$$M_0=\{A: 基礎原子式，ただし\ M[A]=1)$$

この解釈 M_0 が H のエルブランモデルになっていることをいう．明らかに $\langle t\rangle$ 型の変数の処理だけが問題である．

規則 $A \leftarrow B_1, \cdots, B_n$ がこの適用を受けているとする．前提部を満たすすべての個体項の集合 S を，結論部の $\langle t\rangle$ 型変数に代入するとき，M_0 内で S が有限集合となれば問題ない．

S が M では有限だが，M_0 で（関数計算などにより）無限個となることがある．しかし，このときでも基礎代入が（結果的には）無限個存在して，規則自体は成立する．（証明終わり） ■

H のモデル M に対して，p の M での表示 $D(p, M)$ とは次をいう；

$$D(p, M)=\{p(a_1\cdots a_n)\ ただし\ M\models p(a_1\cdots a_n)\}$$

論理プログラムGがクラスタ化されているとは，どの規則 $A \leftarrow B_1, \cdots, B_n$ も次の四つを満たすときをいう；

（i） A が $\langle t\rangle$ の形の項を含めばその中のすべての変数 x は $B_1\cdots B_n$ のどこかで生じる．

（ii） 前提部の集合変数 X は，ちょうどひとつの非メンバシップリテラルにあり，かつひとつ以上のメンバシップリテラルに生じる．

（iii） 負リテラルには集合変数は生じない．

（iv） $\langle x\rangle$ 型の変数は結論部に，X 型の変数は前提部にだけ生じる．

3.2 プログラム変換

第2章で述べた変換を形式的に記述する．（個体）論理プログラム H に次の四つの変換ルールを定義する．個体述語 p が H に生じているとする．P を p に対応する（つまり事実集合の基礎原子式がこれにしたがって非正規化された）集合述語とする．変数 $x_1\cdots x_n$ はどこにもないとする．

［復元ルール］ 次の規則を加える；

$$p(x_1\cdots x_n) \leftarrow P(X_1\cdots X_n), x_1\in X_1, \cdots, x_n\in X_n$$

［ボディルール］ H の規則 $A \leftarrow B_1, \cdots, B_n$ において，B_j が正の基礎原子式 $p(t_1\cdots t_n)$ ならば，B_j を $P(X_1\cdots X_n)$ で置き換え，かつ次を前提部の最後に加える；$t_1\in X_1, \cdots, t_n\in X_n$

［ヘッドルール］ 結論部 A が $p(t_1t_2\cdots t_n)$ の形で，A のどの変数も前提部にあれば，これを $P(\langle t_1\rangle\cdots\langle t_m\rangle\cdots t_n)$ に書き換える．ただし $t_1\cdots t_m$ のどれも変数を含み，述語 p の第1項から第 m 項までが集合化され P になったとする．

［伝播ルール］ p に関するリテラルが書き換えられたとき，書換えの対象となった項を別の述語 q が（同じ規則の正のリテラル内で）含むとする；$p(t_1\cdots t_n)$，$q(t_m\cdots s_k)$ このとき，q の（t_m を含む）第1項に対してこれまでのルールを適用する． ■

このとき，定義から次の定理が成立する．

［定理2］ 上記四つのルールで得られるプログラムGはクラスタ化されている．

3.3 モデルの変換

個体プログラム H と，前節の変換で得られるクラスタ化プログラムGの間のモデル対応を考える．

［変換ルール］ Mod(H) の元 M から，G の解釈 N を次のようにつくる；

述語 p が集合化されていなければ，$N[p]=M[p]$ とする．そうでなければ，集合 $T_1, \cdots, T_n$ に対して，$N[P(T_1\cdots T_n)]=1$ となるのは次のときだけであるとする；すべての i に対して，T_i の任意の元 t_i について $M[p(t_1\cdots t_n)]=1$． ■

［例 3.3.1］

$$H=\{p(x) \leftarrow q(x)\}, G=\{P(\langle x\rangle) \leftarrow q(x)\}$$

$$M=\{p(a), p(b), q(a)\}.$$

p が P に変換されたとき，変換ルールの結果 N は次である；

$$N=\{P(\{ab\}), P(\{a\}), P(\{b\}), q(a)\}$$ ■

この変換ルールで $N[P(T_1\cdots T_n)]=1$ であれば，すべての $S_1\subseteq T_1\cdots S_n\subseteq T_n$ に対して，$N[P(S_1\cdots S_n)]=1$ となる．ただし，どの S_i も空でない．

［逆変換ルール］ Mod(G) の元 N から，H の解釈 M を次のようにつくる；

述語 p が集合化されていなければ，$M[p]=N[p]$

とする．そうでなければ，$M[p(t_1\cdots t_n)]=1$ となるのは，ある集合 $T_1\cdots T_n$ に対して次を満たすときである；$N[P(T_1\cdots T_n)]=1$，かつどの i についても t_i は T_i の元． ■

次の定理は変換で得るプログラムとそれらのモデルとの対応を示している．ただし，例 2.2.2で見たように否定を含むリテラルに対しては成立しない．

［定理 3］

（a） 変換ルールで得られた N は G のモデルで，かつどの個体述語 p についてもその表示は同じである．

（b） 逆変換ルールで得られた M は H のモデルで，かつどの個体述語 p についてもその表示は同じである．

（証明：a）

G の規則 r' を $A\leftarrow B_1,\cdots,B_n$，$N$ に関する代入 Θ が任意の i に対して $\Theta(B_i)=1$ とし，$\Theta(A)=1$（または Θ の拡張についてこれが成立）をいう．帰納法的に各変換ルールでこの条件が成立することをいう．

復元ルール：$p(x_1\cdots x_n)\leftarrow P(X_1\cdots X_n), x_1\in X_1,\cdots, x_n\in X_n$.

N の作り方から，$N[x_i\in X_i]=1$ が成立し，$N[x_i]$ を a_i と表すとすべての $N[X_i]$ の元 a_i について，M から $p(a_1\cdots a_n)$ が含意される．したがって，$N[p(x_1\cdots x_n)]=1$

ボディルール：$A\leftarrow B_1,\cdots,P(X_1\cdots X_n),\cdots,x_1\in X_1,\cdots,x_n\in X_n$.

H の規則 r の前提部 B_i が変換されたとする．M はモデルゆえ，復元ルールと同様に個体変数 x_i への代入はそのまま N に引き継がれている．よって $N[A]=1$.

ヘッドルール：$P(\langle t_1\rangle\cdots)\leftarrow B_1,\cdots,B_n$.

項 t_1 に生じているすべての変数は前提部にも生じているので，代入が生じ M がモデルゆえ，このようなすべての基礎代入に対して，$M[p(t_1\cdots t_n)]=1$ となる．

Nは，このようなものだけを（有限個）集めて $\langle t_1\rangle\cdots$ への代入とするから，この式を満たす．

伝播ルール：これらを組み合わせればよい．

（証明：b）

Mが H のモデルになることをいう．H の規則 r を $A\leftarrow B_1,\cdots,B_n$，$M$ についての代入 θ が任意の i に対して $\theta(B_i)=1$ とし，$\theta(A)=1$ をいう．$n=0$ なら自明なので，$n>0$ とする．対応する G の規則 r' を $J\leftarrow K_1\cdots K_m$ とする（$m\geqq n$）.

［持ち上げ補題］ G の Nについての代入 Θ で，個体項 t については $\Theta(t)=\theta(t)$ で，どの i についても $\Theta(K_i)=1$ となるものが存在する．

（補題の証明） G はクラスタ化されており，K_i 中の任意の集合変数 X は r' では K_i が $x\in X$ としても生じている．$K_i=P(X_1\cdots X_n)$ と $B_i=p(t_1\cdots t_n)$ が対応しているとすると，仮定から $\theta(B_i)=1$ ゆえ，M の作り方より次が成り立つ；ある集合基礎項 A_k があって，$\theta(t_k)=a_k\in A_k$ かつ $N[P(A_1\cdots A_n)]=1$.

このとき，Θ として変数 X_k に A_k を割り当てる代入を考える．$\Theta(t_k\in A_k)$ は自明に成立する．X_k はこれ以外には用いられていないから，各 B_i について同様に考えれば Θ を得る．（補題証明終わり）

（定理証明の続き） N はモデルだから，$\Theta(J)=1$. A に集合変数 $\langle t\rangle$ が生じていなければ，定義から $\Theta(A)=\theta(A)=1$. ヘッドルールとして生じているとする．A の個体変数はすべて前提部にあり，$\langle t\rangle$ への代入の性質から $\theta(A)=1$.（証明終わり） ■

4. 最小モデル

4.1 最小モデルの定義

集合プログラムGはこれが正であっても複数の極小モデルを持つ．次の例でこれがわかる．

［例 4.1.1］

$H=\{p(xy)\leftarrow q(x),r(y).\ q(a).\ r(b).\}$

$G=\{P(\langle x\rangle y)\leftarrow q(x),r(y).\ q(a).\ r(b).\}$

$M=\{q(a),r(b),p(ab)\}$

$N_1=\{q(a),r(b),P(\{a\}b)\}$

$N_2=\{q(a),r(b),P(\{ab\}b)\}$

M は H の最小モデルであるが，G のモデル N_1,N_2 は共に極小である． ■

二つの集合型基礎原子式 $A=P(T_1\cdots T_n)$，$B=P(S_1\cdots S_n)$ に対して，A が B より弱い（$A\leqq B$）とは，どの i についても $T_i\subseteq S_i$ のときをいう．A が真に B より弱い（$A\langle B$）とは，$A\leqq B$ でどれかの i で $T_i\subset S_i$ のときとする．解釈 $M1$ が解釈 $M2$ より弱い（$M_1\leqq M_2$）とは，M_1 の任意の元 A に対してもある B が M_2 にあり $A\leqq B$ を満たすときをいう．解釈 M が縮退しているとは，$A\leqq B$ を満たす基礎原子式 A,B を M が含まないときとする．極小モデルで，縮退していないものがある．

［例 4.1.2］

$H=\{p(xy)\leftarrow q(x),r(y).$

$p(ab).\ q(a).\ q(b).\ r(b).\}$

$G=\{P(\langle x\rangle y)\leftarrow q(x), r(y).$

$P(\{a\}b). q(a). q(b). r(b).\}$

$N_1=\{P(\{ab\}b), P(\{a\}b), q(a), q(b), r(b)\}$

$N_2=\{P(\{ab\}b), q(a), q(b), r(b)\}$

N_1 は極小モデルだが縮退していない．N_2 は縮退しているがモデルでない． ■

このため集合プログラムでのモデル N からの含意性を次のように拡張する；

$N[P(T_1\cdots T_n)]\approx 1 \Leftrightarrow$ ある $P(S_1\cdots S_n)$ で N で真であり，かつどの i についても $T_i\subseteq S_i$，かつ $T_i\neq\phi$.

定義より $=$ の意味で真ならば $\approx$ の意味でも真だから，これまでの充足性の拡張となっている．例えば，定理3はそのまま成立する．以下，本稿では新しい意味で考える．特に，極小性は≦で考えることになる．上記 N_2 は（拡張した意味での）極小モデルとなる．

［定理4］ 極小モデル M は縮退している．

（証明） そうでないとせよ．M は $A\leqq B$ となる基礎原子式を充足しており，A を取り去っても A は真（1）と解釈される．これは極小モデルの定義に反する．（証明終わり） ■

［定理5］ 個体プログラム H が正であれば，対応するクラスタ化プログラム G はただひとつの極小モデル（つまり最小モデル）を持つ．

（証明） G がモデル交差性を有することをいう[3]．すなわち，G の二つのモデル M_1, M_2 に対し，$M_0=M_1\wedge M_2$ が G のモデルになることをいう．ただし M_0 は次で与えられる；

$\{P(x_1\cdots x_n)$ ただしどの i についても

$x_i=b_i\cap c_i\neq\phi$,

$M_1[P(b_1\cdots b_n)]=1,\ M_2[P(c_1\cdots c_n)]=1\}$

このとき，すべての G のエルブランモデル M の $\wedge$ について交差 $(\wedge M)$ は G の最小モデルとなる．

G の規則 $A\leftarrow B_1\cdots B_m$ を考える．

A が個体原子式 $q(x_1\cdots x_n)$ とする．すべての i について $\theta(B_i)=1$ とするとき，$\theta(A)=1$ をいう．$B_i=p(X_1\cdots X_k)$. $\theta(B_i)=\theta(p(a_1\cdots a_k))$ とせよ．また，$M_1\models p(b_1\cdots b_k), M_2\models p(c_1\cdots c_k)$ かつどの i も $a_i=b_i\cap c_i$ を満たすとする．このとき，（定理3証明中の）持ち上げ補題と同様に，M_1 での代入 ζ で，どの i についても X_i に b_i を割り当てるものがある．また $x_i\in X_i$ は θ で1となるので，ζ でも1．M_1 は G のモデルゆえ $\zeta(A)=1$. M_2 での代入 η も同様．θ での個体代入は ζ, η のそれと同じだから $\theta(A)=1$.

A が集合型原子式 $Q(\langle t_1\rangle\cdots\langle t_n\rangle)$ とする．上と同様 M_1, M_2 での各代入 ζ, η を考える．$\zeta(A)=Q(e_1\cdots e_n)$, $\eta(A)=Q(f_1\cdots f_n)$ とせよ．このとき $Q(e_1\cap e_1\cdots e_n\cap f_n)$ は M_0 において真となり，$\theta(A)$ をこのように拡張してよい．元の個体プログラム H ではどの結論部の個体変数も前提部にあるとしているから $B_1\cdots B_m$ で生じ，$e_i\cap f_i=\phi$ となることは起こらない．（証明終わり） ■

4.2 最小モデルの対応

H の最小モデル M に対して，対応する G の最小モデル N でどの述語についても同じ表示を有してもモデル変換ルールで得られるとは限らない．

［例 4.2.1］

$H=\{p(ac). p(bc). p(ad). p(be).\}$

$G=\{P(\{a\}c). P(\{b\}c). P(\{a\}d).$

$P(\{b\}e).\}$

このとき，H（および G）の最小モデル M は H 自身で，G のモデル変換後の結果 N は次である；

$N=\{P(\{b\}e), P(\{a\}d), P(\{ab\}c)\}$ ■

必ずしも正とは限らないプログラム H では定理5は成立しないが，その変換 G については次の結果が得られる．

［定理6］（正とは限らない）G の極小モデル M' に対して，モデル逆変換ルールで得られる M は H の極小モデルとなる．

（証明） 定理2より M は H のモデルである．極小でないとし，M の真部分集合 N で H のモデルとなるものがあるとする．$M-N$ の元 $q(a_1\cdots a_n)$ に対して，M' の元 $Q(e_1\cdots e_n)$ がこれを含むとする．このとき次のような高々 $n+1$ 個の元への分割を行う；

（i） $Q(\{a_1\}\cdots\{a_n\})$

（ii） $Q(e_1\cdots e_{i-1}f_ie_{i+1}\cdots e_n)$

$e_i=\{a_i\}\cup f_i\ \ i=1\cdots n$,

ただし $f_i\neq\phi$ とする

二つの種類の元の和の q に関する表示は $Q(e_1\cdots e_n)$ と同じである．G の解釈 N' を次のようにする；

すべての $M-N$ の元に対して可能ならば上記のように分割する．（i）の形のものを取り去り，残った高々 n 個の元と置き換える．これを可能な限り続け，最終結果を N' とする．

N' の個体述語の表示は N と等しい．N' は $M-N$ の元を含まない M' の部分集合としては極大である．また $N'<M'$. N' が G のモデルになることをいう．これより，M' の極大性に矛盾する．

G の規則 $A \leftarrow B_1, \cdots, B_m$ を考える．

A が個体原子式 $q(x_1 \cdots x_n)$ とする．すべての i について $\theta(B_i)=1$ とするとき，$\theta(A)=1$ をいう．$B_i=p(X_1 \cdots X_k)$．$\theta(B_i)=\theta(p(a_1 \cdots a_k))$ とせよ．他の B_i で $x_k \in X_k$ の形が存在している．すべての k について前提部の X_k を x_k で書き換え，θ による個体変数への代入をそのまま用いた新しい代入を ζ とすれば，矛盾なく定義できて N 上での代入になり，$\zeta(A)=1$．特に N' でも（ζ と同じなので）1となる．

A が集合型原子式 $Q(\langle t_1 \rangle \cdots \langle t_n \rangle)$ とする．すべての i について $\theta(B_i)=1$ とすると上と同様に代入 ζ を考えることができる．また（定理3証明中の）持ち上げ補題と同様に，N' の作り方から M' の代入 η で次を満たすものが構成できる；すべての i で $\theta(B_i) \leqq \eta(B_i)=1$，かつ個体変数への代入は θ と同じ．

M' は G のモデルであるから，η を拡張して $\eta(A)=\eta(p(e_1 \cdots e_n))=1$ とできる．各 e_i は $\langle t_i \rangle$ への代入とする．規則の前提部は t_i の変数を含むから，N がモデルという仮定から，$\zeta(p(t_1 \cdots t_n))=1$．ただし，$p$ は p に対応する H の述語である．このような $t_1 \cdots t_n$ に対するすべての代入結果（有限集合）を $f_1 \cdots f_n$ とする．定義から，すべての i について f_i は a_i の部分集合である．

$M-N$ の元 $p(a_1 \cdots a_n)$ が $p(e_1 \cdots e_n)$ に含まれなければ，$p(f_1 \cdots f_n)$ には存在しない．N が H のモデルで真部分集合だから，$N' \models p(f_1 \cdots f_n)$．（証明終わり）

■

5. 最小モデルの計算

正のプログラム H および対応する G が与えられたとき，G の最小モデル N を不動点オペレータ T_G によって計算する．オペレータ T は次のように定義される；

I が G 上のエルブラン基底 B の部分集合とすれば，$T(I)=\{A' \in B, A'$ は G の I による計算結果$\}$

ただし，A' が G の I による計算結果とは，$A \leftarrow B_1 \cdots B_m \in G$ があって，次を満たす．

(i) A が $\langle t \rangle$ の形の変数を含まないとき，I についてのある基礎代入 θ で $\theta(B_i)=1$ がすべての i で成立し，かつ $A'=\theta(A)$．

(ii) $\langle t \rangle$ が含まれれば，I についての基礎代入 θ で $\theta(B_i)=1$ がすべての i で成立し，高々 t でだけ異なる I についてのすべての代入 θ' で $\theta'(B_i)=1$ が各 i で成立するとき，$\theta'(t) \in \theta(\langle t \rangle)$ を満たし，$\theta(\langle t \rangle)$ はちょうどこのような元だけからなっていて，$A'=\theta(A)$． ■

T は⊆について単調ではない．実際，次の反例がある；

$I=\{p(a)\}$．$J=\{p(a), p(b)\}$

$G=\{Q(\langle x \rangle) \leftarrow p(x).\}$

$T(I)=\{Q(\{a\})\}$，$T(J)=\{Q(\{ab\})\}$

［補題7］ T は≦について単調，特に連続である．

（証明） $I \leqq J$ としたとき，$T(I) \leqq T(J)$ をいう．$p(e_1 \cdots e_n)$ が $T(I)$ の元とし，規則 $p(\langle x_1 \rangle \cdots \langle x_n \rangle) \leftarrow B_1 \cdots B_m$ が与えられているとせよ．I に関するどの代入 θ についても $\theta(B_i)=1$ ならば J に関する代入とみて $\theta(B_i)=1$ が成り立つから，定理3証明中の持ち上げ補題と同様 J 上の代入 θ' で，どの i についても $\theta(B_i) \leqq \theta'(B_i)$ が成り立つ．定義から，$p(e_1 \cdots e_n)=\theta(A) \leqq \theta'(A)$ なので，$T(I) \leqq T(J)$．

≦に関して半順序となっていることをいう．

$A \in T(\mathrm{lub}(X))$

$\Leftrightarrow A$ は G の $\mathrm{lub}(X)$ による計算結果

$\Leftrightarrow A$ は $I \in X$ による計算結果（I は X の上界）

$\Leftrightarrow A$ は $\mathrm{lub}(T(X))$ の元 ■

［定理8］ $T \uparrow \omega$ は T の最小不動点で，G の最小モデルとなる．

（証明） 補題7よりただちに成立する． ■

導出原理による計算を論じるには，$\langle t \rangle$ の形の変数の扱い方，つまり $\langle t \rangle$ をヘッドとする規則の前提部（定義式）を評価し，すべての解を収集する全解収集動作を考察せねばならない．単純な SLD 導出の拡張ではうまくいかない[20),21)]．集約関数に関係して，全解収集の形式的な意味を考察し，SLD 導出に集約項の計算規則を付け加えた SLDAT 導出が提案されている．明らかに，有限で停止しない導出木を持てば計算不可能なため，この導出原理は完全ではない．また，停止性がリテラルの選択規則に依存することが知られ，この意味でも別のアプローチが必要である．

6. む す び

本稿では，演繹データベースの非正規関係による表現を提案した．与えられた事実集合をアプリオリに非正規化し，元の述語を仮想化して，規則中で用いられている述語も同時に書き換えてしまう．この結果，非正規情報を保持したまま計算対象としても，最小モデルの対応があるので宣言的な意味は変わらないことを示した．また，不動点オペレータによって非正規関係の最小モデルがもとまることを述べた．

ここで得た結果は，演繹データベースとその非正規形との理論的な関連性を明らかにしたことにあり，集合化レベルを深くしてもこれらの関連はそのまま保たれる．さらに，否定や集約関数への拡張が考えられる．

本稿の結果は，明示的な事実の外部記憶への格納方法としても利用することができる．集合の表現方法は，さまざまな観点，例えばデータ構造，アクセス頻度や信頼性等からも工夫することができる．別の直接的な利用方法としては，質問処理に新しい観点を提供することで，繰り返しや再試行が大幅に減少し探索空間の縮小を図ることができる．

謝辞 日頃ご指導頂いている小林功武教授（産能大学），有沢博助教授（横浜国立大学）に感謝します．また，有益な討論を行っている新世代技術開発機構DOO作業グループのメンバ，特に横田一正氏（ICOT）に感謝します．査読者の建設的なコメントは本稿を洗練させるのに十分有益でした．

参 考 文 献

1) Kanellakis, P. C.: Logic Programming and Parallel Complexity, *Proc. Int. Conf. Database Theory*, pp. 1-30 (1986).
2) Minker, J. (ed.): *Foundation of Deductive Databases and Logic Programming*, Morgan Kaufmann (1988).
3) Lloyd, J. W.: *Foundations of Logic Programming* (2nd), Springer-Verlag (1987).
4) Campbell, J. A.: *Implementation of Prolog*, John Wiley & Sons (1984).
5) Stering, L. and Shapiro, E.: *The Art of Prolog*, MIT Press (1986).
6) Walther, C.: Many Sorted Unification, *J. ACM*, Vol. 35, No. 1, pp. 1-17 (1988).
7) Bancilhon, F., Maier, D. et al.: Magic Sets and Other Strange Ways to Implement Logic Programs, *Proc. Principles of Database Syst.*, pp. 1-15 (1986).
8) Ullman, J. D.: Implementation of Logical Query Languages for Databases, *ACM Trans. Database Syst.*, Vol. 10, No. 3, pp. 289-321 (1985).
9) 宇田川，大須賀：多層論理のデータベース検索への応用，データベースシステム研究会，Vol. 25, No. 1 (1981).
10) Kuper, G. M.: Logic Programming with Sets, *Proc. Principles of Database Syst.*, pp. 11-20 (1987).
11) Beeri, C., Naqvi, S. et al.: Sets and Negation in a Logic Database Language, *Proc. Principles of Database Syst.*, pp. 21-37 (1987).
12) Abiteboul, S. and Grumbach, S.: COL, A Logic Based Language for Complex Objects, *Proc. Extending Database Technology*, pp. 271-293 (1987).
13) Shmueli, Ō: Decidability and Expressiveness Aspects of Logic Queries, *Proc. Principles of Databases Syst.*, pp. 237-249 (1987).
14) Scholl, M. H.: Theoretical Foundation of Algebraic Optimization Utilizing Unnormalized Relations, *Proc. Int. Conf. Database Theory*, pp. 380-396 (1986).
15) Scholl, M. H.: Nested Relations, *AFCET Workshop*, preprint (1987).
16) Scholl, M. H.: Supporting Flat Relations by a Nested Relational Kernel, *Proc. Very Large Databases*, pp. 137-146 (1987).
17) Paul, H. B., Schek, H. J. et al.: Architecture and Implementation of the Darmstadt Database Kernel System, *Proc. SIGMOD*, pp. 196-207 (1987).
18) 三浦：非正規関係データベース理論の研究動向，データベースシンポジウム，pp. 91-100 (1987).
19) Deshpande, A. and Van Gucht, D.: An Implementation for Nested Relational Databases, *Proc. Very Large Databases*, pp. 76-87 (1988).
20) Naish, L.: All Solutions Predicate in Prolog, *Proc. Symp. on Logic Programming* (1985).
21) Miura, T. and Shioya, I.: Aggregate Horn Logic and Its Semantics, データベースシンポジウム，pp. 21-32 (1988).

(平成元年1月27日受付)
(平成元年6月13日採録)

三浦 孝夫（正会員）

1974年京都大学理学部卒業．産能大学経営情報学部講師．データベース，演繹データベース等に興味を持つ．1984年から4年間データベースシステム研究会連絡委員，1986年まで幹事．現在，電子情報通信学会，ACM各会員．電子情報通信学会データ工学研究会連絡委員．