不完全情報も扱う論理プログラミング LPII における意味論について

酒 井 浩†

論理プログラミングの枠組みにおいて知的処理システムを実現する際の問題の一つに，不完全な知識の利用に関するものがある．本論文では，ある程度知識に不完全性があってもそれらを利用できるような体系の確立と実現を図るために，LPII (Logic Programming with Incomplete Information, 不完全情報も扱う論理プログラミング) を提案する．LPII は論理プログラミングの枠組みを継承し，また選言論理プログラミングや不完全情報データベースの枠組みを統合化した体系になっている．我々は，LPII におけるプログラム節を“確定節における各原子式をそれぞれ正リテラルの選言に置き換えた論理式”で定義し，このプログラム節の集合を LPII におけるプログラムとする．定義されたプログラムは構文的に拡張され，記述能力は増加している．そして，構文に自由度が増えたために，正リテラルの選言に対する解釈に依存した異なる意味論の展開ができる．各意味論に基づく処理系も実現しているが，本論文ではプログラムの意味論を中心にして LPII の枠組みに言及する．

On Semantics for the LPII (Logic Programming with Incomplete Information)

Hiroshi Sakai†

In order to develop intellectual knowledge base systems, it is necessary for us to use knowledge which may have incompleteness in the logic programming. For this issue, we propose a framework named LPII (Logic Programming with Incomplete Information), which is an advancement from the logic programming. In the LPII, we integrate several concepts from disjunctive logic programming and relational databases with incomplete information. A program clause is made by substituting disjunctions of atomic formulas for atomic formulas in every definite clause. The defined program clauses are more expressive than definite clauses. For the defined program, we need theorem provers depending on the interpretation to disjunctions of atomic formulas. In this paper, we focus on the semantics for LPII. Finally, we refer to the theorem provers briefly.

1. はじめに

論理プログラミングとは，定理の自動証明[1]における一証明手続きである導出法[2]を論理式の集合に適用し，論理式をプログラミング言語として扱う枠組みのことである．論理プログラミングでは，従来のプログラミング言語にない知識情報の処理が可能であり，理論面からも応用面からも多くの成果が得られている．中でも特に，確定節と呼ばれる論理式の集合を扱う場合にはホーン論理によるプログラムの意味論が確立しており，SLD 導出[3]と呼ばれる効率的な導出法の存在が示されている．これらの背景の下で prolog 言語が実現されている．

† 九州工業大学工学部
Faculty of Engineering, Kyushu Institute of Technology

次に，以下の論理式A1

$$\forall X[\text{奇数}(X)\vee\text{偶数}(X)\leftarrow\text{自然数}(X)]$$

を考える．A1はよく知られた知識“自然数なら奇数か偶数である”を表現した論理式であるが，この式は確定節になっていない．理由は，A1の頭部：奇数(X) $\vee$偶数(X) が選言になっているためである．原子式の選言で記述される知識を以後，選言情報と呼ぶ．A1のように選言情報を頭部に持つ一般の節集合に対して，ホーン論理の意味論を適用すると多くの問題点が生じる．一般にプログラムの最小モデルは存在しない．また，導出に選言的な意味での枝分かれが生じるために，SLD 導出はゴールに対する証明手続きにならない．ロビンソンの導出[2]にまで戻れば選言情報も扱えるが，これは SLD 導出のように効率的な証明手続きではない．このような問題が生じるために，選言情報の扱いは論理プログラミングにおける重要な研究

課題になっている．選言情報を扱うための論理プログラミングの枠組みは，一般に選言論理プログラミング (Disjunctive Logic Programming) と呼ばれている．選言論理プログラムは，以下に示す第1階述語論理式の有限集合，つまり節の有限集合である．

$\forall[(p_1 \vee \cdots \vee p_m) \leftarrow (q_1 \wedge \cdots \wedge q_k)]$

ただし，$m \geqq 1$，$k \geqq 0$，節中の自由変数は先頭の全称記号で束縛されている．

$k \geqq 1$ の場合の節をルールといい，$k=0$ の場合の節をファクトという．選言論理プログラムでは，最近になり意味論と処理系が提案されている．意味論については，Minker らのグループによって確立された．Minker らはロビンソンの導出まで戻るのではなく，論理プログラムの拡張版として，選言論理プログラムの意味論を展開している[4),5)]．また，実際の処理系は Loveland らのグループによって実現され Near-Horn prolog という名ですでに知られている[6),7)]．

本論文では，論理プログラムや選言論理プログラムの理論に基づいて，不完全情報も扱う論理プログラミング LPII (*Logic Programming with Incomplete Information*) の枠組みを提案する．全体の構成を簡単に示しておく．2，3 章において問題点を示し LPII の必要性と定義を示す．4章では意味論を展開し，5章では処理系について簡単に言及する．なお，第1階述語論理と確定節を扱う論理プログラミングの詳細については参考文献 8)，選言論理プログラミングの詳細については参考文献 5) に従うものとする．

2. 選言情報と不完全情報について

本章では，選言情報の解釈に依存した問題点を示す．

例1 以下の選言論理プログラム $P1$ を考える．

A1：$\forall X$[奇数$(X)\vee$偶数$(X)\leftarrow$自然数(X)]

A2：$\forall X$[正の数$(X)\vee$負の数$(X)\leftarrow$自然数(X)]

A3：自然数(5)

A1は“自然数なら奇数か偶数である”という正しい命題，A2は“自然数なら正の数か負の数である”という正しい命題を意味する．このときA1では，選言情報，奇数$(X)\vee$偶数(X) の一方が欠けても誤った命題となる．それに対してA2では，頭部：正の数$(X)\vee$負の数(X)を正の数(X) だけに置き換えても正しい命題のままである．つまり，A2には冗長性が含まれている．また，A2は本来以下の A2′

$\forall X$[正の数$(X)\leftarrow$自然数(X)]

であるべきところが，情報の不完全性のために頭部が選言になっていると見ることもできる．我々は，このように不完全な情報を記述するために選言を用いることがある．選言情報にはA1のように不完全性を持たないものと，A2のように不完全性を持ったものがあることをここで注意しておく．このとき，Minker の結果を $P1$ に適用すると，正の数(5)$\vee$負の数(5)が $P1$ の論理的帰結になり，正の数(5)は $P1$ の論理的帰結とは見なされない．選言論理プログラミングにおいては，冗長性のないプログラムを対象にすることが暗に仮定されており，上記の A2 に見られる不完全な情報に関する問題点は扱われていない．

我々は不完全な情報を記述する際に選言を用いることもあるので，選言を不完全な情報と解釈する場合の意味論や処理系の作成は重要な問題になると考えられる．次の例を考える．

例2 *append* のルールを作る際に，

$append([A|B], C, [A|D])$ または

$append(B, [A|C], [A|D])$

のいずれか一方が頭部であるという知識がある場合に，

$$\forall[append([A|B], C, [A|D]) \vee append(B, [A|C], [A|D]) \leftarrow append(B, C, D)]$$

なるルールを書いたとする．このルールは構文上は選言論理プログラムのルール形をしているが，この場合には例えば質問 $append([1], [2], X)$ を入力し，

$$\forall[append([A|B], C, [A|D]) \leftarrow append(B, C, D)]$$

を用いれば，$append([1], [2], [1, 2])$ はプログラムの論理的帰結になるという応答を得ることが利用者には望ましいと考えられる．逆にいえば $append([1], [2], [1, 2])$を論理的帰結にするように頭部を決定する手続きが望まれる．

選言情報と関係した不完全情報の枠組みは，Lipski によって導入されている．Lipski は選言 $p_1 \vee p_2 \vee \cdots \vee p_m$ に対して“個々の原子式の真偽値は不明だが，$\{p_1, p_2, \cdots, p_m\}$ の中には正しい一つの原子式が含まれている”という解釈を与えた．さらに，選言 $p_1 \vee p_2 \vee \cdots \vee p_m$ を選言的な内容を持つ一つの原子式とみなす解釈も与えた．これらの解釈によって関係データベースにおける空値問題を解決し，不完全情報に基づく処理の提案を行っている[9),10)]．

我々は，冗長な選言を扱う選言論理プログラミング

の問題と Lipski が提案している不完全情報の枠組みを統合化し，LPII を以下に提案する．

3. LPII の定義

LPII では，第1階述語論理の構文規則に従って構成される以下の論理式の集合を考え，この集合に対するさまざまな解釈とそれらに基づく処理系の作成を行う．

$$\forall[(p_1\vee\cdots\vee p_n)\leftarrow(q_{11}\vee\cdots\vee q_{1m_1})\wedge\cdots\wedge(q_{k1}\vee\cdots\vee q_{km_k})]$$

ただし，p_i，q_{ij} 等は原子式，

$n\geqq 1, k\geqq 0, m_1\geqq 0, \cdots, m_k\geqq 0$，

各原子式に含まれる自由変数は先頭の全称記号で束縛されている．

以後，先頭に全称記号がある場合には，すべての自由変数が先頭の全称記号で束縛されているものとする．定義から明らかなように，LPII で扱うプログラム節は確定節における原子式を正リテラルの選言に置き換えた形をしている．$k=0$ の場合をファクト，$k\neq 0$ の場合をルールといい，ファクトとルールの有限集合を LPII におけるプログラムという．本論文では，プログラム節における正リテラルの選言に対して以下の3通りの解釈を与え，それぞれの解釈に基づいたプログラムの意味論を展開する．

解釈1：正リテラルの選言を通常の選言と見る場合．これを Minker の解釈と呼ぶ．

解釈2：正リテラルの選言を，選言を構成する原子式の中に正しい原子式が一つだけ含まれていると見る場合，これを排他的な解釈と呼ぶ．

解釈3：正リテラルの選言を選言的内容を持つ一つの原子式と見る場合．これを巨視的な解釈と呼ぶ．

排他的な解釈と巨視的な解釈は Lipski の提案を論理プログラムに応用したものである．例1と例2で示した冗長性の問題は，正リテラルの選言に対して排他的な解釈を行った場合に対応する．また，巨視的な解釈では正リテラルの選言を選言ではなく一つの原子式と見なす．通常，基礎選言式の真偽値は各基礎原子式の真偽値に基づいて計算される，いわゆるボトムアップ的なものだが，巨視的な解釈は基礎選言式に対するトップダウン的な解釈になっている．つまり，この解釈は個々の真偽値は不明でも巨視的に見て全体としては正しいと主張できるような不完全性を持った情報に対する解釈になっている．また巨視的な解釈は，真である命題論理式 $p\vee q$ に対して閉世界仮説を適用すると矛盾することとも関係している[11]．$p\vee q$ から p も q も証明できないので p と q の真偽値をともに偽とすると，$p\vee q$ も偽となり矛盾する．この場合 $p\vee q$ が真であることは，p と q の真偽値から決定しているのではなく，巨視的な解釈によって真と割り当てられていると考えることができる．以下順次，意味論を展開する．

4. LPII における解釈と意味論

本章では3章で示した三つの解釈に基づいたプログラムの意味論を考える．

4.1 Minker の解釈に基づくプログラムの意味論

この場合は，正リテラルの選言を通常の選言と見る．つまり，プログラム中のルール，ファクトに冗長性がないと考える場合である．このとき，プログラム中の各ルールは以下の形の式の連言に等価変形できる．

$$\forall[(p_1\vee p_2\vee\cdots\vee p_m)\leftarrow(q_1\wedge\cdots\wedge q_k)]$$

つまり，各ルールは節の連言に等価変形できる．プログラムを解釈する際には，ルールとファクトすべての連言を考えるので，変形前の各ルールは等価変形された論理式の集合に置き直せる．この置き換えを行ったプログラムは構文から明らかに選言論理プログラムである．つまり，Minker の解釈を考える場合には，LPII の枠組みは選言論理プログラミングの枠組みに帰着される．

Minker は確定節に対するエルブランの理論を基礎にして拡張エルブランベースを与え，非確定節に対応できるエルブランの解釈を定義した．そして，解釈のための領域はエルブラン領域だけで十分であることを示している．また非確定節集合には一般に最小モデルが存在しないために，極小モデルやモデル状態 (model state) と呼ばれるモデルを導入することによって選言論理プログラムの意味論を展開している[5]．モデル状態とは，極小モデルのある要素を部分式に持つ正リテラルの選言の集合である．例えば命題論理式 $r\vee s$ に対して，極小モデルは $\{r\}$ と $\{s\}$ であり，モデル状態は $\{r, r\vee s\}$ と $\{s, r\vee s\}$ である．以後，Minker の解釈で考えた場合に，閉論理式 Φ がプログラム P の論理的帰結であることを，$P\models_{MIN}\Phi$ とかくことにする．

4.2 排他的な解釈に基づくプログラムの意味論

排他的な解釈では，正リテラルの選言を構成する原

子式中に正しい原子式が一つだけ含まれていると解釈する．また，以後重要な概念となるプログラムの確定例を定義する．

定義1 プログラムの確定例を以下のように定義する．

（1）正リテラルの選言 $\Phi: p_1 \vee \cdots \vee p_n$ において，論理式 $p_i(1 \leqq i \leqq n)$ を Φ の確定例という．

（2）プログラム節 C: $\forall[A \leftarrow B_1 \wedge \cdots \wedge B_n]$（$A$ と各 $B_i(0 \leqq i \leqq n)$ は正リテラルの選言）において a を A，b_i を $B_i(0 \leqq i \leqq n)$ の確定例とするとき，確定節 $\forall[a \leftarrow b_1 \wedge \cdots \wedge b_n]$ を C の確定例という．

（3）プログラム P において，各プログラム節をそれぞれの確定例で置き換えたものを P の確定例という．

定義より明らかなように，プログラムPの確定例は確定節の集合であり，通常の論理プログラムである．以後，Pの確定例全体の集合を $DEX(P)$ と書く．例えば，例1において以下の確定節

$\forall X$[正の数(X)←自然数(X)]

は A2の確定例であり，$DEX(P1)$ は四つの要素から成る．我々はこの確定例の概念によって，冗長性の問題を考える．つまり，プログラムPに冗長性があるとわかっている場合には，Pの確定例の集合 $DEX(P)$ 中に正しい論理プログラムが含まれていると考える．この場合に我々が欲しいものは，Pの論理的帰結ではなく $DEX(P)$ 中に存在する正しい論理プログラムに基づく論理的帰結である．しかし，どれが正しい確定例であるかはわからない．

プログラムの各確定例は一論理プログラムと見なせるから，エルブラン解釈を考えることができる．そこで，各確定例での解釈を次のように定義する．

定義2 Pをプログラムとする．このとき，Pのエルブラン領域によって作られるエルブランベースの部分集合をPの確定例における解釈という．

ここで，プログラム P の確定例 Q での解釈は Q のエルブランベースの部分集合ではなく，P のエルブランベースの部分集合である点に注意しておく．したがって，プログラムとその各確定例の解釈には同じ領域を用いる．例えば，$P=\{p(a) \vee q(b)\}$ では二つの確定例があり，$\{p(a)\}$ と $\{q(b)\}$ である．各解釈は共通するエルブランベース $\{p(a), p(b), q(a), q(b)\}$ の部分集合である．プログラム P とその確定例 Q に対する解釈の領域が異なると，後の補題1や定理1が成立しなくなる．プログラム P の確定例 Q におけるすべてのモデルで真となる閉論理式を Q の論理的帰結とし，新たに以下の定義を与える．

定義3 P をプログラム，Φ を閉論理式とする．

（1）Φ が P の任意の確定例における論理的帰結である場合に，Φ を P の BOX 帰結と呼び $P \models_{BOX} \Phi$ と書く．

（2）Φ が P のある確定例における論理的帰結である場合に，Φ を P の DMD (<u>D</u>IA<u>M</u>ON<u>D</u>) 帰結と呼び $P \models_{DMD} \Phi$ と書く．

二つの帰結は，$DEX(P)$ に含まれる正しい論理プログラムにおいても確実に主張できる言明と，主張できる可能性を持つ言明を規定している．BOX と DMD は様相論理における確実な概念を扱うための BOX 記号，可能性を扱うための <u>D</u>IA<u>M</u>ON<u>D</u> 記号から名付けた．

Φを基礎原子式に限定すれば，以下の命題を容易に得る．

命題1 P をプログラムとし，Φ を基礎原子式とする．また，$M(Q)$ で確定節プログラム Q の最小エルブランモデルを表すものとする．

（1）$P \models_{BOX} \Phi$ と Φ が $M(Q)$ $(Q \in DEX(P))$ の共通部分に含まれることは同値である．

（2）$P \models_{DMD} \Phi$ と Φ が $M(Q)$ $(Q \in DEX(P))$ の和集合に含まれることは同値である．

［証明］ $DEX(P)$ の各要素は通常の論理プログラムであるから最小モデルが存在する．このとき，二つの命題は自明である． ■

次に BOX 帰結，DMD 帰結と Minker の論理的帰結の関係を考える．まず，以下の具体例を考える．

例3 例1から A1を取り除き A4を追加した以下のプログラム $P3$を考える．

A2：$\forall X$[正の数(X)∨負の数(X)←自然数(X)]

A3：自然数(5)

A4：自然数(6)

この場合，A2には以下の二つの確定例が存在し，

A2′：$\forall X$[正の数(X)←自然数(X)]

A2″：$\forall X$[負の数(X)←自然数(X)]

以下の結果を得る．

（1）$P3 \models_{DMD}$ 正の数(5)，$P3 \models_{DMD}$ 正の数(6)，
$P3 \models_{DMD}$ 負の数(5)，$P3 \models_{DMD}$ 負の数(6)．

（2）$P3 \models_{BOX}$ 正の数(5)∨負の数(5)，
$P3 \models_{BOX}$ 正の数(5)∨負の数(6)，
$P3 \models_{BOX}$ 正の数(6)∨負の数(5)，

$P3 \models_{BOX}$ 正の数(6)∨負の数(6).

(3) $P3 \models_{MIN}$ 正の数(5)∨負の数(5),

$P3 \models_{MIN}$ 正の数(6)∨負の数(6).

例3において可能性を扱う DMD 系では，正の数(5)や正の数(6)を DMD 帰結にするが，不要な負の数(5)や負の数(6)も DMD 帰結とする．Minker の体系では，冗長性を持った正の数(5)∨負の数(5)が論理的帰結になる．この例からプログラム P と閉論理式 Φ において，$P \models_{MIN} \Phi$ なら $P \models_{BOX} \Phi$ が成立し，$P \models_{BOX} \Phi$ なら $P \models_{DMD} \Phi$ が成立することが推測できる．実際に補題1から定理1が成立する．

補題1 P をプログラムとし，$Q \in DEX(P)$ を任意の確定例とする．Q の最小エルブランモデル $M(Q)$ は P のモデルになる．

[証明] 定義2より，P と Q は同じ解釈の領域を持つ．次に，Q のモデルを考える．Q は確定節の集合であるから最小エルブランモデル $M(Q)$ が存在する．$M(Q)$ は命題1より各エルブランモデルの共通部分である．ここで，$[A \leftarrow B_1 \wedge \cdots \wedge B_n]\theta$ を P のプログラム節の基礎例 C とし，$[a \leftarrow b_1 \wedge \cdots \wedge b_n]\theta$ を Q に含まれる C の確定例とする．我々は，$M(Q)$ が C の本体に真を割り当て頭部 A に偽を割り当てるとすると矛盾することを示す．仮定より，$A\theta$ に含まれるいかなる基礎原子式も $M(Q)$ には含まれないから，$a\theta$ も $M(Q)$ には含まれない．しかし，エルブランベース全体は常に Q のモデルであるから，Q のモデルで $a\theta$ と $[b_1 \wedge \cdots \wedge b_n]\theta$ に真を割り当てる解釈が存在する．したがって，$a\theta$ と $[b_1 \wedge \cdots \wedge b_n]\theta$ に偽を割り当てる解釈で Q のモデルとなるものが存在する．$M(Q)$ はモデルの共通部分なので，$b_1\theta, \cdots, b_n\theta$ のすべてが $M(Q)$ に含まれることはない．この $M(Q)$ は C の本体に真を割り当てない．■

この補題において，確定例 Q のエルブラン領域に基づく解釈を考えると P と Q の解釈の領域が異なるので，Q の最小エルブランモデルでも P のモデルとならない．また，命題論理式 $C: a \leftarrow b_1 \vee b_2$ において，エルブラン解釈 $\{b_2\}$ は確定例 $a \leftarrow b_1$ のモデルであるが C のモデルではない．つまり，プログラム節のモデルでなく，かつ確定例のモデルとなるエルブラン解釈が存在することもある．しかし，最小モデルを考えるとこの問題は生じない．

定理1 P をプログラムとし，Φ を閉論理式とする．

(1) $P \models_{MIN} \Phi$ ならば $P \models_{BOX} \Phi$.

(2) $P \models_{BOX} \Phi$ ならば $P \models_{DMD} \Phi$.

[証明] (1)補題1より，任意の $Q \in DEX(P)$ の最小モデルは Minker の解釈による P のモデルである．これより，Minker の解釈で P のモデルとなるものは，P の各確定例での最小エルブランモデルを包含している．したがって $P \models_{MIN} \Phi$ のとき，Φ は P の任意の確定例 Q における最小エルブランモデルで真である．Q は確定節の集合であるから，Φ は Q の論理的帰結である．したがって，$P \models_{MIN} \Phi$ ならば $P \models_{BOX} \Phi$ である．(2)は定義より自明．■

4.3 巨視的な解釈に基づくプログラムの意味論

ここでは，正リテラルの選言を選言的な内容を持つ一つの原子式と解釈する．まず，命題論理に基づく Lipski の解釈を第1階述語論理における解釈に拡張する．LPII のプログラム節は，確定節における原子式を正リテラルの選言に置き換えたものであったので，正リテラルの選言に対する真偽値の割り当てができれば，そのままホーン論理を応用できる．通常，第1階述語論理においては領域を決め，定数，関数，変数の割り当てを考え，論理式を領域上の言明に割り当てる．その後，各基礎原子式に真偽値を割り当てることで，論理式の真偽値を決定する．しかし，我々は基礎原子式ではなく，基礎化された正リテラルの選言に真偽値を割り当てることを考える．というより，情報の不完全性のために基礎化された正リテラルの選言に対してしか真偽値を割り当てられないのである．以下に二つの定義を与える．

定義4 正リテラルの選言 $\Phi = p_1 \vee p_2 \vee \cdots \vee p_m$ に含まれる原子式で互いに異なる式の選言を $dist(\Phi)$ とする．

例えば，

$$dist(p(x) \vee p(a) \vee q(b) \vee p(a)) = p(x) \vee p(a) \vee q(b)$$

である．代入 $\theta = \{x = a\}$ を考えると，

$$dist((p(x) \vee p(a) \vee q(b) \vee p(a))\theta) = p(a) \vee q(b)$$

である．

定義5 巨視的な解釈の下でのプログラム P に対する解釈とは，領域と，定数，関数，変数の割り当てを考え，論理式を領域上の言明に割り当てたもので，基礎化された正リテラルの選言に対して以下の真偽値の割当条件を満たすものである．

(1) 基礎化された正リテラルの選言 Φ が $dist(p_1 \vee p_2 \vee \cdots \vee p_m)$ を部分式に持ち，かつ $p_1 \vee p_2 \vee \cdots \vee p_m$ が真であれば，Φ は真．

(2) それ以外の Φ は偽．

LPII のプログラムに対する論理記号，全称記号，モデル等に関する定義は従来の通りである．この定義に従えば，$P=\{r\vee s\}$ で $r\vee s$ が命題論理式の場合には，以下の I_1 と I_2 は P の解釈でありモデルでもある．

I_1: r は真，s は偽，$r\vee s$ は真．

I_2: r は偽，s は偽，$r\vee s$ は真．

I_2 をモデルとする点に注意する必要があり，この点が従来とは異なる意味論の原因になる．

定義 6 P をプログラム，Φ を閉論理式とする．Φ が P の各モデルで真と割り当てられるとき，Φ を P の IBOX 帰結と呼び，$P\models_{IBOX}\Phi$ と書く．

命題 2 プログラム P がモデルを持てば，エルブラン領域を領域とする P のモデルが存在する．

［証明］ 領域 D をもつ解釈 I が P のモデルであるとき，P における定数記号は D の要素に割り当てられる．この割り当てを AS としておく．さらに基礎化された正リテラルの選言は D 上の言明に直され真偽値を割り当てられる．このとき，真を割り当てられる D 上の基礎化された正リテラルの選言に対して，AS の逆対応により D の要素を P の定数記号に置き換えた正リテラルに真を割り当ててできたモデルは領域がエルブラン領域である． ■

命題 2 より，以後領域はエルブラン領域を考える．すると，基礎化された正リテラルの選言に対する真偽値の割り当ては，Minker が定義した拡張エルブランベースをそのまま利用できる．したがってプログラムの解釈として，以後拡張エルブランベースの部分集合であり定義 4 における割当条件を満たすものを考える．例えば，論理式 $p(a)\vee q(a)\vee r(a)$ に対して，I_1, I_2 と I_3 はモデルであり I_4 は解釈でない．

$$I_1=\{p(a), p(a)\vee q(a), p(a)\vee r(a), p(a)\vee q(a)\vee r(a)\}.$$

$$I_2=\{p(a)\vee q(a), p(a)\vee q(a)\vee r(a)\}.$$

$$I_3=\{p(a)\vee q(a)\vee r(a)\}.$$

$$I_4=\{p(a)\}.$$

Minker の解釈では基礎原子式の真偽値を基に選言の真偽値を決定するために，上記の I_2 や I_3 はモデルではない．この解釈の違いが Minker の体系と巨視的な解釈に基づく体系の違いを生み出している．二つの体系の違いについて，以下の定理を得る．

定理 2 P をプログラム，Φ を閉論理式とする．

$P\models_{IBOX}\Phi$ ならば $P\models_{MIN}\Phi$.

したがって，

$$\{\Phi|\Phi \text{ は } P \text{ の IBOX 帰結}\}$$
$$\subseteq\{\Phi|\Phi \text{ は } P \text{ の論理的帰結}\}$$
$$\subseteq\{\Phi|\Phi \text{ は } P \text{ の BOX 帰結}\}$$
$$\subseteq\{\Phi|\Phi \text{ は } P \text{ の DMD 帰結}\}$$

［証明］ Minker が定義した P のモデルは，拡張エルブランベースの部分集合であり，P の各論理式に真を割り当てる．このモデルは定義 4 の真偽値の割当条件を満たすから，Minker の定義によるモデルは IBOX 系のモデルになる．定理 1 と組み合わせると結論を得る． ■

例 4 以下のプログラム $P4$ を考える．

B1: $\forall X[r(X)\leftarrow p(X)]$

B2: $\forall X[r(X)\leftarrow q(X)]$

B3: $p(a)\vee q(b)$

このとき，$P4\models_{MIN}r(a)\vee r(b)$ が成立する．IBOX 系では $p(a)$ が偽，$q(b)$ が偽，$p(a)\vee q(b)$ が真と割り当てる解釈はモデルとなるので，$r(a)\vee r(b)$ は $P4$ の IBOX 帰結ではない．

命題 3 P をプログラムとする．P の巨視的な解釈に基づくモデルの共通部分は P のモデルである．

［証明］ 基礎化された正リテラルの選言 $\Phi=p_1\vee p_2\vee\cdots\vee p_m$ に真を割り当てる解釈は一般に複数個あるが，各解釈は必ず Φ 自身を含む．共通部分を取っても Φ はその要素になり，共通部分はモデルになる．この性質を基にすれば，参考文献 8) と同様モデル積性を示すことができる． ■

したがって，P を巨視的な解釈で考えると最小モデルが存在する．我々は P の最小モデルを $M_{IBOX}(P)$ と書く．次に，$M_{IBOX}(P)$ の要素に対する証明手続きに言及する．

定義 7 正リテラルの選言 Φ と Ψ に対して，$dist(\Psi\theta)\subseteq dist(\Phi\theta)$ を満たす変数への代入 θ が存在するとき，（巨視的な解釈のもとで）Ψ は Φ に単一化可能であるという．また，θ を単一化子という．

確定節による論理プログラムでは，単一化子は変種を除いて一意に決定されるが，定義 7 では変種を除いても単一化子が複数存在する場合があることに注意する必要がある．

定義 8 $\forall[\leftarrow A_1\wedge\cdots\wedge A_n]$（各 $A_i(1\leqq i\leqq n)$ は正リテラルの選言）をゴールとよぶ．ゴール中の A_j に代入 θ で単一化可能な頭部 B を持つルール $\forall[B\leftarrow B_1\wedge\cdots\wedge B_m]$ がある場合に，新たにゴール $\forall[\leftarrow(A_1\wedge\cdots\wedge A_{j-1}\wedge B_1\wedge\cdots\wedge B_m\wedge A_{j+1}\wedge\cdots\wedge A_n)\theta]$ を導くことを（巨視的な解釈のもとでの）導出という．

反駁，成功集合等の定義は通常のとおりである．この導出では，単一化子の誤った選択によるバックトラックも起こりうる．ただし，通常の場合と同様に，次の定理が成立する．

定理 3 Pをプログラムとする．(巨視的な解釈のもとでの) 導出は，P の最小モデル $M_{IBOX}(P)$ に対して健全かつ完全である．

［証明］ 参考文献12), 13)による． ■

5. 処理系の実現について

4章までにおいて，正リテラルの選言に対する解釈によって四つの論理的帰結，つまり Minker の論理的帰結，BOX 帰結，DMD 帰結，IBOX 帰結を議論した．1章で示したように，Minker の意味論による処理系は実現されている．我々は，残る三つの処理系の実現について簡単に示す．各処理系は，SPARC station 10 上にK-prologを用いて実現されている[14),15)]．LPII のプログラムに基づく処理の実行は，実現されたメタ述語 *box*, *dmd*, *ibox* の引数にゴールを与えることで行われる．

5.1 DMD 帰結のための処理系

DMD 系では，プログラム P の確定例の集合 $DEX(P)$ を扱う．そこでまず，プログラム P を入力とし，

hypo (仮説番号，確定例である確定節)

ic (仮説番号のリスト)

を出力するプログラムを作成した．述語 *hypo* は，第2引数が仮説であることを意味し，第1引数は仮説番号である．述語 *ic* は仮説番号のリスト中の一仮説のみ成立することを意味する．これらを基にして，メタ述語 *dmd* を実現した．質問 ?-*dmd* (ゴール) (ゴールは原子式の連言) と入力すると，各原子式を左から解いていく．まず仮説を用いずにゴールの SLD 導出を行い，失敗した時点で仮説を用いる．もし，ゴールに対する反駁があれば，その反駁で用いた仮説番号のリストが中間変数 *LIST* に代入され，*LIST* が述語 *ic* の条件を満たしていれば，質問に対する一つの解を得たことになる．

5.2 BOX 帰結のための処理系

BOX 系のメタ述語 *box* は，メタ述語 *dmd* を用いて実現した．質問 ?-*box* (ゴール) (ゴールは基礎原子式の連言) と入力すると，内部で，?-*dmd* (ゴール) が実行され，すべての可能な解が見出される．このとき，各解を解くために必要であった仮説番号のリストが逐次アサートされる．この仮説番号のリストを調べることによって，プログラム P のすべての確定例でゴールが解けるか否かの判定を行い，BOX 帰結になるか否かを応答する．ゴールに変数が含まれる場合には，さらに複雑な処理が必要になる[15)]．

5.3 IBOX 帰結のための処理系

IBOX 系のメタ述語 *ibox* は，*clause* 述語を拡張することによって実現した．質問 ?-*ibox* (ゴール) (ゴールは正リテラルの和積形) と入力すると，処理系は各正リテラルの選言を左から解いていく．$p_1 \vee p_2 \vee \cdots \vee p_m$ を実際にはリストを用いて $[p_1, p_2, \cdots, p_m]$ と記述する．*ibox* 述語は部分リストの選択と *permutation* 述語により $[p_1, p_2, \cdots, p_m]$ の部分リストを逐次生成し，通常の単一化を用いた SLD 導出を行うことによって IBOX 系の導出を行う．

5.4 各処理系の特徴

プログラムを P とする．DMD 系において P に対する質問 ?-$dmd(p(X))$ を行うと，処理系は $p(X)$ を充足する変数 X の値と P の確定例を応答する．それに対して BOX 系で質問 ?-$box(p(X))$ を行うと，任意の確定例において $\exists X p(X)$ が成立するか否かを判定する．このとき，代入されるXの値は確定例ごとに異なってもよい．もし，各確定例で $\exists X p(X)$ が成立し，X の値がそれぞれ $a_i (1 \leqq i \leqq m)$ であるとすれば，$p(a_1) \vee p(a_2) \vee \cdots \vee p(a_m)$ がプログラム P の BOX 帰結である．実際に我々が必要とする解は，BOX 帰結であり $p(a_1) = p(a_2) = \cdots = p(a_m)$ となる場合である．次に IBOX 系において，質問 ?-$ibox([p(X), q(X)])$ を行い反駁があり，変数 $X=a$ であるとする．このとき，$p(a) \vee q(a)$ は P の各確定例における論理的帰結になっている．BOX 系では各確定例における代入が異なってもよかったが，IBOX 系では各確定例で同じ代入になっていなければならない．IBOX 系は BOX 系より条件が強く，もし IBOX 系で証明ができれば BOX 系より強い言明を主張できる．

6. おわりに

冗長な選言を扱う選言論理プログラミングの問題と Lipski が提案している不完全情報の枠組みを統合化し，新たな枠組み LPII を提案し，各処理系を実現した．利用者は，同じ LPII のプログラムに対して，メタ述語 *box*, *dmd*, *ibox* を使用することで，異なる論理的帰結に基づく処理を実行でき，不完全情報に対する多面的な情報の利用ができる．また，*append* の例のよ

うに，プログラムの正しい確定例を見出すことが問題である枠組みは，不完全情報からの論理プログラム作成へ応用できると考えられる．LPII は，選言情報に対する異なる解釈を導入することで，不完全情報も扱う論理プログラミングの枠組みを提案している．

謝辞 本論文の内容について詳細にコメントしていただいた査読委員の方々に感謝いたします．本研究の一部は，平成5年度文部省科学研究費奨励研究（A）（課題番号 05780298）の支援による．

参 考 文 献

1) Chang, C. L. and Ree, R. C. T.: *Symbolic Logic and Mechanical Theorem Proving*, p. 331, Academic Press (1973).
2) Robinson, J. A.: A Machine-Oriented Logic Based on the Resolution Principle, *J. ACM*, Vol. 12, No. 1, pp. 23-41 (1965).
3) van Emden, M. N. and Kowalski, R. A.: The Semantics of Predicate Logic as a Programming Language, *J. ACM*, Vol. 23, No. 4, pp. 733-742 (1976).
4) Minker, J. and Rajasekar, A.: A Fixpoint Semantics for Disjunctive Logic Programs, *J. Logic Programming*, Vol. 9, No. 1, pp. 45-74 (1990).
5) Lobo, J., Minker, J. and Rajasekar, A.: *Foundation of Disjunctive Logic Programming*, p. 307, MIT press (1992).
6) Loveland, D. W.: Near-Horn Prolog, *Proc. 4th Int. Conf. on Logic Programming*, pp. 456-469 (1987).
7) Reed, D. W. and Loveland, D. W.: A Comparison of Three Prolog Extensions, *J. Logic Programming*, Vol. 12, No. 1-2, pp. 25-50 (1992).
8) Lloyd, J. W.: *Foundation of Logic Programming*, p. 122, Springer-Verlag (1987).
9) Lipski, W.: On Semantic Issues Connected with Incomplete Information Databases, *ACM Trans. Database Syst.*, Vol. 4, No. 3, pp. 262-296 (1979).
10) Lipski, W.: On Databases with Incomplete Information, *J. ACM*, Vol. 28, No. 1, pp. 41-70 (1981).
11) Reiter, R.: *On Closed World Data Bases, Logic and Databases*, Plenum Press, pp. 55-76 (1978).
12) Sakai, H.: Inference Methods and Semantics on Or-type Knowledge Bases, *LNAI*, Springer-Verlag, Vol. 383, pp. 136-155 (1989).
13) Sakai, H.: On a Framework for Logic Programming with Incomplete Information, Fundamenta Informaticae, EATCS, Vol. 19, No. 3, pp. 223-234 (1993), also in *Technical Report CS-1991-30*, Department of Computer Science, Duke University, USA.
14) Sakai, H.: A Knowledge Base System Handling Incomplete Information, *AI '93*, World Scientific Publishing, pp. 181-190 (1993).
15) Sakai H.: On Two Proof Procedures for Uncertain Knowledge, *Japan/Korea Joint Conf. on Expert Systems '94* (to appear).

（平成5年4月 5 日受付）
（平成6年1月13日採録）

酒井　浩（正会員）
昭和34年生．昭和57年九州大学理学部数学科卒業．昭和62年同大学院総合理工学研究科情報システム学専攻博士課程単位取得退学．同年，九州工業大学情報科学センター助手．平成元年，九州工業大学工学部講師，現在に至る．理学博士．平成2年9月より10カ月間米国デューク大学情報科学客員研究員．論理プログラミング，知識システムの研究に従事．日本人工知能学会，日本数学会各会員．