論　文

ソート述語を導入した論理プログラミングの完全性

兼岩　　憲[†]

The Completeness of Logic Programming with Sort Predicates

Ken KANEIWA[†]

あらまし　順序ソート論理は，複数のソートと半順序のサブソート関係を導入した一階述語論理の拡張である．このようなソート付きの論理は，ソートによって概念階層を記述できる有用性から知識表現言語に広く応用されている．実際，ソート付き論理の節形式に対する導出原理やソート付き論理プログラミングなどの効率の良い推論体系が研究されてきた．そのうち，Beierle の順序ソート論理は，ソートの順序によって概念階層的な知識を与えて，そのソートに対応する単項述語 (ソート述語と呼ぶ) を導入することで言明的 (公理的) 知識との融合を可能にしている．本研究では，Beierle の論理のさらに実用的な推論体系を目指し，ソート述語を導入したソート付き論理プログラミングを形式化する．その形式化では，ソート述語とサブソートに関する導出規則を新たに追加し，ゴール節からの線形導出に組み込む．そこで使われる言語に対しては，ソートシグネチャをソート述語によって拡張し，それに即した Σ^+ 構造上で意味論を定義する．最後に，ソート述語を含むプログラムのエルブランモデルを構築して，線形導出の健全性と完全性を証明する．

キーワード　順序ソート論理, ソート述語，論理プログラミング，知識ベースシステム

1. はじめに

論理を使った知識表現言語は，厳密な構文，意味論および推論方法により，その推論体系の安全性を保障することができる．しかしながら，知識表現として一般的に使われている述語論理は，フラットな表現しか持たない欠点が指摘されている [4]．そのため，論理の利点を保持しつつその表現力の乏しさを解消するのに概念階層 (ソート階層) を導入した様々な論理型言語 [1], [2], [11]〜[13] が提案されてきた．それらはハイブリッド言語と呼ばれ，概念階層的な知識と言明的な知識との 2 通りの知識表現を持つ．その言語を使えば，順序ソートにより概念階層的な知識が与えられ，言明的な記述 (論理式) を構成する変数，関数や述語にはソートを付加することができる．ハイブリッド言語の中には，Frisch [6] の論理のように，単項述語による論理式でソート階層を宣言する体系もある．

数理論理の研究では古くから，一階述語論理に複数のソートを導入した多ソート論理 [8] が研究されている．特に，ソート間に順序を与えたものを順序ソート論理 [15] と呼ぶ．このようなソート付きの論理体系は，概念階層を備えた知識ベースシステムに理論的な基盤を与えている．表 1 は，ソート付き論理の分類とその節形式およびホーン節による推論方法を一覧している．ソート付き論理はソート表現の拡張 (単一ソート，多ソートおよび順序ソートなど) により分類される．単一ソートしか持たない論理は，一階述語論理に対応し導出原理や論理プログラミング言語などの実用的な推論体系をもつ．また，多ソート論理と順序ソート論理にも導出原理による推論体系 [16], [18]〜[20] とソート付きの論理プログラミング言語 [7], [9] が提案されている．

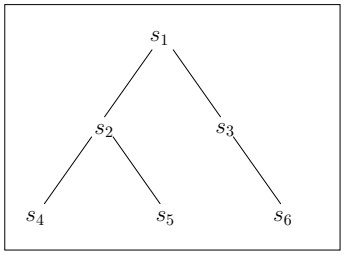
その中で知識ベースでの利用を想定して，より実用的なソート論理が Beierle ら [3] によって提案された．通常，ソート記号はソート階層を構築するとともに，変数，関数や述語のソートを明示するのに使われる．さらに各ソートは，意味論の上で単項述語と同等なことから，知識表現ではソートを単項述語として利用する要求が生じる．しかし，ソートと述語は構文上独立して存在するため，言語上は互いに全く関連性をもたず，推論にも反映されない．そこで，Beierle のソート論理では，図 1 の左側のようにソート階層を宣言し

[†] 国立情報学研究所情報学基礎研究系，東京都
Foundations of Informatics Research Division, National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430 Japan

表 1　ソート論理の研究
Table 1　Studies of sorted logics

ソート付き論理の種類	言語の拡張	節形式の推論方式	ホーン節の推論方式
一階述語論理	単一ソート	導出原理	論理プログラミング
多ソート論理	複数のソート	ソート項の単一化 + 導出原理	ソート付き論理プログラミング
順序ソート論理	順序付きソート		
Beierle のソート論理	ソート階層と知識ベースの融合	ソート項の単一化 + ソート述語の規則 + 導出原理	なし

図 1　知識ベースにおけるソート述語の利用
Fig. 1　The use of sort predicates in a knowledge base



ソート階層

$p_1(x\colon s_1) \vee p_2(y\colon s_2)$
$p_1(c\colon s_1)$
$s_1(c_2\colon s_2) \vee p_1(c_3\colon s_1)$
$s_4(c_4\colon s_1)$

知識ベース

たとき，ソートに対応した単項述語 (ソート述語と呼ぶ) を導入することで右側の知識ベース (言明的知識) との融合を可能にしている．図中で，$x\colon s$ はソート s 上の変数を意味し，$c\colon s$ はソート s に属す定数を意味する．任意の述語 p に対して $p(x\colon s)$ は「ソート s に属す x は p である」を示す．さらに，ソート述語 s により $s(t)$ は「項 t は s である」を意味する．こうした言語の拡張に加え，知識ベース上の推論ではサブソート関係 $s_1 \leqq s_2$ を，包含関係 $s_1(x) \to s_2(x)$ とみなし新しい規則が定義された．C, C_1, C_2 を節，s, s_1, s_2 をソートまたはソート述語，t, t_1, t_2 をソート項，$sort(t)$ は項 t のソートを返す関数とする．文献 [3] で定義されたソート代入と単一化により，Beirele が追加した 2 つの推論規則は以下のように表される(注1)．

$$\frac{\neg s_1(t_1) \vee C_1 \quad s_2(t_2) \vee C_2}{(C_1 \vee C_2)\theta}\ (\text{サブソート})$$

(但し，$s_2 \leqq s_1$ かつ，$t_1\theta = t_2\theta$ とする)．

$$\frac{\neg s(t) \vee C}{C\theta}\ (\text{ソート述語})$$

(但し，$sort(t\theta) \leqq s$ とする)．1 つ目の規則はサブソートに関する規則であり，2 つ目は偽となるソート述語の削除に関する規則である．これらの規則は，ソート述語を利用した公理的な知識に関する推論を実現している．しかしながら，実際に知識表現言語の実装を想定しその基盤を与えるには，ホーン節の線形導出を採用して，より実用的な推論体系を設計することが求められる．

従って本研究では，Beierle のソート論理に基づいてソート述語を導入した論理プログラミングを形式化し，その健全性と完全性を示す．そのために，次の項目を実現する．

- 論理式をホーン節に限定
- 複数の推論規則を 1 つに集約
- ゴール節による線形導出
- ソート述語を含むプログラムのエルブラン Σ^+ モデルを構築

まず，論理式の形式をホーン節 $L \leftarrow L_1 \wedge \ldots \wedge L_n$ に限定して，その集合をプログラムとする．さらに，項目 2 と項目 3 は，ソート述語を導入したことによって増えた 2 つの推論規則と従来の規則を 1 つに集約して，その線形導出を定義することを示している．その際に，空節以外にも反駁の終了を示しているゴール表現 (成功ゴールと呼ぶ) を認識して，規則を減らす試みを導入する．最後の項目では，ソート階層とソート述語に対する意味論として，Σ^+ 解釈を定義する．それにより，ソート述語を考慮したエルブラン Σ^+ モデルを構築して，完全性を証明する．

2. では，ソート述語を導入した順序ソート論理の言語，シグネチャおよび意味論を定義する． 3. では，2. で定義した言語を使ってソート述語とサブソートに

(注1)：ここでは，Beirele が定義した推論規則を簡略化している．しかし，オリジナルの規則はこれらを包含しているので，推論の妥当性は保障されている．

関する線形導出の拡張を行い，論理プログラミングの体系を構築する． 4. では，線形導出の健全性と完全性を証明する．最後に， 5. で本研究の結論を述べる．

2. ソート述語を含んだ論理型言語

2.1 構　　文

最初に，ソート述語を含んだ順序ソート言語の構文を定義する．本稿では，予めホーン節に限定した論理式の定義を行う．

定義 2.1 ソート階層をもつ一階言語 $\mathcal{L}$ のアルファベットは次の記号を含む．

(1) S*:* 最大ソート $\top$ を含むソート記号 $s_1, s_2, \ldots$ の集合

(2) F_n*:* n 引数 $(n \geqq 0)$ の関数記号 $f_1, f_2, \ldots$ の集合

(3) P_n*:* n 引数 $(n \geqq 0)$ の述語記号 $p_1, p_2, \ldots$ の集合

(4) V_s*:* ソート s の変数 $x{:}\, s, y{:}\, s, z{:}\, s, \ldots$ の集合

(5) $\leftarrow, (,)$*:* 補助記号

すべてのソート $s \in S - \{\top\}$ に対して，ソート名の添え字をもつ述語 p_s を導入し，これらをソート述語と呼ぶ．ソート述語は単項述語 $(p_s \in P_1)$ であり，ソート述語の集合を $P_S = \{p_s \mid s \in S - \{\top\}\}$ で表す．尚，本稿で扱う言語 $\mathcal{L}$ は，ソート述語を含むものとする．

定義 2.2 (シグネチャ) 言語 $\mathcal{L}$ のソート述語を含んだシグネチャは，次の条件をみたす 3 つ組 $\Sigma = (S, \Omega, \leqq)$ である．

(1) $(S, \leqq)$ は，ソートの半順序集合

(2) $f \in F_n$ ならば， $f{:}\, s_1 \times \ldots \times s_n \to s \in \Omega$

(3) $p \in P_n$ ならば， $p{:}\, s_1 \times \ldots \times s_n \in \Omega$. 特に，ソート述語 $p_s \in P_S$ に対して，$p_s{:}\, \top \in \Omega$

続いて，シグネチャ Σ が与えられたとき，言語 $\mathcal{L}$ の表現: 項，原子論理式 (アトム)，ゴールおよび節を定義する．

定義 2.3 (項) ソート s の項の集合 $\mathcal{T}_s$ は，次により定義される．

(1) $x{:}\, s \in V_s$ ならば， $x{:}\, s \in \mathcal{T}_s$

(2) $t_1 \in \mathcal{T}_{s_1}, \ldots, t_n \in \mathcal{T}_{s_n}$, $f \in F_n$ かつ $f{:}\, s_1 \times \ldots \times s_n \to s \in \Omega$ ならば， $f(t_1, \ldots, t_n){:}\, s \in \mathcal{T}_s$

(3) $t \in \mathcal{T}_{s'}$ かつ $s' \leqq s$ ならば， $t \in \mathcal{T}_s$

すべてのソートに対する項の集合 $\bigcup_{s \in S} \mathcal{T}_s$ を $\mathcal{T}$ で表す．すべてのソートに対する変数の集合を $V = \bigcup_{s \in S} V_s$ で表す．任意の項に対して，そのソートを返す関数 $sort{:}\, \mathcal{T} \to S$ は，(i) $sort(x{:}\, s) = s$ および，(ii) $f \in F_n$ かつ $f{:}\, s_1 \times \ldots \times s_n \to s \in \Omega$ ならば，$sort(f(t_1, \ldots, t_n){:}\, s) = s$，により定義される．

定義 2.4 関数 $Var{:}\, \mathcal{T} \to 2^V$ を，次のように定義する．

(1) $Var(x{:}\, s) = \{x{:}\, s\}$

(2) $c \in F_0$ に対して， $Var(c{:}\, s) = \emptyset$

(3) $f \in F_n (n \geqq 1)$ に対して，$Var(f(t_1, \ldots, t_n){:}\, s) = \bigcup_{1 \leqq i \leqq n} Var(t_i)$

$\mathcal{T}_0 = \{t \in \mathcal{T} | Var(t) = \emptyset\}$ は，変数を含まないすべての項の集合を表す．変数を含まない項 $t \in \mathcal{T}_0$ のことを基礎項と呼ぶ．さらに，ソート s の基礎項の集合を $\mathcal{T}_{0,s} = \mathcal{T}_0 \cap \mathcal{T}_s$ で表す．

定義 2.5 シグネチャ Σ が与えられたとき，アトムの集合 $\mathcal{A}$，ゴールの集合 $\mathcal{G}$，節の集合 $\mathcal{C}$ は次により定義される．

(1) $t_1 \in \mathcal{T}_{s_1}, \ldots, t_n \in \mathcal{T}_{s_n}$, $p \in P_n$ かつ $p{:}\, s_1 \times \ldots \times s_n \in \Omega$ ならば， $p(t_1, \ldots, t_n) \in \mathcal{A}$

(2) $L_1, \ldots, L_n \in \mathcal{A}$ $(n \geqq 0)$ ならば， $\{L_1, \ldots L_n\} \in \mathcal{G}$

(3) $G \in \mathcal{G}$ かつ $L \in \mathcal{A}$ ならば， $L \leftarrow G \in \mathcal{C}$

$p_s \in P_S$ に対するアトム $p_s(t)$ は，特に誤解を招かない限り $s(t)$ で表すことができる．定義 2.5 の (2) において，$n = 0$ のとき，そのゴールを空ゴールと呼び，$G = \square$ で表す．また，節 $L \leftarrow G$ に出現するゴール G が空ゴールの場合，$L \leftarrow$ によって表される．

例 2.1 以下のような記号を含む言語 $\mathcal{L}$ のシグネチャ $\Sigma = (S, \Omega, \leqq^*)$ が与えられているとする *(*但し,$\leqq^*$ は $\leqq$ の反射的推移的閉包である*)*.

$$
\begin{aligned}
S &= \{\ man, student, male\_student, person, \top\ \},\\
\leqq &= \{\ (man, person), (student, person),\\
&\qquad (male\_student, man),\\
&\qquad (male\_student, student), (person, \top)\ \},\\
\Omega &= \{p_s{:}\, \top \mid s \in S\} \cup
\end{aligned}
$$

$$\{\ studying\colon person,\ john\colon \to man\ \}.$$

このとき，節の例を以下に示す．

$$male\_student(john\colon man),$$
$$studying(x\colon person) \leftarrow \{student(x\colon person)\}.$$

定義 2.6 関数 $EVar\colon \mathcal{A} \cup \mathcal{G} \cup \mathcal{C} \to 2^V$ は次により定義される．

(1) $EVar(p(t_1, \ldots, t_n)) = Var(t_1) \cup \cdots \cup Var(t_n)$

(2) $EVar(\{L_1, \ldots, L_n\}) = EVar(L_1) \cup \cdots \cup EVar(L_n)$

(3) $EVar(L \leftarrow G) = EVar(L) \cup EVar(G)$

2.2 意 味 論

定義 2.7 (Σ 構造) シグネチャΣ が与えられたとする．Σ 構造 M は，次の条件を満たす対 (U, I) である．

- U: 空でない集合
- I: 以下の条件を満たす関数
  - $s \in S$ のとき，$I(s) \subseteqq U$ *(*特に，$I(\top) = U$*)*
  - $s_i \leqq s_j$ のとき，$I(s_i) \subseteqq I(s_j)$
  - $f \in F$ かつ $f\colon s_1 \times \ldots \times s_n \to s \in \Omega$ のとき，$I(f)\colon I(s_1) \times \ldots \times I(s_n) \to I(s)$
  - $p \in P$ かつ $p\colon s_1 \times \ldots \times s_n \in \Omega$ のとき，$I(p) \subseteqq I(s_1) \times \cdots I(s_n)$

加えて，以下をみたす Σ 構造 $M = (U, I)$ を Σ^+ 構造という．

- すべての $s \in S$ に対して, $I(s) \subseteqq I(p_s)$
- $s_i \leqq s_j$ のとき，$I(p_{s_i}) \subseteqq I(p_{s_j})$

前者の条件は，ソート s の解釈からソート述語 p_s の解釈への制約を意味する．これにより，$d \in I(s)$ であるようなすべての d は，$d \in I(p_s)$ となる．各ソートに対応して導入されたソート述語は，最低限この制約を満たすことが要求される．さらに，逆方向も加えて $I(s) = I(p_s)$ と定義することも可能である．しかし，本稿ではむやみに制約を増やしてモデルのクラスを小さくすることを避けるため，$I(s) \subseteqq I(p_s)$ とする．

関数 f が与えられたとき，$Dom(f)$ は f の定義域を示す．V' を V の部分集合とする．すべての変数 $x\colon s \in V'$ に対して，$\alpha(x\colon s) \in I(s)$ であるような関数 $\alpha\colon V' \to U$ を Σ 構造 $M = (U, I)$ 上の変数割り当てと呼ぶ．特に，$Dom(\alpha) = V$ の場合，単に変数割り当てと呼び，そうでないとき部分変数割り当てと呼ぶ．変数割り当て α に対する部分変数割り当て β の合成は，$\alpha\beta = \alpha - \{(x\colon s, \alpha(x\colon s)) \mid x\colon s \in Dom(\beta)\} \cup \beta$ によって定義される．Σ 解釈 $\mathcal{I}$ は，Σ 構造 M と変数割り当て α の対 (M, α) である．そのとき，解釈 $(M, \alpha\beta)$ は，$\mathcal{I}\beta$ によって表すことができる．

定義 2.8 Σ 解釈 $\mathcal{I} = (M, \alpha)$ に対して，解釈関数 $[\![\ ]\!]_\alpha\colon \mathcal{T} \to U$ は以下により定義される．

- $[\![x\colon s]\!]_\alpha = \alpha(x\colon s)$
- $[\![c\colon s]\!]_\alpha = I(c)$
- $[\![f(t_1, \ldots, t_n)\colon s]\!]_\alpha = I(f)([\![t_1]\!]_\alpha, \ldots, [\![t_n]\!]_\alpha)$

以上の定義を使って，解釈と式の充足関係を定義する．

定義 2.9 $\mathcal{I} = (M, \alpha)$ を Σ 解釈とする．Σ 充足関係 $\models_\Sigma \subseteqq \mathcal{I} \times (\mathcal{A} \cup \mathcal{G} \cup \mathcal{C})$ は，以下のように帰納的に定義される．

- $\mathcal{I} \models_\Sigma p(t_1, \ldots, t_n)$ *iff* $([\![t_1]\!]_\alpha, \ldots, [\![t_n]\!]_\alpha) \in I(p)$
- $\mathcal{I} \models_\Sigma \{L_1, \ldots, L_n\}$ *iff* $\mathcal{I} \models_\Sigma L_1, \ \ldots, \ \mathcal{I} \models_\Sigma L_n$
- $\mathcal{I} \models_\Sigma L \leftarrow G$ *iff* $Dom(\gamma) = EVar(L \leftarrow G)$ であるような任意の変数割り当て γ に対して，$\mathcal{I}\gamma \models_\Sigma G$ ならば $\mathcal{I}\gamma \models_\Sigma L$

$\mathcal{I}$ を Σ 解釈とする．論理式の集合 Γ のすべての元 A に対して，$\mathcal{I} \models_\Sigma A$ が成り立つとき，$\mathcal{I}$ を Γ の Σ モデルといい，$\mathcal{I} \models_\Sigma \Gamma$ で表す．Γ が少なくとも 1 つの Σ モデルを持つとき，Γ は Σ 充足可能であるといい，そうでないときは，Σ 充足不可能であるという．論理式の集合 Γ の任意の Σ モデルが，論理式 A の Σ モデルでもあるとき，A は Γ の Σ 構造のクラスにおける論理的帰結といい，$\Gamma \models_\Sigma A$ と表す．特に，Γ の元が 1 つのとき，$\{B\} \models_\Sigma A$ を $B \models_\Sigma A$ と表す．さらに，Σ 解釈 $\mathcal{I} = (M, \alpha)$ の構造 M が Σ^+ 構造のとき，$\mathcal{I}$ を Σ^+ 解釈という．その充足関係を Σ^+ 充足関係といい，$\models_{\Sigma^+}$ で表す．よって，同様に Σ^+ モデル，Σ^+ 充足可能，Σ^+ 充足不可能，および Σ^+ 構造のクラスにおける論理的帰結を定義できる．

3. 論理プログラミング

定義 3.1 (プログラム) シグネチャ Σ が与えられているとする．プログラムは, 節の有限集合 $\mathcal{P}_\Sigma \subseteqq \mathcal{C}$ である．

定義 3.2 (ソート代入) ソート代入は，以下をみたす

部分関数 $\theta: V \to \mathcal{T}$ である．

- $\theta(x:s) \in \mathcal{T}_s - \{x:s\}$
- $Dom(\theta) \subseteqq V$ は有限

前者の条件により，ソート代入は変数 $x:s$ と同じソートもしくはそのサブソートの項 $t(\in \mathcal{T}_s$(注2)$)$ への変換に制限される．ソート代入は，有限集合 $\{x_1:s_1/t_1, \ldots, x_n:s_n/t_n\}$ として表すことができる．

θ をソート代入とする．すべての $x:s \in Dom(\theta)$ に対して $\theta(x:s)$ が基礎項，すなわち $Var(\theta(x:s)) = \emptyset$ ならば，θ をソート基礎代入という．変数の集合 $V' \subseteqq V$ が与えられたとする．V' への代入 θ の制限は，$\theta\restriction V' = \{\theta(x:s)/x:s \mid x:s \in V' \cap Dom(\theta)\}$ により定義される．また，$Dom(\theta) \subseteqq V'$ かつ $\theta\restriction V'$ がソート基礎代入ならば，θ を V' に対するソート基礎代入という．空集合によって与えられる恒等代入は，ϵ によって表される．

続いて，ソート代入を項，アトム，ゴールおよび節へ拡張する．

定義 3.3　$E \in \mathcal{T} \cup \mathcal{A} \cup \mathcal{G} \cup \mathcal{C}$ とすると，$E\theta$ は以下のように定義される．

- $x:s \in Dom(\theta)$ ならば，$x:s\theta = \theta(x:s)$
- $x:s \notin Dom(\theta)$ ならば，$x:s\theta = x:s$
- $f(t_1, \ldots, t_n)\theta = f(t_1\theta, \ldots, t_n\theta)$
- $p(t_1, \ldots, t_n)\theta = p(t_1\theta, \ldots, t_n\theta)$
- $\{L_1, \ldots, L_n\}\theta = \{L_1\theta, \ldots, L_n\theta\}$
- $(L \leftarrow G)\theta = L\theta \leftarrow G\theta$

代入 θ_1, θ_2 が与えられたとする．代入の合成 $\theta_1\theta_2$ は，$(x:s)\theta_1\theta_2 = ((x:s)\theta_1)\theta_2$ によって定義される．また，$\theta_1 = \theta_2\gamma$ であるような代入 γ が存在するとき，θ_2 は θ_1 より一般的であるという．$E_1, E_2 \in \mathcal{T} \cup \mathcal{A} \cup \mathcal{G} \cup \mathcal{C}$ とする．$E_1\theta = E_2\theta$ が成り立つとき，ソート代入 θ を E_1 と E_2 の単一化子という．E_1 と E_2 の単一化子 θ が，E_1 と E_2 のすべての単一化子より一般的であるとき，最汎単一化子であるという．本稿では，最汎な単一化が一意に決まるように，ソート階層の構造を交わり準束 (meet-semilattice) に限定する．即ち，任意の 2 つのソートに対して，必ず極大下界 (greatest lower bound) が存在する．また，ソート階層が準束でない場合は，文献 [4] などで述べられているように，ソート階層を修正する方法がある．

(注2)：定義 2.3 により，$\mathcal{T}_s$ はソート s もしくはそのサブソートの項の集合を示す．

補題 3.1　Σ 解釈 $\mathcal{I}$，節 $L \leftarrow G$，ソート代入 θ が与えられたとする．このとき，$\mathcal{I} \models_\Sigma L \leftarrow G$ ならば，$\mathcal{I} \models_\Sigma (L \leftarrow G)\theta$ が成り立つ．

(証明) Σ 解釈 $\mathcal{I} = (M, \alpha)$ に対して，$\mathcal{I} \models_\Sigma L \leftarrow G$ とする．定義 2.9 により，$Dom(\beta) = EVar(L \leftarrow G)$ であるような任意の部分変数割り当て β に対して，$\mathcal{I}\beta \models_\Sigma G$ ならば $\mathcal{I}\beta \models_\Sigma L$ である．一方，$Dom(\gamma) = EVar((L \leftarrow G)\theta)$ であるような任意の部分変数割り当て γ に対して，β を $\beta(x:s) = [\![\theta(x:s)]\!]_{\alpha\gamma}$ と置く．すると，$\mathcal{I}\beta \models_\Sigma L \leftarrow G$ iff $\mathcal{I}\gamma \models_\Sigma (L \leftarrow G)\theta$ が成り立つ．従って，$\mathcal{I} \models_\Sigma (L \leftarrow G)\theta$　□

θ をソート代入，C を節とすると，$C\theta$ を C の代入例 (もしくは，単に例) という．すべての C の基礎例からなる集合を $ground(C)$ で表し，$\bigcup_{c \in \Delta} ground(C)$ を $ground(\Delta)$ で表す．

$\mathcal{P}_\Sigma$ をプログラム，G をゴールとする．ゴール G からの線形導出は通常以下の規則を適用して行われる．

定義 3.4 (導出規則 1：カット)　$L' \leftarrow G' \in \mathcal{P}_\Sigma$ とする．θ が $L \in G$ と L' の単一化子ならば，$(G - \{L\})\theta \cup G'\theta$ は L と $L' \leftarrow G'$ に関する G のサブゴールであり，以下のように表される．

$$G \xrightarrow{\theta}_{R1} (G - \{L\})\theta \cup G'\theta$$

さらに，ソート述語を導入したことによって，2 つの導出規則が必要になってくる．そこで，1. で説明した Beierle の推論規則に基づいて，線形導出の形式に変形させた規則を次のように導入する．

定義 3.5 (導出規則 2：サブソート)　$s(t) \in G$ かつ $s'(t') \leftarrow G' \in \mathcal{P}_\Sigma$ とする．$s' \leqq s$，かつ θ が t と t' の単一化子ならば，$(G - \{s(t)\})\theta \cup G'\theta$ は $s(t)$ と $s'(t') \leftarrow G'$ に関する G のサブゴールであり，以下のように表される．

$$G \xrightarrow{\theta}_{R2} (G - \{s(t)\})\theta \cup G'\theta$$

定義 3.6 (導出規則 3：ソート述語)　$s(t) \in G$ かつ $sort(t\theta) \leqq s$ ならば，$(G - \{s(t)\})\theta$ は $s(t)$ に関する G のサブゴールであり，以下のように表される．

$$G \xrightarrow{\theta}_{R3} (G - \{s(t)\})\theta$$

しかしながら，このままでは合計 3 つの規則が必要で，通常の論理プログラミングのように単一の導出規

則による証明器の特徴を損なってしまう．そこで，できる限り 1 つの規則への集約を試みる．まず，導出規則 3 を削除するために，新たに成功ゴール (successful goal) という概念を導入する．

定義 3.7 (成功ゴール) $G \in \mathcal{G}$ をゴールとする．すべての $L \in G$ がソート述語によるアトム $L = s(t)$ であり，t の例 t' が存在し $sort(t') \leqq s$ であるならば，G を成功ゴールと呼ぶ．空ゴールまたは成功ゴールを $\bot$ で表す．

線形導出 3 を適用してソート述語によるアトムを削除する代わりに，導出の最後までそれらのアトムを残すようにする．例えば，$sort(t_1) \leqq s_1$ かつ $sort(t_2) \leqq s_2$ のとき，サブゴール

$$\{s_1(t_1), s_2(t_2)\}$$

が導かれたならば，この状態を線形導出の成功とみなして反駁を完了すればいいのである．

さらに，導出規則 1 と導出規則 2 を統合するために，関数

$$\langle L \rangle = \{L\} \cup \{s'(t) \mid L \equiv s(t) \ and \ s' \leqq s\}$$

を使って，導出規則 1 を次のように変形させる．

定義 3.8 (導出規則 4) $L \in G$ かつ $L' \leftarrow G' \in \mathcal{P}_\Sigma$ とする．θ が $L_0 \in \langle L \rangle$ と L' の単一化子ならば，$(G - \{L\})\theta \cup G'\theta$ は L と $L' \leftarrow G'$ に関する G のサブゴールであり，以下のように表される．

$$G \xrightarrow{\theta}_{R4} (G - \{L\})\theta \cup G'\theta.$$

続いて，導出規則 4 のみを使って制限なし線形導出を定義する．

定義 3.9 (線形導出) $\mathcal{P}_\Sigma$ をプログラムとする．有限列

$$\mathcal{P}_\Sigma\colon G_0 \xrightarrow{\theta_1}_{R4} G_1 \xrightarrow{\theta_2}_{R4} \cdots \xrightarrow{\theta_n}_{R4} G_n$$

をプログラム $\mathcal{P}_\Sigma$ に関する G_0 の制限なし線形導出という *($n \geqq 0$)*．制限なし線形導出は，$\mathcal{P}_\Sigma\colon G_0 \xrightarrow{\theta}\!\!\!\!\rightarrow G_n$ *(*但し，$\theta = \theta_1 \cdots \theta_n$*)* によって表される．また，$1 \leqq i \leqq n$ に対して，各ゴール G_{i-1} に制限された代入 $\theta_i{\uparrow}EVar(G_{i-1})$ を $\theta_i^{\uparrow}$ で表す．さらに，$\theta^{\uparrow} = \theta_1^{\uparrow} \cdots \theta_n^{\uparrow}$ とする．

制限なし線形導出に伴うすべての単一化子が最汎単一化子ならば，単に線形導出と呼ぶ．G_n が成功ゴール (即ち，$\mathcal{P}_\Sigma\colon G_0 \xrightarrow{\theta}\!\!\!\!\rightarrow \bot$) のとき，線形導出は成功したという．そのとき，初期ゴール G_0 に含まれる変数への代入の合成 $(\theta_1 \cdots \theta_n){\uparrow}EVar(G_0)$ を計算解代入と呼ぶ．

例 3.1 例 *2.1* のシグネチャ $\Sigma = (S, \Omega, \leqq^*)$ を使って線形導出の例を示す．プログラム $\mathcal{P}_\Sigma$ は，以下のような節の集合とする．

$$\mathcal{P}_\Sigma = \{\ male\_student(john\colon man),$$
$$studying(x\colon person) \leftarrow \{student(x\colon person)\}\ \}.$$

そのとき，ゴール $\{studying(john\colon man)\}$ の線形導出は次のように成功する．

$$\mathcal{P}_\Sigma : \{studying(john\colon man)\} \xrightarrow{\theta}_{R4}$$
$$\{student(john\colon man)\} \xrightarrow{\epsilon}_{R4} \bot$$

ただし，$\theta = \{x\colon person/john\colon man\}$ である．上記では，導出規則の一回目の適用がゴールと節による通常の導出であり，二回目がサブソート関係による節 *(*ソート述語を含む*)* への導出である．

4. 線形導出の完全性

線形導出の完全性(注3)を証明するために，3. で定義した線形導出に次の 2 つの条件を与える．(1) 各ゴールとプログラム内の節，およびプログラム内の節同士で，同じ変数を含まない．(2) 導出途中で出現しなくなった変数は，再び導入されない．

補題 4.1 $\mathcal{I}$ を Σ 解釈，L をアトムとする．$\mathcal{I} \models_\Sigma L \leftarrow$ ならば，$\mathcal{I} \models_\Sigma L$ である．

(証明) Σ 解釈 $\mathcal{I} = (M, \alpha)$ に対して，$\mathcal{I} \models_\Sigma L \leftarrow$ とする．定義 2.9 により，$Dom(\beta) = EVar(L \leftarrow)$ であるような任意の部分変数割り当て β に対して，$\mathcal{I}\beta \models_\Sigma L$ である．β を $\beta(x\colon s) = \alpha(x\colon s)$ と置く．すると，$\mathcal{I}\beta \models_\Sigma L$ iff $\mathcal{I} \models_\Sigma L$ となる．従って，$\mathcal{I} \models_\Sigma L$ が導かれる． □

(注3)：論文 [17] で述べられているように，文献 [14] における論理プログラミングの完全性定理は，代入に関して誤りを含んでいる．従って，本稿では Doets の形式化 [5] に基づいて完全性を証明する．

定理 4.1（線形導出の健全性） $\mathcal{P}_\Sigma$ をプログラム，G をゴールとする．$\mathcal{P}_\Sigma{:}\,G \overset{\theta}{\longrightarrow\!\!\!\!\!\rightarrow} \bot$ ならば，$\mathcal{P}_\Sigma \models_{\Sigma^+} G\theta$ である．

(証明) 線形導出の長さ n に関する帰納法で証明する．$n=1$ のとき，線形導出 $\mathcal{P}_\Sigma{:}\,G(=L) \xrightarrow{\theta}_{R4} \bot$ が存在する．定義 3.8 より，ある節 $L' \leftarrow \in \mathcal{P}_\Sigma$ に対して，$L_0 \in \langle L\rangle$ と L' の単一化子 θ が存在する．一方，$\mathcal{I}$ を $\mathcal{P}_\Sigma$ の Σ^+ モデルとすると，補題 3.1 より，$\mathcal{I} \models_{\Sigma^+} (L' \leftarrow)\theta$ である．(i)$L_0 = L$ のとき，$L_0\theta = L'\theta$ なので，$\mathcal{I} \models_{\Sigma^+} (L \leftarrow)\theta$ となる．(ii)$L_0 = s'(t)$ のとき，$L = s(t)$, $L' = s'(t')$ かつ $s' \leqq s$ である．よって，$L_0\theta = L'\theta$ なので，$\mathcal{I} \models_{\Sigma^+} (s'(t') \leftarrow)\theta$ と $I(s')\subseteqq I(s)$ により，$\mathcal{I} \models_{\Sigma^+} (s(t) \leftarrow)\theta$ が言える．故に，補題 4.1 から $\mathcal{I} \models_{\Sigma^+} L\theta$ が導かれる．
$n > 1$ のとき，線形導出

$$\mathcal{P}_\Sigma{:}\,G \xrightarrow{\theta_1}_{R4} (G - \{L\})\theta_1 \cup G'\theta_1 \xrightarrow{\theta_2}_{R4} G_2 \xrightarrow{\theta_3}_{R4} \cdots \xrightarrow{\theta_n}_{R4} \bot$$

が存在し，$L' \leftarrow G' \in \mathcal{P}_\Sigma$ かつ $L_0\theta_1$(但し，$L_0 \in \langle L\rangle) = L'\theta_1$ である．帰納法の仮定により，$\mathcal{I} \models_{\Sigma^+} ((G-\{L\})\theta_1 \cup G'\theta_1)\theta'$ である (但し，$\theta' = \theta_2 \cdots \theta_n$)．一方，$\mathcal{I} \models_{\Sigma^+} (L' \leftarrow G')\theta_1\theta'$ が成り立つから，(i)$L_0 = L$ のとき，$L\theta_1 = L'\theta_1$ より，$\mathcal{I} \models_{\Sigma^+} (L \leftarrow)\theta_1\theta'$ となる．(ii)$L_0 = s'(t)$ のとき，$L_0\theta_1 = L'\theta_1$ なので，$L = s(t)$, $L' = s'(t')$ かつ $s' \leqq s$ である．よって，$\mathcal{I} \models_{\Sigma^+} (s'(t') \leftarrow)\theta_1$ と $I(s')\subseteqq I(s)$ により，$\mathcal{I} \models_{\Sigma^+} (s(t) \leftarrow)\theta_1\theta'$ が言える．故に，補題 4.1 から $\mathcal{I} \models_{\Sigma^+} L\theta_1\theta'$ が導かれる． □

続いて，完全性の証明を与える準備として順序ソート言語に対応したエルブラン構造を定義する．

定義 4.1 エルブラン構造 $M_H = (I_H, U_H)$ は，次の条件を満たす構造である．

(1) $U_H = \mathcal{T}_0$

(2) $I_H(s) = \mathcal{T}_{0,s}$

(3) $c \in F_0$ かつ $c{:}\to s \in \Omega$ のとき，$I_H(c) = c{:}\,s$

(4) $f \in F_n$ かつ $f{:}\,s_1 \times \ldots \times s_n \to s \in \Omega$ のとき，$I_H(f)(t_1, \ldots, t_n) = f(t_1 \ldots, t_n){:}\,s$

(5) $p \in P_n$ かつ $p{:}\,s_1 \times \ldots \times s_n \in \Omega$ のとき，$I_H(p)\subseteqq I_H(s_1) \times \cdots \times I_H(s_n)$

エルブラン解釈 $\mathcal{I}_H$ は，その構造がエルブラン構造である解釈をいう．さらに，任意のエルブラン構造が Σ 構造であることは，$s \leqq s'$ に対して，$\mathcal{T}_{0,s}\subseteqq\mathcal{T}_{0,s'}$ であることから明らかである．

補題 4.2 $\mathcal{I}_H$ をエルブラン解釈，$L \leftarrow G$ を節とする．このとき，$\mathcal{I}_H \models_\Sigma L \leftarrow G$ ならば，かつそのときに限り $\mathcal{I}_H \models_\Sigma ground(L \leftarrow G)$ である．

(証明)
($\Rightarrow$) $\mathcal{I}_H \models_\Sigma L \leftarrow G$ とする．補題 3.1 により，$L \leftarrow G$ のすべての例に対して，$\mathcal{I}_H \models_\Sigma (L \leftarrow G)\theta \in ground(L \leftarrow G)$ なので明らかである．
($\Leftarrow$) $\mathcal{I}_H \models_\Sigma ground(L \leftarrow G)$ を仮定する．$Dom(\beta) = EVar(L \leftarrow G)$ であるような任意の部分変数割り当て β に対して，$\mathcal{I}_H\beta \models_\Sigma G$ ならば $\mathcal{I}_H\beta \models_\Sigma L$ であることを証明する．定義 4.1 から $I_H(s_i) = \mathcal{T}_{0,s_i}$ が言えるので，$EVar(L \leftarrow G)$ への任意のソート基礎代入 θ に対して，$(L \leftarrow G)\theta \in ground(L \leftarrow G)$ である．したがって，$x_i{:}\,s_i \in EVar(L \leftarrow G)$ に対して $\theta(x_i{:}\,s_i) = \beta(x_i{:}\,s_i)$ と置くと，仮定より $\mathcal{I}_H\beta \models_\Sigma G$ ならば $\mathcal{I}_H\beta \models_\Sigma L$ である． □

$\mathcal{P}_\Sigma$ から生成される基礎節 C の推論木を以下のように定義する．

定義 4.2 $\mathcal{P}_\Sigma$ をプログラムとする．$\mathcal{P}_\Sigma$ に関する基礎節 C の推論木は，以下をみたすものとする．

(1) ルートは C である．

(2) 各ノードは基礎節である

(3) リーフは，次のいずれかである．

- 節 $L \leftarrow G \in ground(\mathcal{P}_\Sigma)$
- $s' \leqq s$ のとき，$s(t) \leftarrow$*(*但し，$sort(t) = s'$*)* もしくは，$s(t') \leftarrow s'(t')$

(4) $L \leftarrow G_1 \cup \{L'\}$ と $L' \leftarrow G_2$ がノードならば，$L \leftarrow G_1 \cup G_2$ はその親ノードである．

定義 4.3 $\mathcal{P}_\Sigma$ をプログラムとする．標準解釈 $\mathcal{I}_\mathcal{P}$ は，以下をみたすエルブラン解釈である．

$\mathcal{I}_\mathcal{P} \models_{\Sigma^+} L$ *iff* $\mathcal{P}_\Sigma$ に関する基礎節 $L \leftarrow$ の推論木が存在する

補題 4.3 $\mathcal{P}_\Sigma$ をプログラムとすると，標準解釈 $\mathcal{I}_\mathcal{P}$ は $\mathcal{P}_\Sigma$ の Σ^+ モデルである．

(証明) まず，$\mathcal{I}_\mathcal{P}$ が Σ^+ 解釈であることを示す．$\mathcal{I}_\mathcal{P}$ はエルブラン解釈なので，Σ 解釈であるのは明らかである．$I(s)\subseteqq I(p_s)$ を示す．$t \in I(s)(= \mathcal{T}_{0,s})$ とすると，ある $s' \leqq s$ が存在して，$sort(t) = s'$ であ

る．よって $s(t) \leftarrow$ の推論木が存在する．定義 4.3 より $\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} s(t)$ なので，$t \in I(p_s)$ である．続いて，$s' \leqq s$ のとき，$I(p_{s'}) \subseteqq I(p_s)$ を示す．$t \in I(s')$ を仮定すると，$\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} s'(t)$ なので，$s'(t) \leftarrow$ の推論木が存在する．一方，$s(t) \leftarrow s'(t)$ の推論木も存在し，$s(t) \leftarrow$ の推論木が導けるので，$\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} s'(t)$ である．故に，$t \in I(p_{s'})$ である．

次に，$\mathcal{I}_{\mathcal{P}}$ が $\mathcal{P}_\Sigma$ のモデルであることを示す．$L \leftarrow G(= \{L_1, \ldots, L_n\}) \in \mathcal{P}_\Sigma$ であり，θ は $EVar(L \leftarrow G)$ に対する任意の基礎代入とする．$\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} G\theta$ のとき，$\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} L_i\theta$ であるので定義 4.2 より，$L_i\theta \leftarrow$ の推論木が存在する．また，$(L \leftarrow G)\theta \in ground(\mathcal{P}_\Sigma)$ なので，$(L \leftarrow G)\theta$ の推論木が存在する．よって，推論木の定義から $L\theta$ の推論木が導かれる．従って，$\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} L\theta$ であり，$\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} (L \leftarrow G)\theta$ となる．補題 4.2 を用いて，$\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} L \leftarrow G$.
$\mathcal{I}_{\mathcal{P}}$ は Σ^+ 解釈であるので，充足関係の定義より以下を充たさなければならない．$s' \leqq s$ のとき，

- $sort(t) = s'$ に対して，$\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} s(t)$
- $\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} s'(t)$ ならば，$\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} s'(t)$

上記の $\mathcal{I}_{\mathcal{P}}$ が Σ^+ 解釈である証明により明らかである． □

補題 4.4 $\mathcal{P}_\Sigma$ がプログラムであり，$\mathcal{P}_\Sigma$ に関する基礎節 $L \leftarrow G$ の推論木が存在するとする．そのとき，$\mathcal{P}_\Sigma{:}\, G \longrightarrow\!\!\!\!\!\rightarrow \bot$ ならば，$\mathcal{P}_\Sigma{:}\, L \longrightarrow\!\!\!\!\!\rightarrow \bot$ である．

(証明) 基礎節 $L \leftarrow G$ の推論木の高さ n に関する帰納法で証明する．

$n = 1$ のとき，$L \leftarrow G$ は次の 3 通りが考えられる．(i)$L \leftarrow G \in ground(\mathcal{P}_\Sigma)$ のとき，$\mathcal{P}_\Sigma{:}\, G \longrightarrow\!\!\!\!\!\rightarrow \bot$ ならば，$\mathcal{P}_\Sigma{:}\, L \xrightarrow{\epsilon}_{R4} G \longrightarrow\!\!\!\!\!\rightarrow \bot$ が成り立つ．(ii)$s(t) \leftarrow$ のとき，$s' \leqq s$ かつ $sort(t) = s$ が成り立つはずである．よって，$\{s(t)\}$ は成功ゴールなので，$\mathcal{P}_\Sigma{:}\, s(t) \longrightarrow\!\!\!\!\!\rightarrow \bot$ は明らか．(iii)$s(t) \leftarrow s'(t)$ のとき，$s' \leqq s$ が成り立つはずである．従って，$\mathcal{P}_\Sigma{:}\, s'(t) \longrightarrow\!\!\!\!\!\rightarrow \bot$ ならば，$\mathcal{P}_\Sigma{:}\, s(t) \xrightarrow{\epsilon}_{R4} s'(t) \longrightarrow\!\!\!\!\!\rightarrow \bot$ が言える．

$n > 1$ のとき，帰納法の仮定より，$\mathcal{P}_\Sigma{:}\, G_1 \cup \{L'\} \longrightarrow\!\!\!\!\!\rightarrow \bot$ ならば $\mathcal{P}_\Sigma{:}\, L \longrightarrow\!\!\!\!\!\rightarrow \bot$ であり，$\mathcal{P}_\Sigma{:}\, G_2 \longrightarrow\!\!\!\!\!\rightarrow \bot$ ならば，$\mathcal{P}_\Sigma{:}\, L' \longrightarrow\!\!\!\!\!\rightarrow \bot$ である (但し，$G = G_1 \cup G_2$)．よって，$\mathcal{P}_\Sigma{:}\, G_1 \cup G_2 \longrightarrow\!\!\!\!\!\rightarrow \bot$ ならば，$\mathcal{P}_\Sigma{:}\, L \xrightarrow{\epsilon}_{R4} G_1 \cup G_2 \longrightarrow\!\!\!\!\!\rightarrow \bot$ が成り立つ． □

定理 4.2 (基礎ゴールに対する線形導出の完全性) $\mathcal{P}_\Sigma$ をプログラム，G を基礎ゴールとする．$\mathcal{P}_\Sigma \models_{\Sigma^+} G$ であるならば，線形導出 $\mathcal{P}_\Sigma{:}\, G \xrightarrow{\theta}\!\!\!\!\!\rightarrow \bot$ が存在する．

(証明) $\mathcal{I} \models_{\Sigma^+} \mathcal{P}_\Sigma$ であるとする．このとき，前提より $\mathcal{I} \models_{\Sigma^+} G$ であるので，補題 4.3 より $\mathcal{I}_{\mathcal{P}} \models_{\Sigma^+} G$ が言える．ここで定義 4.3 より，すべての $L \in G$ に対して，$\mathcal{P}_\Sigma$ に関する基礎節 $L \leftarrow$ が存在する．従って，補題 4.4 を使って $\mathcal{P}_\Sigma{:}\, L \longrightarrow\!\!\!\!\!\rightarrow \bot$ が言える．故に，$\mathcal{P}_\Sigma{:}\, G \xrightarrow{\theta}\!\!\!\!\!\rightarrow \bot$ の存在が証明される． □

次の補題とその証明は，論文 [5] における補題 5.33 に基づく．

補題 4.5 $\mathcal{P}_\Sigma$ をプログラムとする．$\mathcal{P}_\Sigma$ の制限なし線形導出

$$\mathcal{P}_\Sigma{:}\, G_0 \xrightarrow{\theta_1}_{R4} G_1 \xrightarrow{\theta_2}_{R4} \cdots \xrightarrow{\theta_n}_{R4} G_n$$

が存在し，かつ $x{:}\, s \in EVar(G_0\theta_1^{\uparrow}\cdots\theta_n^{\uparrow}) - EVar(G_n)$ とする．このとき，ある $G_i, G_j (i < j)$ に対して，$x{:}\, s \in EVar(G_i\theta_j^{\uparrow}) - EVar(G_j)$ である．

(証明) 線形導出の長さ n に関する帰納法によって証明する．

$n = 1$ のとき，仮定より明らかである．

$n > 1$ のとき，$x{:}\, s \in EVar(G_0\theta_1^{\uparrow}\cdots\theta_n^{\uparrow})$ に対して，$x{:}\, s \in EVar(x'{:}\, s'\theta_n^{\uparrow})$ のような，$x'{:}\, s' \in EVar(G_0\theta_1^{\uparrow}\cdots\theta_{n-1}^{\uparrow})$ が存在する．このとき，次の場合分けを考える．(i)$x'{:}\, s' \in EVar(G_{n-1})$ のとき，$x{:}\, s \in EVar(x'{:}\, s'\theta_n^{\uparrow})$ なので，$x{:}\, s \in EVar(G_{n-1}\theta_n^{\uparrow})$ である．よって，$x{:}\, s \in EVar(G_{n-1}\theta_n^{\uparrow}) - EVar(G_n)$ である．(ii)$x'{:}\, s' \notin EVar(G_{n-1})$ のとき，帰納法の仮定により，$x{:}\, s \in EVar(G_0\theta_1^{\uparrow}\cdots\theta_{n-1}^{\uparrow})$ ならば，ある $G_i, G_j (i < j)$ に対して，$x'{:}\, s' \in EVar(G_i\theta_j^{\uparrow}) - EVar(G_j)$ である．($\theta_n^{\uparrow} = \theta_n \uparrow EVar(G_{n-1})$ より) $x'{:}\, s' = x'{:}\, s'\theta_n^{\uparrow}$ なので，$x{:}\, s = x'{:}\, s' (\in EVar(G_0\theta_1^{\uparrow}\cdots\theta_{n-1}^{\uparrow}\theta_n^{\uparrow}))$ となる． □

補題 4.6 (持ち上げ補題) $\mathcal{P}_\Sigma$ をプログラムとする．$\mathcal{P}_\Sigma$ の制限なし線形導出

$$\mathcal{P}_\Sigma{:}\, G_0\theta_0 \xrightarrow{\theta_1}_{R4} G_1 \xrightarrow{\theta_2}_{R4} \cdots \xrightarrow{\theta_n}_{R4} G_n$$

が存在するならば，$\mathcal{P}_\Sigma$ の線形導出

$$\mathcal{P}_\Sigma{:}\, G_0 \xrightarrow{\theta'_1}_{R4} G'_1 \xrightarrow{\theta'_2}_{R4} \cdots \xrightarrow{\theta'_n}_{R4} G'_n$$

が存在し，以下が成り立つ．

(i) $\gamma_0 = \theta_0$ であり，$1 \leqq i \leqq n$ に対して $(\gamma_{i-1}\uparrow EVar(G_{i-1}))\theta_i = \theta'_i\gamma_i$ かつ $G_i = G'_i\gamma_i$ である．

(ii) $G_0\theta_0\theta_1^{\uparrow}\cdots\theta_n^{\uparrow} = G_0\theta_1'^{\uparrow}\cdots\theta_n'^{\uparrow}\gamma_n'$ であるような代入 γ_n' が存在する．

(証明) 線形導出の長さ n に関する帰納法によって証明する．

$n=1$ のとき，制限なし線形導出 $\mathcal{P}_\Sigma\colon G_0\theta_0 \xrightarrow{\theta_1}_{R4} G_1$ が存在する．但し，$L\theta_0 \in G_0\theta_0$, $L' \leftarrow G' \in \mathcal{P}_\Sigma$ かつ，θ_1 が $L_0\theta_0 \in \langle L\theta_0\rangle$ と L' の単一化子である．ここで，$EVar(G_0) \cap EVar(L' \leftarrow G') = \emptyset$ なので，$(L' \leftarrow G')\theta_0 \upharpoonright EVar(G_0) = L' \leftarrow G'$ が言える．よって，$\mathcal{P}_\Sigma\colon G_0 \xrightarrow{(\theta_0 \upharpoonright EVar(G_0))\theta_1}_{R4} G_1$ となる．一方，θ_1' を L_0 と L' の最汎単一化子とする．このとき，$\mathcal{P}_\Sigma\colon G_0 \xrightarrow{\theta_1'}_{R4} G_1'$ が存在し，$(\gamma_0 \upharpoonright EVar(G_0))\theta_1 = \theta_1'\gamma_1$ かつ $G_1 = G_1'\gamma_1$ が成り立つ．

続いて，$G_0\theta_0\theta_1^{\uparrow} = G_0\theta_1'^{\uparrow}\gamma_1'$ を示す．上記より，$(\theta_0 \upharpoonright EVar(G_0))\theta_1 = \theta_1'\gamma_1$ なので，$G_0\theta_0\theta_1 = G_0\theta_1'\gamma_1$ である．故に，$G_0\theta_0\theta_1^{\uparrow} = G_0\theta_1'^{\uparrow}\gamma_1$ が成り立つ．

$n>1$ のとき，帰納法の仮定より線形導出

$$\mathcal{P}_\Sigma\colon G_0 \xrightarrow{\theta_1'}_{R4} G_1' \xrightarrow{\theta_2'}_{R4} \cdots \xrightarrow{\theta_{n-1}'}_{R4} G_{n-1}',$$

が存在する．但し，$\gamma_0 = \theta_0$ であり，$1 \leqq i \leqq n-1$ に対して $(\gamma_{i-1} \upharpoonright EVar(G_{i-1}))\theta_i = \theta_i'\gamma_i$ かつ $G_i = G_i'\gamma_i$ であるような代入 γ_i が存在する．このとき，前提より制限なし線形導出 $\mathcal{P}_\Sigma\colon G_{n-1} \xrightarrow{\theta_n}_{R4} G_n$ $(G_{n-1} = G_{n-1}'\gamma_{n-1})$ が存在する．このとき，$L \in G_{n-1}$, $L' \leftarrow G' \in \mathcal{P}_\Sigma$ かつ，θ_n が $L_0 \in \langle L\rangle$ と L' の単一化子である．ここで，$EVar(G_{n-1}') \cap EVar(L' \leftarrow G') = \emptyset$ なので，$(L' \leftarrow G')\gamma_{n-1} \upharpoonright EVar(G_{n-1}') = L' \leftarrow G'$ が言える．よって，$\mathcal{P}_\Sigma\colon G_{n-1}' \xrightarrow{(\gamma_{n-1} \upharpoonright EVar(G_{n-1}'))\theta_n}_{R4} G_n$ となる．一方，θ_n' を L_0 と L' の最汎単一化子とする．このとき，$\mathcal{P}_\Sigma\colon G_{n-1}' \xrightarrow{\theta_n'}_{R4} G_n'$ が存在し，$(\gamma_{n-1} \upharpoonright EVar(G_{n-1}))\theta_n = \theta_n'\gamma_n$ かつ $G_n = G_n'\gamma_n$ が成り立つ．

続いて，$G_0\theta_0\theta_1^{\uparrow}\cdots\theta_n^{\uparrow} = G_0\theta_1'^{\uparrow}\cdots\theta_n'^{\uparrow}\gamma_n'$ を示す．まず，代入 γ_i を拡張した γ_i' を定義する．ある $G_j, G_k (1 \leqq j < k \leqq n)$ に対して，$x\colon s \in EVar(G_j\theta_k^{\uparrow}) - EVar(G_k)$ ならば，$x\colon s\gamma_i' = x\colon s\gamma_{i-1}'\theta_i^{\uparrow}$ とする．$x\colon s \in EVar(G_{i-1}'\theta_i') \cup EVar(G_i')$(但し，$G_0' = G_0$) ならば，$x\colon s\gamma_i' = x\colon s\gamma_i$ とする．これにより，次の条件において $x\colon s\theta_n'^{\uparrow}\gamma_n' = x\colon s\gamma_{n-1}'\theta_n^{\uparrow}$ を示したい．(a) $x\colon s \in EVar(G_{n-1}')$ のとき，γ_i' の定義と $(\gamma_{n-1} \upharpoonright EVar(G_{n-1}))\theta_n = \theta_n'\gamma_n$ より，$x\colon s\theta_n'^{\uparrow}\gamma_n' = x\colon s\theta_n'^{\uparrow}\gamma_n = x\colon s\gamma_{n-1}\theta_n^{\uparrow} = x\colon s\gamma_{n-1}'\theta_n^{\uparrow}$ である．(b) ある $G_j, G_k (1 \leqq j < k \leqq n)$ に対して，$x\colon s \in EVar(G_j\theta_k^{\uparrow}) - EVar(G_k)$ のとき，$x\colon s\theta_n'^{\uparrow} = x\colon s$ である．よって，$x\colon s\theta_n'^{\uparrow}\gamma_n' = x\colon s\gamma_n' = x\colon s\gamma_{n-1}'\theta_n^{\uparrow}$ である．ここで，$y\colon s' \in EVar(G_0)$ かつ $y\colon s'\theta_1^{\uparrow}\cdots\theta_{n-1}'^{\uparrow} = t$ を考える．すると，補題 4.5 より，$Var(t)$ の要素 $x\colon s$ は，(a) または (b) のいずれかに該当する．従って，$y\colon s'\theta_0\theta_1^{\uparrow}\cdots\theta_n^{\uparrow} =_{I.H.} y\colon s'\theta_1'^{\uparrow}\cdots\theta_{n-1}'^{\uparrow}\gamma_{n-1}'\theta_n^{\uparrow} = t\gamma_{n-1}'\theta_n^{\uparrow} = t\theta_n'^{\uparrow}\gamma_n' = y\colon s'\theta_1'^{\uparrow}\cdots\theta_n'^{\uparrow}\gamma_n'$ が導かれる． □

定理 4.3（線形導出の完全性） $\mathcal{P}_\Sigma$ をプログラム，G をゴール，θ をソート代入とする．$\mathcal{P}_\Sigma \models_{\Sigma^+} G\theta$ であるならば，$G\theta = G\theta'^{\uparrow}\gamma$ であるような線形導出 $\mathcal{P}_\Sigma\colon G \xrightarrow{\theta'}\!\!\!\!\rightarrow \bot$ が存在する．

(証明) まず，すべての $x_i\colon s_i \in EVar(G\theta)$ に対して新しい定数 $c_i\colon s_i$ を導入する．代入 β を $Dom(\beta) = EVar(G\theta)$ かつ $\beta(x_i\colon s_i) = c_i\colon s_i$ と定義する．このとき，$\mathcal{P}_\Sigma \models_{\Sigma^+} G\theta$ から，$\mathcal{P}_\Sigma \models_{\Sigma^+} G\theta\beta$ が成り立つ．定理 4.2 より，線形導出 $\mathcal{P}_\Sigma\colon G\theta\beta \xrightarrow{\delta}\!\!\!\!\rightarrow \bot$ が言える．さらに，補題 4.6 は線形導出 $\mathcal{P}_\Sigma\colon G \xrightarrow{\theta'}\!\!\!\!\rightarrow \bot$ および $G\theta\beta\delta^{\uparrow} = G\theta'^{\uparrow}\gamma$ を導く．さらに，$G\theta\beta$ は変数を含まないので，$G\theta\beta = G\theta'^{\uparrow}\gamma$ となる．また，$G\theta'^{\uparrow}$ は新しい定数 $c_i\colon s_i$ を含まないはずなので，$x_i\colon s_i$ と同じソートをもつある変数 $y_i\colon s_i$ に対して $\gamma(y_i\colon s_i) = c_i\colon s_i$ が成り立つ．ここで，代入 γ_0 を $(y_i\colon s_i)\gamma_0 = x_i\colon s_i$ により定義し，$\gamma' = \{(a,b) \in \gamma \mid a \neq y_i\colon s_i\} \cup \gamma_0$ とする．故に，$G\theta = G\theta'^{\uparrow}\gamma'$ が証明できる． □

5. おわりに

本論文ではソートを単項述語として利用できるように，ソート述語を加えたソート付き論理プログラミング言語を形式化して，その完全な推論体系を実現した．Beierle によるソート述語をもつ推論体系を基に，論理プログラミング特有の線形導出による実装に向いた推論メカニズムを構築できた．具体的にはゴール節から空節を導く線形導出に，サブソートとソート述語に関する推論規則を組み込むことで完全な体系へと拡張させている．その完全性は，プログラムからの推論木によって標準的なモデルを定義し，それが Σ^+ モデルであることから証明される．その際，サブソート関係の解釈をソート述語を含んだ論理式の充足可能性に反映されることで，推論体系の拡張に対応した意味論を定義している．

本研究に関連して，知識表現の観点からソート付き

の論理プログラミングの研究がなされている．その一つとして，論理プログラミング言語 LOGIN [1] がある．LOGIN では，ψ 項という複雑な項表現に，順序と内部構造をもつ一種のソート表現 (ordered type structure と呼ばれる) を導入している．このアプローチでは，これまで述語で表していたもの (例えば，$person(x)$) を述語ではなく ψ 項内のソート表現で記述し推論の効率化を図っている．従って，本研究のように，知識表現を目的にソート述語を導入し，述語とソートとの間での融合を扱うことは対象ではない．他に，法的推論への応用のために，Kakuta ら [10] によるソート付きの論理プログラミングを用いた研究がある．この研究も LOGIN 同様，ソート述語を使った知識表現は行っていない．

本研究では，論理プログラミングを扱うことで実用的な側面を重要視しているが，その結果は厳密で形式的なものである．近年では，プログラム言語の安全性を理論的に示すのに，言語の形式化が不可欠と認識されている．例えば，関数型プログラミング言語 ML は，形式的な言語として理論的に整備されており，それを踏まえてコンパイラーなどが実装されている．同様に，本稿の結果により，ソート述語を含んだ論理プログラミング言語を実装する際に，完全な言語として理論的な保証を与えることができる．

文　献

[1] H. Aït-Kaci and R. Nasr. LOGIN: A logic programming language with built-in inheritance. *Journal of Logic Programming*, pages 185–215, 1986.

[2] H. Aït-Kaci and A. Podelski. Towards a meaning of LIFE. *Journal of Logic Programming*, pages 195–234, 1993.

[3] C. Beierle, U. Hedtsuck, U. Pletat, P.H. Schmitt, and J. Siekmann. An order-sorted logic for knowledge representation systems. *Artificial Intelligence*, 55:149–191, 1992.

[4] A. G. Cohn. Taxonomic reasoning with many sorted logics. *Artificial Intelligence Review*, 3:89–128, 1989.

[5] K. Doets. *From Logic to Logic Programming*. The MIT Press, 1994.

[6] A. M. Frisch. A genaral framework for sorted deduction: Fundamental results on hybrid reasoning. In *In Preceedings of the First International Conference on Principles of Knowledge Represntation and Reasoning*, 1989.

[7] M. Hanus. Logic programming with type specifications. In F. Pfenning, editor, *Types in Logic Programming*. The MIT Press, 1992.

[8] J. Herbrand. In W. D. Goldfarb, editor, *Logical Writtings*. Harvard University Press, 1971.

[9] P. M. Hill and R. W. Topor. A semantics for typed logic programs. In F. Pfenning, editor, *Types in Logic Programming*. The MIT Press, 1992.

[10] T. Kakuta, M. Haraguchi, and Y. Okubo. A goal-dependent abstraction for legal reasoning by analogy. *International Journal of Artificial Intelligence and Law*, 5(1-2):97–118, 1997.

[11] K. Kaneiwa and S. Tojo. Event, property, and hierarchy in order-sorted logic. In *Proceedings of the 1999 Int. Conf. on Logic Programming*, pages 94–108. The MIT Press, 1999.

[12] K. Kaneiwa and S. Tojo. An order-sorted resolution with implicitly negative sorts. In *Proceedings of the 2001 Int. Conf. on Logic Programming*, pages 300–314. Springer-Verlag, 2001. LNCS 2237.

[13] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *J. ACM*, 42(4):741–843, 1995.

[14] J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, 1987.

[15] A. Oberschelp. Untersuchungen zur mehrsortigen quantorelogik. *Mathematische Annalen 145*, pages 297–333, 1962.

[16] M. Schmidt-Schauss. *Computational Aspects of an Order-Sorted Logic with Term Declarations*. Springer-Verlag, 1989.

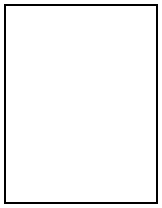
[17] J. C. Shepherdson. The role of standardising apart in logic programming. *Theoretical Computer Science*, 129(1):143–166, 1994.

[18] C. Walther. *A Many-Sorted Calculus Based on Resolution and Paramodulation*. Pitman and Kaufman Publishers, 1987.

[19] C. Walther. Many-sorted unification. *Journal of the Association for Computing Machinery*, 35:1, 1988.

[20] T. Weibel. An order-sorted resolution in theory and practice. *Theoretical Computer Science*, 185(2):393–410, 1997.

（平成 13 年 9 月 18 日受付，14 年 1 月 17 日再受付）



兼岩　憲　（正員）

1993〜1996 年富士通 (株) 勤務．1998 年北陸先端科学技術大学院大学情報科学研究科修士課程修了．2001 年同大学院情報科学研究科博士後期課程修了．現在，国立情報学研究所情報学基礎研究系助手．論理プログラミング，ソート論理に関する研究に従事．ソフトウェア科学会，情報処理学会，人工知能学会，ALP，ASL 各会員．