解 説

Prolog の 基 礎†

萩 谷 昌 己††

0. はじめに

本稿は，Prolog の基礎である，導出やユニフィケーションなどの概念を整理することを目的とする．

1 章では，いくつかの導出法を統一的に扱うために，証明項という概念を導入する．

2 章では，ユニフィケーションの定式化を，Prolog の実際のインプリメンテーションにより近い形で行うために，マルチ変数，マルチ項という概念を使って代入の定義を行う．これによって，mgu の唯一性などが，より厳密な形で成立する．

1 章の証明項と 2 章の mgu を使って，3 章で，導出の定義を与える．

4 章では最小不動点について，5 章では，最小不動点から自然に帰結される帰納法について述べる．

6 章で最大不動点意味論と有限的失敗について触れ，最後に 7 章で，Prolog の計算可能性についてまとめておく．

1. 証 明 項

本章では，証明項という概念を定義するが，その前に，本稿で用いる記号の説明と基本概念の定義を与えておく．

変数は，$x, y, \cdots$ 等で表す．

関数記号は，$f, g, \cdots$ 等で表す．各関数記号の項数 (arity) は特に明示しないが，関数記号ごとに定まっているものとする．0 項の関数記号は定数記号と呼ぶ．

【定義】 項は以下のように帰納的に定義される．

- 変数は項である．
- f が n 項 ($n \geqq 0$) の関数記号で，$t_1, \cdots, t_n$ が項のとき，$f(t_1, \cdots, t_n)$ は項である．

f が定数記号のとき，項 $f(\ )$ は，単に f と書く．

項は，$t, s, \cdots$ 等で表す．

述語記号は，$P, Q, \cdots$ 等で表す．各述語記号の項数も特に明示しないが，述語記号ごとに定まっているものとする．

【定義】 P が n 項 ($n \geqq 0$) の述語記号で，$t_1, \cdots, t_n$ が項のとき，$P(t_1, \cdots, t_n)$ は素論理式である．

素論理式は，$A, B, \cdots$ 等で表す．

項，素論理式が変数を含まないとき，「閉じている」という．

【定義】 $A, B, \cdots, B_n (n \geqq 0)$ が素論理式のとき，$A \leftarrow B_1, \cdots, B_n$ を，n 項の確定節と呼ぶ．

確定節は，$c, d, \cdots$ 等で表す．

$c = A \leftarrow B_1, \cdots, B_n$ のとき，A を c の結論，B_i $(1 \leqq i \leqq n)$ を c の条件という．

【定義】 確定節の有限集合をプログラムという．

プログラムは，$S, S', \cdots$ 等で表す．

本章の以下の部分では，プログラム S を固定して考える．

S の各確定節 c に対して，c と項数の等しい関数記号 f_c が対応しているものとする．f_c を c の名前という．もちろん，異なる確定節には，異なる名前が対応する．以下では，f_c を単に c と書くことにする．

$\perp$ を S に現れないような変数とする．

【定義】 S の証明項とは，次のように帰納的に定義される項のことである．$c \in S$ で，$t_1, \cdots, t_n$ が S の証明項または $\perp$ のとき，$c(t_1, \cdots, t_n)$ は S の証明項である．

［例 1.1］（例の中では，なるべくそれらしい記号を使うことにする．）$S = \{c_1, c_2, c_3, c_4\}$ とし，

$$c_1 = N(0) \leftarrow$$
$$c_2 = N(S(x)) \leftarrow N(x)$$
$$c_3 = Tree(x) \leftarrow N(x)$$
$$c_4 = Tree(cons(x, y)) \leftarrow Tree(x),\ Tree(y)$$

とおく．このとき，

$$c_4(c_3(c_1),\ c_3(c_2(\perp)))$$

は S の証明項である．

† Foundation of Prolog by Masami HAGIYA (Research Institute for Mathematical Sciences, Kyoto University).

†† 京都大学数理解析研究所

【定義】 t, s を項または素論理式とする．$t \equiv s$ を等式という．

本稿では，記述を簡単にするため，等式を扱う場合は，述語記号を関数記号とみなし，素論理式を項として扱うことにする．

【定義】 等式の有限列を等式列という．

等式列は，$E, E', \cdots$等で表す．空の等式列は ε で表す．等式列 E, E' が，等式の順番を入れ換えると等しくなるとき，$E \cong E'$ と書く．

確定節に現れる変数は，全称記号で束縛されていると考えられる．S の確定節を適用する場合，各変数を，考えている文脈に現れていないような新しい変数で置き換える操作が必要になる．このために，本稿では，インデックスというものを使うことにする．

【定義】 正整数の有限列をインデックスという．

インデックスは，$\sigma, \tau, \cdots$等で表す．空のインデックスは Λ で表す．

変数 x とインデックス σ に対して，インデックスつき変数 $x^{(\sigma)}$ が対応しているものとする．もちろん，x または σ が異なれば，$x^{(\sigma)}$ も異なる．インデックスつき変数は，通常のインデックスなし変数とは一致しない．インデックスつき変数にさらにインデックスをつけることは考えない．

プログラムの確定節に現れる変数にはインデックスはついていないとする．

【定義】 $c \in S$ とインデックス σ に対して，c の σ バリアントとは，c の中の各変数 x をインデックスつき変数 $x^{(\sigma)}$ で置き換えてできる確定節のことである．

項 t とインデックス σ に対して，t の中の各インデックスつき変数 $x^{(\tau)}$ を $x^{(\sigma,\tau)}$ で置き換えてできる項を，${}^{\sigma}t$ で表す．同様に，${}^{\sigma}E$ などの記法も用いる．

インデックスは，項の中の部分項（のオカレンス）を指定する．

【定義】 項 t の部分項（のオカレンス）の t におけるインデックスとは，以下のように帰納的に定義される．

- t 自身の t におけるインデックスは Λ である．
- $t=f(t_1, \cdots, t_n)$ で，t_i の部分項 s の t_i におけるインデックスが σ のとき，s の t におけるインデックスは i, σ である．

［例 1.2］ x の $cons(0, S(x))$ におけるインデックスは，2, 1 である．

【定義】 S の証明項 t に対して，素論理式 End(t) と等式列 Eqs (t) を次のように帰納的に定義する．

$t=c(t_1, \cdots, t_n)$ とし，$t_1, \cdots, t_n$ のうち$\perp$でないものを，$t_{i_1}, \cdots, t_{i_m}$ とする．c の Λ バリアントを，$A \leftarrow B_1, \cdots, B_n$ とおくと，

$$\mathrm{End}\,(t)=A,$$
$$\mathrm{Eqs}\,(t)=B_{i_1} \equiv {}^{i_1}\mathrm{End}\,(t_{i_1}), \cdots, B_{i_m} \equiv {}^{i_m}\mathrm{End}\,(t_{i_m}), {}^{i_1}\mathrm{Eqs}\,(t_{i_1}), \cdots, {}^{i_m}\mathrm{Eqs}\,(t_{i_m}).$$

［例 1.3］ 先の［例 1.2］のプログラムと証明項を考える．$t=c_4(c_3(c_1), c_3(c_2(\perp)))$ とおくと，

$$\mathrm{End}\,(t)= Tree\,(cons\,(x^{(\Lambda)}, y^{(\Lambda)})),$$
$$\mathrm{Eqs}\,(t)= Tree\,(x^{(\Lambda)}) \equiv Tree\,(x^{(1)}), Tree\,(y^{(\Lambda)}) \equiv Tree\,(x^{(2)}), N(x^{(1)}) \equiv N(0), N(x^{(2)}) \equiv N(S(x^{(2,1)})).$$

t, s を S の証明項，σ を$\perp$の（あるオカレンスの）t におけるインデックスとする．t の中の σ で指定される$\perp$のオカレンスを s で置き換えてできる S の証明項を，$t[s/\sigma]$ で表す．

【定義】 S の証明項 t と，$\perp$の t におけるインデックス $\sigma(\sigma \neq \Lambda)$ に対して，素論理式 Top (t, σ) を，以下のように帰納的に定義する．$t=c(t_1, \cdots, t_n)$ とおく．

- $\sigma=i(1 \leqq i \leqq n)$ のとき，c の Λ バリアントを $A \leftarrow B_1, \cdots, B_n$ とすると，$\mathrm{Top}\,(t, \sigma)=B_i$
- $\sigma=i, \tau(\tau \neq \Lambda)$ のとき，$\mathrm{Top}\,(t, \sigma)={}^{i}\mathrm{Top}\,(t_i, \tau)$.

［例 1.4］ t を［例 1.3］のようにおき，$\sigma=2, 1, 1$ とおくと，$\mathrm{Top}\,(t, \sigma)=N(x^{(2,1)})$

［補題 1.1］ t, s を S の証明項，σ を$\perp$の t におけるインデックスとする．

$$\mathrm{Eqs}\,(t[s/\sigma]) \cong \mathrm{Eqs}\,(t), \mathrm{Top}\,(t, \sigma) \equiv {}^{\sigma}\mathrm{End}\,(s), {}^{\sigma}\mathrm{Eqs}\,(s).$$

証明）略．□

2. ユニフィケーション

【定義】 変数の空でない有限集合を，マルチ変数という．

マルチ変数は，$X, Y, \cdots$等で表す．

【定義】 互いに素なマルチ変数の有限集合を，変数分割という．

変数分割は，$\Gamma, \Delta, \cdots$等で表す．

【定義】 変数分割 Γ 上のマルチ項* は以下のように帰納的に定義される．

- Γ のマルチ変数は Γ 上のマルチ項である．
- f が n 項の関数記号で，$t_1, \cdots, t_n$ が Γ 上のマルチ項のとき，$f(t_1, \cdots, t_n)$ は Γ 上のマルチ項である．

* 文献 1)の multiterm とはまったく関係がない．

マルチ項は，項と同じく，$t, s, \cdots$ 等で表す．

【定義】 Γ と Δ を変数分割，Σ を，Γ の部分集合から，Δ 上のマルチ項の集合への関数とする．Δ と Σ が以下の条件を満足するとき，Δ と Σ の組 $\langle\Delta, \Sigma\rangle$ を，Γ への代入という．

- $\Gamma\backslash\mathrm{Dom}(\Sigma)$ は，集合の分割として，Δ の細分になっている ($\mathrm{Dom}(\Sigma)$ は Σ の定義域で Γ の部分集合)．すなわち，Δ の元は，$\Gamma\backslash\mathrm{Dom}(\Sigma)$ の元のいくつかをマージしたもの．
- 任意の $X\in\mathrm{Dom}(\Sigma)$ に対して，$\Sigma(X)\notin\Delta$．すなわち，$\Sigma(X)$ は，マルチ変数ではないようなマルチ項．

代入は，$\theta, \theta', \cdots$ 等で表す．$\theta=\langle\Delta, \Sigma\rangle$ のとき，Δ を $\Delta(\theta)$，Σ を $\Sigma(\theta)$ と書く．

代入 $\theta=\langle\Delta, \Sigma\rangle$ に対して，$\mathrm{Dom}(\Sigma)=\{X_1, \cdots, X_n\}$，$\Sigma(X_i)=t_i(1\leqq i\leqq n)$ のとき，Σ を，$\{t_1/X_1, \cdots, t_n/X_n\}$ と書く．

本来ならば，Γ への代入 $\langle\Delta, \Sigma\rangle$ は，三つ組 $\langle\Gamma, \Delta, \Sigma\rangle$ で表さねばならないが，本稿では，簡単のため，Γ を省略する．以後，代入を扱う場合，それがどこへの代入なのかをよく注意する必要がある．

【定義】 θ を変数分割 Γ への代入，t を Γ 上のマルチ項とする．$\Delta(\theta)$ 上のマルチ項 $t\theta$ は以下のように帰納的に定義される．

- $t=X\in\Gamma$ の場合．$X\in\mathrm{Dom}(\Sigma(\theta))$ ならば，$t\theta=\Sigma(\theta)(X)$．$X\notin\mathrm{Dom}(\Sigma(\theta))$ ならば，$X\subseteq Y$ となる $Y\in\Delta(\theta)$ があるから，この Y に対して，$t\theta=Y$．
- $t=f(t_1, \cdots, t_n)$ ならば，$t\theta=f(t_1\theta, \cdots, t_n\theta)$．

【定義】 $\theta_1=\langle\Delta_1, \Sigma_1\rangle$ を変数分割 Γ への代入，$\theta_2=\langle\Delta_2, \Sigma_2\rangle$ を Δ_1 への代入とする．このとき，θ_1 と θ_2 の合成 $\theta_1\theta_2$ を次のように定義する．

$$\theta_1\theta_2=\langle\Delta_2, \Sigma_1\theta_2\cup\Sigma_2/\Gamma\rangle.$$

ただし，

$$\Sigma_1\theta_2=\{\Sigma_1(X)\theta_2/X\,;\ X\in\mathrm{Dom}(\Sigma_1)\}$$
$$\Sigma_2/\Gamma=\{\Sigma_2(Y)/X\,;\ X\in\Gamma, X\subseteq Y\in\mathrm{Dom}(\Sigma_2)\}$$

θ_1 と θ_2 の合成は，明らかに，Γ への代入になる．

［補題 2.1］ $\theta_1=\langle\Delta_1, \Sigma_1\rangle$ を変数分割 Γ への代入，$\theta_2=\langle\Delta_2, \Gamma_2\rangle$ を Δ_1 への代入，t を Γ 上のマルチ項とする．このとき，

$$(t\theta_1)\theta_2=t(\theta_1\theta_2)$$

証明) 場合分けによる．$t=X\in\Gamma$ の場合のみ考えればよい．$X\in\mathrm{Dom}(\Sigma_1)$ ならば，

$$(X\theta_1)\theta_2=\Sigma_1(X)\theta_2=(\Sigma_1\theta_2)(X)=X(\theta_1\theta_2).$$

$X\subseteq Y\in\Delta_1$ ならば，$Y\in\mathrm{Dom}(\Sigma_2)$ かどうかをさらに見る．$Y\in\mathrm{Dom}(\Sigma_2)$ ならば，

$$(X\theta_1)\theta_2=Y\theta_2=\Sigma_2(Y)=(\Sigma_2/\Gamma)(X)=X(\theta_1\theta_2).$$

$Y\subseteq Z\in\Delta_2$ ならば，$X\subseteq Z$ より，

$$(X\theta_1)\theta_2=Y\theta_2=Z=X(\theta_1\theta_2).\ \Box$$

［補題 2.2］ 代入 $\theta_1=\langle\Delta_1, \Sigma_1\rangle$，$\theta_2=\langle\Delta_2, \Sigma_2\rangle$，$\theta_3=\langle\Delta_3, \Sigma_3\rangle$ に対して，

$$(\theta_1\theta_2)\theta_3=\theta_1(\theta_2\theta_3).$$

証明) θ_1 を Γ への代入とすると，

$$\begin{aligned}
&(\theta_1\theta_2)\theta_3\\
&=\langle\Delta_2, \Sigma_1\theta_2\cup\Sigma_2/\Gamma\rangle\theta_3\\
&=\langle\Delta_3, (\Sigma_1\theta_2\cup\Sigma_2/\Gamma)\theta_3\cup\Sigma_3/\Gamma\rangle\\
&=\langle\Delta_3, (\Sigma_1\theta_2)\theta_3\cup\Sigma_2\theta_3/\Gamma\cup\Sigma_3/\Gamma\rangle\\
&=\langle\Delta_3, \Sigma_1(\theta_2\theta_3)\cup(\Sigma_2\theta_3\cup\Sigma_3/\Delta_1)/\Gamma\rangle\\
&=\langle\Delta_1, \Sigma_1\rangle\langle\Delta_3, \Sigma_2\theta_3\cup\Sigma_3/\Delta_1\rangle\\
&=\theta_1(\theta_2\theta_3).\ \Box
\end{aligned}$$

【定義】 t，s を変数分割 Γ 上のマルチ項とする．$t\equiv s$ を Γ 上の等式という．

【定義】 変数分割 Γ 上の等式の有限列を Γ 上の等式列という．

変数分割上の等式列も，$E, E', \cdots$ 等で表す．

【定義】 変数分割 Γ 上の等式列 $E=t_1\equiv s_1, \cdots, t_n\equiv s_n$ と，Γ への代入 θ に対して，

$$t_i\theta=s_i\theta\ (1\leqq i\leqq n)$$

が成り立つとき，θ を E のユニファイヤという．

【定義】 E を変数分割 Γ 上の等式列とする．Γ への代入 μ が次の二つの条件を満足するとき，μ は E の mgu であるという．

- μ は E のユニファイヤである．
- E の任意のユニファイヤ θ に対して，代入 θ' があって，$\theta=\mu\theta'$ となっている．

変数分割 Γ 上の等式列 $E=t_1\equiv s_1, \cdots, t_n\equiv s_n$ と，Γ への代入 θ に対して，等式列 $t_1\theta\equiv s_1\theta, \cdots, t_n\theta\equiv s_n\theta$ を，$E\theta$ と書く．

［定理 2.1］ 変数分割 Γ 上の等式列 E がユニファイヤを持てば，E の mgu が唯一存在する．

唯一性の証明) μ_1, μ_2 が共に mgu であるとする．

$$\mu_1=\mu_2\theta'_1,\quad \mu_2=\mu_1\theta'_2$$

とおく．代入とその合成の定義から，

- $\Delta(\mu_2)\backslash\mathrm{Dom}(\Sigma(\theta'_1))$ は $\Delta(\mu_1)$ の細分．
- $\Delta(\mu_1)\backslash\mathrm{Dom}(\Sigma(\theta'_2))$ は $\Delta(\mu_2)$ の細分．

したがって，$\Delta(\mu_1)=\Delta(\mu_2)$ となり，

$$\theta'_1=\theta'_2=\langle\Delta(\mu_1), \phi\rangle$$

がわかる．これから明らかに，$\Sigma(\mu_1)=\Sigma(\mu_2)$．$\Box$

存在性の証明) θ を E のユニファイヤとする．$j\geqq 0$

に対して，等式列 E_j，代入 μ_j, θ_j を，以下の 1)～3) を満たすように構成する．

1) $\theta=\mu_j\theta_j$

2) θ_j は E_j のユニファイヤ．

3) 任意の ρ に対して，ρ が E_j のユニファイヤならば，$\mu_j\rho$ は E のユニファイヤ．

$j=0$ に対しては，

$E_0=E$

$\mu_0=\langle\Gamma, \phi\rangle$

$\theta_0=\theta$

とおく．明らかに，1)～3)を満足する．

以下，$j>0$ に対して，1)～3)を満足する E_{j-1}，μ_{j-1}，θ_{j-1} から，E_j，μ_j，θ_j を作る．

$E_{j-1}=t_1\equiv s_1, \cdots, t_n\equiv s_n$

とおく．$1\leqq i\leqq n$ なる i を任意に選び，

$E'=t_1\equiv s_1, \cdots, t_{i-1}\equiv s_{i-1}$

$E''=t_{i+1}\equiv s_{i+1}, \cdots, t_n\equiv s_n$

とおく．t_i と s_i の形で場合分けをする．

I．$t_i=f(t'_1, \cdots, t'_m)$，$s_i=f(s'_1, \cdots, s'_m)$ の場合．

$E_j=E', t'_1\equiv s'_1, \cdots, t'_m\equiv s'_m, E''$

$\mu_j=\mu_{j-1}$

$\theta_j=\theta_{j-1}$

とする．1)～3)は明らか．

II．$t_i=s_i=X$ の場合．

$E_j=E', E''$

$\mu_j=\mu_{j-1}$

$\theta_j=\theta_{j-1}$

とする．1)～3)は明らか．

III．$t_i=X$，$s_i=Y$，$X\neq Y$ の場合．変数分割 Δ と $\Delta(\mu_{j-1})$ への代入 δ を，

$\Delta=(\Delta(\mu_{j-1})\backslash\{X, Y\})\cup\{X\cup Y\}$

$\delta=\langle\Delta, \phi\rangle$

とする．このとき，

$E_j=E'\delta, E''\delta$

$\mu_j=\mu_{j-1}\delta=\langle\Delta, \Sigma(\mu_{j-1})\delta\rangle$

とする (E_{j-1} は $\Delta(\mu_{j-1})$ 上の等式列)．θ_{j-1} は E_{j-1} のユニファイヤだから，

$X\theta_{j-1}=Y\theta_{j-1}.$

X，$Y\in\mathrm{Dom}\,(\Sigma(\theta_{j-1}))$ かどうかをみる．X，$Y\in\mathrm{Dom}\,(\Sigma(\theta_{j-1}))$ ならば，

$$\Sigma=\{Z\theta_{j-1}/Z\,;\,Z\in\mathrm{Dom}\,(\Sigma(\theta_{j-1})), Z\neq X,\ Z\neq Y\}\cup\{X\theta_j/X\cup Y\}$$

とおくと，

$\Sigma/\Delta(\mu_{j-1})=\Sigma(\theta_{j-1}).$

Δ への代入 θ_j を，

$\theta_j=\langle\Delta(\theta_{j-1}),\ \Sigma\rangle$

とすると，

$\delta\theta_j=\langle\Delta(\theta_{j-1}),\ \Sigma/\Delta(\mu_{j-1})\rangle=\theta_{j-1}$

が成り立つ．次に，X，$Y\notin\mathrm{Dom}\,(\Sigma(\theta_{j-1}))$ ならば，$X\theta_{j-1}=Y\theta_{j-1}\in\Delta(\theta_{j-1})$，$X\subseteq X\theta_{j-1}$，$Y\subseteq X\theta_{j-1}$ だから，θ_{j-1} は，Δ への代入とみなすこともできる．θ_{j-1} を Δ への代入とみなしたものを θ_j とすると，明らかに，

$\delta\theta_j=\theta_{j-1}.$

どちらにしても，

$\mu_j\theta_j=\mu_{j-1}\delta\theta_j=\mu_{j-1}\theta_{j-1}=\theta$

となる．これで 1)がいえた．E_j の定義から 2)も明らか．ρ を E_j のユニファイヤとすると，$\delta\rho$ は E_{j-1} のユニファイヤだから，$\mu_{j-1}\delta\rho=\mu_j\rho$ は E のユニファイヤである．これで 3)がいえた．

IV．$t_i=X$，$s_i=f(\cdots)$，または，$t_i=f(\cdots)$，$s_i=X$ の場合．θ_{j-1} は E_{j-1} のユニファイヤだから，X は $f(\cdots)$ には現れない．$\Delta(\mu_{j-1})$ への代入 δ を，

$\Delta=\Delta(\mu_{j-1})\backslash\{X\}$

$\Sigma=\{f(\cdots)/X\}$

$\delta=\langle\Delta, \Sigma\rangle$

とし，

$E_j=E'\delta, E''\delta$

$\mu_j=\mu_{j-1}\delta$

とする．また，$X\in\mathrm{Dom}\,(\Sigma(\theta_{j-1}))$ だから，Δ への代入 θ_j を，

$\theta_j=\langle\Delta(\theta_{j-1}), \Sigma(\theta_{j-1})\backslash\{X\theta_{j-1}/X\}\rangle$

とする．

$$\begin{aligned}\delta\theta_j &=\langle\Delta(\theta_j),\ \Sigma\theta_j\cup\Sigma(\theta_j)/\Delta(\mu_{j-1})\rangle\\ &=\langle\Delta(\theta_{j-1}), \{f(\cdots)\theta_j/X\}\cup\Sigma(\theta_j)\rangle\\ &=\langle\Delta(\theta_{j-1}), \{X\theta_{j-1}/X\}\cup(\Sigma(\theta_{j-1})\backslash\{X\theta_{j-1}/X\})\rangle\\ &=\theta_{j-1}\end{aligned}$$

1)～3)は，IIIの場合と同様に証明される．

θ_{j-1} は E_{j-1} のユニファイヤであるから，I～IVの場合以外はあり得ない．

IIIまたはIVを適用すると，$\Delta(\mu_j)$ の大きさは1だけ減る．IIIまたはIVが適用される間に，IとIIは高々有限回しか適用できないから，$l\geqq 0$ があって，

$E_l=\varepsilon$

となる．このとき，3)から，μ_l は E のユニファイヤになっていて，1)から，

$\theta=\mu_l\theta_l.$

E_i と μ_i の構成は i の選択だけにより，θ にまったく依存していない．θ は任意のユニファイヤとしてよい．したがって，μ_i は E_i の mgu である．□

［系］ $E \cong E'$ とする．E の mgu が存在すれば，E' の mgu も存在して，両者は等しい．

証明） mgu の構成は，等式列の中の等式の順序によらない．□

［系］ 変数分割 Γ 上の等式列 $E = E', E''$ に対して，μ' を E' の mgu，μ'' を $E''\mu'$ の mgu とする．すると，$\mu'\mu''$ は E の mgu である．

証明） ［定理 2.1］の手続きを，E の E' の部分にだけ適用してゆくと，やがて，E' の mgu μ' が求まる．このとき，$\mu' = \mu_j$ とすると，

$$E_j = E''\mu'$$

となっている．次に，$E''\mu'$ の mgu μ'' を求めれば，全体としては，E の mgu を求めたことになっているので，$\mu'\mu''$ は E の mgu である．□

特に，E' と E'' が，共通のマルチ変数を含まない場合は次のようになる．E' を等式分割 Γ' 上の等式列，E'' を等式分割 Γ'' 上の等式列とし，$\Gamma' \cup \Gamma'' = \phi$ とする．E' の mgu を μ', E'' の mgu を μ'' とすれば，

$$\langle \Delta(\mu') \cup \Delta(\mu''),\ \Sigma(\mu') \cup \Sigma(\mu'') \rangle$$

が E', E'' の mgu になる．以下では，これを，

$$\mu' \cup \mu''$$

と書く．

［定理 2.1］は，等式列から mgu を求めるアルゴリズムを与えている．t_i と s_i がⅠ～Ⅳの場合に入らないとき，すなわち，$t_i = f(\cdots)$, $s_i = g(\cdots)$, $f \neq g$ となっているとき，もとの E は mgu を持たないことがわかる．また，Ⅲで，X が $f(\cdots)$ に含まれていれば，E は mgu を持たない（出現チェック）．

各 j での i の選び方は任意である．求まる mgu はすべて等しい．

ⅣよりもⅠ～Ⅲを優先的に適用するという方法がある．こうすると，Ⅳが適用されるときは，等式はすべて $X \equiv f(\cdots)$ または $f(\cdots) \equiv X$ という形になっている．このとき，$f(\cdots)$ の中のマルチ変数 Y について，

$$X < Y$$

とおくと，$<$ の推移閉包 $\overset{*}{<}$ に対して，次のことが成り立つ．もし等式列が mgu（ユニファイヤ）を持てば，$\overset{*}{<}$ は well-founded である．$\overset{*}{<}$ に関して X が極大であるような等式 $X \equiv f(\cdots)$ または $f(\cdots) \equiv X$ にⅣを適用すると，

$$E'\delta = E', \quad E''\delta = E''$$

となる．すなわち，E_{j-1} から $t_i \equiv s_i$ を除けば E_j になる．$\mu_j = \mu_{j-1}\delta$ とするとき，実際に μ_j を計算せず，単に δ を記録するだけにすれば，mgu を求める際に，マルチ変数のマージ以外の代入を行う必要がない．以上が，Martelli と Montanari のアルゴリズム[1]の基礎であった．

等式の長さを n とすると，Martelli と Montanari のアルゴリズムは $O(n \log n)$ である．線型のアルゴリズムは，Paterson と Wegman が発見している[2]．また，ユニフィケーションの並列計算量に関しては，Yasuura の仕事[3]がある．

有限の項ではなく，サイクルを含む項（有理項）の間のユニフィケーションに関しては，Mukai[4]，Jaffar[5] らのアルゴリズムがある．

証明項 t に対して，End(t)，Eqs(t) に現れる変数の全体を V とし，変数分割 Γ を，

$$\Gamma = \{\{x\}\ ;\ x \in V\}$$

とおく．End(t)，Eqs(t) の変数 x はマルチ変数 $\{x\}$ とみなすことにする．このとき，Γ 上の等式列 Eqs(t) の mgu（ユニファイヤ）を，t の mgu（ユニファイヤ）という．

3. 導　出

まず，1章の証明項と2章の mgu を用いて，導出という概念を定義するが，その前に，証明すべきもの，すなわち，ゴールというものを定義せねばならない．

【定義】 $B_1, \cdots, B_n$ $(n \geq 0)$ が素論理式のとき，$\leftarrow B_1, \cdots, B_n$ を，n 項のゴール節と呼ぶ．

ゴール節は，$g, g', \cdots$ 等で表す．

ゴール節 g に対しても，確定節と同様に，g の名前 f_g が対応しているものとする．f_g は g と項数の等しい関数記号である．以下では，f_g を単に g と書く．

プログラム S とゴール節の組 $\langle S, g\rangle$ を考える．

【定義】 $\langle S, g\rangle$ の証明項とは，以下のように定義される項のことである．

- S の証明項は，$\langle S, g\rangle$ の証明項である．
- $t_1, \cdots, t_n$ が S の証明項または $\perp$ のとき，$g(t_1, \cdots, t_n)$ は $\langle S, g\rangle$ の証明項である．

$\langle S, g\rangle$ の証明項 t に対しても，等式列 Eqs(t) が定義できる．また，σ を $\perp$ の（あるオカレンスの）t におけるインデックス，s を S の証明項としたとき，$t[s/\sigma]$ も $\langle S, g\rangle$ の証明項になる．

$\langle S, g\rangle$ に対して，証明項の集合 I を，以下のようにおく．

$I=\{c(\perp, \cdots, \perp);\ c\in S$ または $c=g\}$

また，

$t_0=g(\perp, \cdots, \perp)$

としておく．

【定義】 $\langle S, g\rangle$ の証明項の有限列 $D=t_1, \cdots, t_m$ が以下の条件を満足するとき，Dを $\langle S, g\rangle$ の導出という．

- t_i $(1\leqq i\leqq m)$ は mgu を持つ．
- $1\leqq i\leqq m$ に対して，次のどちらかが成り立つ．
  1) $t_i\in I$.
  2) t_j, t_k $(1\leqq j, k<i)$ と$\perp$の t_j におけるインデックスσがあって，$t_i=t_j[t_k/\sigma]$.
- $t_m=g(t'_1, \cdots, t'_n)$ で，しかも，t_m は閉じている（$\perp$を含まない）．

［補題 1.1］と［定理 2.1］の系から，

$t_i=t_j[t_k/\sigma]$

のとき，t_i の mgu は，t_j と t_k の mgu を使って求められることがわかる．なぜなら，

$\mathrm{Eqs}(t_i)$

$\cong\mathrm{Eqs}(t_j), \mathrm{Top}(t_j, \sigma)\equiv^{\sigma}\mathrm{End}(t_k), {}^{\sigma}\mathrm{Eqs}(t_k)$

だから，t_i, t_j, t_k の mgu を μ_i, μ_j, μ_k としたとき，μ_i は，$\mu_j\cup{}^{\sigma}\mu_k$ と，$\Delta(\mu_i\cup{}^{\sigma}\mu_k)$ 上の等式列

$\mathrm{Top}(t_j, \sigma)\mu_j\equiv^{\sigma}(\mathrm{End}(t_k)\mu_k)$

の mgu の合成になる．

【定義】 導出の定義で，2)の場合に，いつも $t_k\in I$ となっているとき，Dを入力導出という．

【定義】 導出の定義で，2)の場合に，いつも t_k が閉じているとき，Dを単位導出という．

【定義】 導出の定義において，ある l $(1\leqq l\leqq m)$ があって，$1\leqq i\leqq l$ に対しては 1）が成り立ち，$l<i\leqq m$ に対しては 2)が成り立ち，2)の場合に，いつも $t_k\in I$, $j=i-1$ となっていて，しかも，$t_1=t_0$ であるとき，Dを線型入力導出という．

【定義】 線型入力導出Dにおいて，いつも σ が，t_j における$\perp$の最も左のオカレンスを指定しているとき，Dを順序つき線型入力導出という．

t を，$t=g(t'_1, \cdots, t'_n)$ という形をした，$\langle S, g\rangle$ の閉じた証明項とする．t が mgu を持てば，明らかに，t を最後の証明項とする導出が存在する．同様に，t を最後の証明項とする，入力導出，単位導出，線型入力導出，順序つき線型入力導出が存在する．この意味で，各導出法はすべて完全である．

本来ならば，導出は，証明項の単なる有限列ではなく，木構造を持ったものにすべきだが，ここでは，記法が煩雑になるのを防ぐために，妥協した．

Wolfram は，本稿と同様に，文献 6)で，導出を等式列の構成ととらえ，各導出法を統一的に扱おうとしている．

4. 最小不動点意味論

【定義】 プログラムSに現れる関数記号から作られる閉じた項の全体を，S の Herbrand 空間といい，$H(S)$ で表す．

【定義】 プログラムSに現れる述語記号と関数記号から作られる閉じた素論理式の全体を，SのHerbrand base といい，$\hat{H}(S)$ で表す．

【定義】 プログラム S の確定節 c の中の変数を $H(S)$ の元で置き換えてできる閉じた確定節を，c のインスタンスと呼ぶ．

【定義】 $M\subseteq\hat{H}(S)$ が次の条件を満たすとき，MをSのモデルという．Sの確定節のインスタンス $A\leftarrow B_1, \cdots, B_n$ に対して，$B_1, \cdots, B_n\in M$ ならば，必ず，$A\in M$.

Sのモデルの中で，$\subseteq$の意味で最小のものを，Sの最小モデルという．

$M\subseteq\hat{H}(S)$ に対して，$T_S(M)(\subseteq\hat{H}(S))$ を対応させる関数 T_S を，次のように定義する．

$T_S(M)$

$=\{A\in\hat{H}(S);\ S$ の確定節のインスタンス $A\leftarrow B_1, \cdots, B_n$ で，$B_1, \cdots, B_n\in M$ となるものがある$\}$

$T_S(M)=M$ となる $M(\subseteq\hat{H}(S))$ を，T_S の不動点という．

T_S の不動点の中で，$\subseteq$ の意味で最小のものを，T_S の最小不動点といい，lfp (T_S) で表す．

$T_S\uparrow n\subseteq\hat{H}(S)$ $(0\leqq n\leqq\omega)$ を次のように定義する．

- $T_S\uparrow 0=\phi$.
- $T_S\uparrow n=T_S(T_S\uparrow n-1)$ $(0<n<\omega)$.
- $T_S\uparrow\omega=\bigcup_{0\leqq n<\omega} T_S\uparrow n$.

［定理 4.1］[7] 次の三つの集合は等しい．

- Sの最小モデル
- lfp (T_S)
- $T_S\uparrow\omega$

証明) $M\subseteq\hat{H}(S)$ がSのモデルであるための必要十分条件は，$T_S(M)\subset M$ となることである．したがって，T_S の連続性，すなわち，$X_0\subseteq X_1\subseteq\cdots\subseteq\hat{H}(S)$ に

対して，

$$T_S(\bigcup_{0\leq n<\omega} X_n)=\bigcup_{0\leq n<\omega} T_S(X_n)$$

をいえば，束上の連続関数の一般論を用いて証明できる[7]．$A\in T_S(\bigcup_{0\leq n<\omega} X_n)$ とすると，S の確定節のインスタンス $A\leftarrow B_1, \cdots, B_n$ があって，$B_1, \cdots, B_n\in \bigcup_{0\leq n<\omega} X_n$. ある N があって，$B_1, \cdots, B_n\in \bigcup_{0\leq n\leq N} X_n$ となるから，$A\in T_S(X_N)$. これで，

$$T_S(\bigcup_{0\leq n<\omega} X_n)\subseteq \bigcup_{0\leq n<\omega} T_S(X_n)$$

がいえた．逆は，T_S の単調性から明らか．□

［定理 4.2］ $A\in\hat{H}(S)$, $g_A=\leftarrow A$ とする．次の二つは同値．

- $A\in T_S\uparrow\omega$.
- $\langle S, g_A\rangle$ の導出が存在する．

証明） $0\leqq n<\omega$ に対して，

- $T_0=\phi$.
- $T_n=\{c(t_1, \cdots, t_m)\,;\, c\in S, t_1, \cdots, t_m\in T_{n-1}\}$ $(n>0)$.

とおくと，$A\in M_n$ ならば，ある $t\in T_n$ があって，$g_A(t)$ は mgu を持つ（$A\in M_n$ をいうために使われた S の確定節のインスタンスを得るための代入を重ね合わせれば，ある $t\in T_n$ に対して，t がユニファイヤを持つことがわかる．ここでは，厳密な議論は省略する）．したがって，$g_A(t)$ を最後の証明項とする導出が存在する．逆も同様．□

5. 帰 納 法

最小不動点意味論は，いわゆる一般化された帰納的定義というものに対応している．帰納的定義からは，帰納法が自動的に生成される．本章では，帰納法の生成について簡単に述べる．

変数 x の入った論理式を $F[x], G[x], \cdots$ 等で表す．本稿では，論理式の一般的な定義は行わない．$F[x]$ の x を項 t で置き換えて得られる論理式は，$F[t]$ で表す．$F[x,y]$, $G[t,s]$ 等の記法も用いる．

プログラム S の中に，

$N(0)\leftarrow$

$N(S(x))\leftarrow N(x)$

という二つの確定節があり，述語 N を結論に持つ確定節は，この二つに限るとする．これから，

$$F[0]\wedge\forall x(N(x)\wedge F[x]\rightarrow F[S(x)])$$
$$\rightarrow\forall x(N(x)\rightarrow F[x])$$

という帰納法を得ることができる．これを一般化すると，以下のようになる[8),9)]．

n 項の述語記号 P に着目し，S の中で P を結論に持つ確定節を $c_1, \cdots, c_m$ とする．

$$c_i=P(t_1, \cdots, t_n)\leftarrow B_1, \cdots, B_l$$

とする．B_j に対して，

$$C_j=\begin{cases}F[s_1, \cdots, s_n] & \text{if } B_j=P(s_1, \cdots, s_n)\\ \text{true} & \text{otherwise}\end{cases}$$

とおく．c_i の中の変数を $x_1, \cdots, x_k$ とし，

$$G_i=\forall x_1\cdots\forall x_k(B_1\wedge\cdots\wedge B_l\wedge C_1\wedge\cdots\wedge C_l$$
$$\rightarrow F[t_1, \cdots, t_n])$$

とおく．すると，

$$G_1\wedge\cdots\wedge G_m\rightarrow\forall x_1\cdots\forall x_n(P(x_1, \cdots, x_n)$$
$$\rightarrow F[x_1, \cdots, x_n])$$

が得られる．

［例 5.1］［例 1.1］の $Tree$ に関して，

$$\forall x(N(x)\wedge true\rightarrow F[x])\wedge$$
$$\forall x\forall y(Tree(x)\wedge Tree(y)\wedge F[x]\wedge F[y]$$
$$\rightarrow F[cons(x,y)])$$
$$\rightarrow\forall x(Tree(x)\rightarrow F[x])$$

が得られる．

Prolog プログラムに関する検証系[9),10)]を構成するためには，帰納法の他に，Herbrand 空間の性質を表す公理系が必要になる．これは，等式に関する公理として表現することができる．一般の等式の公理の他に，Herbrand 空間に関して特に必要となるのは，次のようなものである[11)]．

- $f\neq g$ ならば，$f(t_1, \cdots, t_n)\neq g(s_1, \cdots, s_m)$.
- t が s の部分項として現れているならば，$t\neq s$.

6. 最大不動点意味論

T_S の不動点の中で，$\subseteq$ の意味で最大のものを，T_S の最大不動点といい，$\mathrm{gfp}(T_S)$ で表す．

順序数 α に対して，$T_S\downarrow\alpha\subseteq\hat{H}(S)$ を次のように定義する．

- $T_S\downarrow 0=\phi$.
- $T_S\downarrow\alpha=T_S(T_S\downarrow\alpha-1)$（$\alpha$ は極限順序数でない順序数）．
- $T_S\downarrow\alpha=\bigcap_{\beta<\alpha} T_S\downarrow\beta$（$\alpha$ は極限順序数）．

順序数の全体を O_n で表す．

［定理 6.1］[12)] 次の二つの集合は等しい．

- $\mathrm{gfp}(T_S)$
- $\bigcap_{\alpha\in O_n} T_S\downarrow\alpha$

証明） T_S は，$\subseteq$ の意味で単調だから，完備束上の単調関数の一般論を用いることができる．□

$T_S\downarrow\alpha+1=T_S\downarrow\alpha$ となる最小の順序数 α を $|T_S|$ で

表す．次のことが知られている[13]．

［定理 6.2］ プログラムSに対して，

$|T_S| \leqq \omega_1$

（ω_1 は，帰納的でない最小の順序数）

［定理 6.3］

$|T_S| = \omega_1$

となるプログラムSが存在する．

次章で述べるように，gfp(T_S) は一般には計算することができない．$T_S\downarrow\omega$ も計算することができないが，$T_S\downarrow\omega$ の補集合 $\hat{H}(S)\backslash T_S\downarrow\omega$ は，帰納的可算である．$\hat{H}(S)\backslash T_S\downarrow\omega$ の元は，有限的に失敗する素論理式として特長づけられる[9),11),12),14),15)]．

$g_A = \leftarrow A(A\in\hat{H}(S))$ とし，$\langle S, g_A\rangle$ の証明項 t, s に対して，$\perp$の t におけるインデックスσとSの証明項 s' があって $s=t[s'/\sigma]$ となっているとき，$t<s$ と書く．$<$の反射推移閉包を$\leqq$とおく．

$G_A = \{g_A(t)$; t は S の証明項$\}$

とする．$B\subseteq G_A$ に対して，

$G_A = \{s$; $t\in B$, $t\leqq s\}$

のとき，Bを G_A の基底という．

G_A の基底Bが有限で，Bのすべての証明項が mgu を持たないとき，Bを有限失敗基底という．

［定理 6.4］ $A\in\hat{H}(S)$ に対して，次の二つは同値．

- $A\in\hat{H}(S)\backslash T_S\downarrow\omega$.
- G_A は有限失敗基底を持つ．

証明）

- $T_0 = \{\perp\}$.
- $T_n = \{c(t_1, \cdots, t_n)$; $c\in S$, $t_1, \cdots, t_n \in T_{n-1}\}$ $(n>0)$.

とおくと，次の二つは同値．

- $A\in T_S\downarrow n$
- ある $t\in T_n$ があって，$g_A(t)$ は mgu を持つ．

したがって，$A\in\hat{H}(S)\backslash T_S\downarrow n$ ならば，$\{g_A(t)$; $t\in T_n\}$ は G_A の有限失敗基底になる．逆に，Bを G_A の有限失敗基底とすると，$n\geqq 0$ があって，

$B\subseteq \{s$; $t\in T_n$, $g_A(t)\leqq s\}$

このとき，$\{g_A(t)$; $t\in T_n\}$ も G_A の有限失敗基底だから，$A\in\hat{H}(S)\backslash T_S\downarrow n$ が成り立つ．□

7. 計算可能性

本章では，関数記号の（有限）集合の方があらかじめ定まっている場合を考える．Hを，その集合の関数記号から作られる閉じた項の全体とする．Hは，適当な記号化によって，自然数の全体，もしくは，その帰納的な部分集合と一対一に対応する．$M\subseteq H$ に対して，M が帰納的（帰納的可算，Π_1^0, Σ_1^1）であるとは，M に対応する自然数上の集合が帰納的（帰納的可算，Π_1^0, Σ_1^1）ということである．

プログラム S, S の中の1項の述語記号 P, 及び，$M\subseteq\hat{H}(S)$ に対して，

$M(P) = \{t\in H(S)$; $P(t)\in M\}$

とおく．

次のことが知られている[13]．

［定理 7.1］ $M\subseteq H$ に対して，M が帰納的可算（Π_1^0, Σ_1^1）であるための必要十分条件は，$H=H(S)$ となるプログラムSと，Sの中の1項の述語記号Pがあって，$M=$lfp$(T_S)(P)$ ($T_S\downarrow\omega(P)$, gfp$(T_S)(P)$) となることである．

参 考 文 献

1) Martelli, A. and Montanari, U.: An Efficient Unification Algorithm, ACM Trans. Prog. Lang. Syst., Vol. 4, No. 2 (1982).
2) Paterson, M. S. and Wegman, M. N.: Linear Unification, J. Comp. Syst. Sci., Vol. 16 (1978).
3) Yasuura, H.: On the Parallel Computational Complexity of Unification, ICOT TR-027(1983).
4) Mukai, K.: A Unification Algorithm for Infinite Trees, Proceedings of the 8th IJCAI (1983).
5) Jaffar, J.: Efficient Unification over Infinite Terms, Department of Computer Science, Monash University (1984).
6) Wolfram, D. A., Maher, M. J. and Lassez, J. L.: A Unified Treatment of Resolution Strategies for Logic Programs, Department of Computer Science, University of Melbourne (1984).
7) van Emden, M. H. and Kowalski, R. A.: The Semantics of Predicate Logic as a Programming Language, J. ACM, Vol. 23, No. 4 (1976).
8) Martin-Löf, P.: Hauptsatz for the Intuitionistic Theory of Iterated Inductive Definitions, Proceedings of the Second Scandinavian Logic Symposium (1970).
9) Hagiya, M. and Sakurai, T.: Foundation of Logic Programming Based on Inductived Definition, New Generation Computing, Vol. 2 (1984).
10) Clark, K. L. and Tarnlund, S.-A.: A First Order Theory of Data and Programs, IFIP 77 (1977).
11) Clark, K. L.: Negation as Failure, Logic and Databases, Gallaire, H. and Minker, J. (eds.), Plenum Press (1978).

12) Apt, K. R. and van Emden, M. H.: Contributions to the Theory of Logic Programming, J. ACM, Vol. 29, No. 3 (1982).
13) Blair, H. A.: The Recursion-Theoretic Complexity of the Semantics of Predicate Logic as a Programming Language, Inf. Control(1982).
14) Lloyd, J. W.: Foundations of Logic Programming, Technical Report 82/7, Department of Computer Science, University of Melbourne (1982).
15) Sato, T.: Negation and Semantics of Prolog Programs, Proceedings of the First International Logic Programming Conference (1982).

(昭和59年8月23日受付)