

Horn節集合による計算について

京都大学 工学部 石橋稔彦

鈴木 博

山崎 進

堂下修司

1. まえがき

Kowalski は、[1]において、第一階述語論理のHorn節集合によってプログラムを表現し、そのHorn節集合からの導出反ばくをプログラムの計算(実行)とみなして、Horn節集合による計算を提案した。それは、Horn節集合からの具体的な頂点節を指定した入力導出反ばくと捉えられる。このHorn節集合による計算は、チューリング機械で計算可能な関数や、部分的帰納関数をシミュレートできる。[2,3] 又、命題論理におけるHorn節集合による計算はP-completeである[5,11]という点で興味深い。本稿では、第一階述語論理におけるHorn節集合による計算の能力を、表現能力と停止性の点から吟味し、いくつかの結果を述べる。2章で、Horn節集合による計算を頂点

節を指定した入力反ばくとして定義し、入力導出と単位導出が等価であること[4]を拡張し、頂点節を指定した入力導出反ばくをその節を含む単位導出反ばくにより得る手続き^[6]に基づき Horn 節集合による計算を、指定した節を含む単位反ばくとして捉える。3章で、リカーシブ図式を Horn 節集合に変換して、リカーシブ図式の停止性が Horn 節集合による計算の停止性と等価であるようにできることを手順によって示す。次に、Horn 節集合による計算の停止性について論ずる。一般には、Horn 節集合による計算の停止性判定が非可解であることが証明されている[7]が、ここでは、Horn 節集合中のリテラルの項は、関数のネスティングがいくらでも許されている。本稿では、項に関数のネスティングがない場合の Horn 節集合を扱う。4章で Horn 節集合による計算の停止性判定問題の複雑さを論じ、命題論理における Horn 節集合による計算の複雑さの拡張結果と Horn 節集合による計算の非可解性を述べる。

2. Horn 節集合による計算

2.1 Horn 節集合の定義

第一階述語論理とそこにおける導出および(導出)反ばくに関する諸定義は、[4]に従う

[定義 2.1] (Horn 節集合の定義)

Horn 節集合は、Horn 節の集合である。Horn 節は高ター

つしかポジティブリテラルを含まないリテラルの論理和である。ポジティブリテラルは、否定記号を持たない原子論理式である。否定記号のある原子論理式をネガティブリテラルとも呼ぶ。■

以下、ポジティブリテラルは、原子論理式に '+' をつけて表わし、ネガティブリテラルは、原子論理式に '-' をつけて表わす。節は、リテラルを一行に並べて表現する。

定義2.1より、4種類のHorn節が考えられ、それぞれをプログラムの文とみなした時の意味は、次のようになる。ここで、 $P, N_1, N_2, \dots, N_i$ は原子論理式、 $\square$ は空節を表わす。

- (i) $+P - N_1 - N_2 \dots - N_i$: 手続き宣言を意味する。 $+P$ は手続き名、 $-N_1 - N_2 \dots - N_i$ は手続き本体を意味し、各 N_j は手続き呼び出しとなる。
- (ii) $+P$: 本体なしの手続き宣言で、事実の表明を意味する。
- (iii) $-N_1 - N_2 \dots - N_i$: 各 N_j が意味する手続き呼び出しのすべてが実行されなければならないことを示す実行文を意味する。
- (iv) $\square$: 停止文を意味する。

(i), (iii), (iv) の形のHorn節をそれぞれ、以下、手続き宣言、実行文、停止文と呼ぶ。Horn節集合表現のプログラムは、実行文を含むHorn節からなる。

2.2 Horn節集合による計算

(1) に示されているHorn節集合表現のプログラムの実行は、第一階述語論理における包含 $A \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_m$ において、 A

を手続き名, 各 B_i を手続き呼び出し、として、包含と手続き宣言と解釈することに基礎をおいている。これは、各 B_i のすべてが、実行されたとき、 A なる手続きが、実行されることを意味している。従って、 A が実行されてしまうには、各 B_i なる手続きが実行されなくてはならず、 $B_i \leftarrow C_{i1} \wedge C_{i2} \wedge \dots \wedge C_{ik} \wedge \dots \wedge C_{im}$ なる手続き宣言の実行が、必要である。この過程は、 $A \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_i \wedge \dots \wedge B_m$ より、 $(A \leftarrow B_1 \wedge B_2 \wedge \dots \wedge C_{i1} \wedge C_{i2} \wedge \dots \wedge C_{im} \wedge \dots \wedge B_m) \cdot \theta$ (θ は置換) が、得られることである。これは、第一階述語論理の導出である。従って、Horn 節集合による計算は、与えられた Horn 節集合からの導出による空節の演繹であると捉えられる。

実行文と他の一つの Horn 節とから、導出をとる際の置換が、計算における入力と出力の関係を表現しているとみなせる^[1]。特に、実行文が含むある変数 x の他の項 t への置換 $\theta = t/x$ は、実行文が他の Horn 節 C_i と導出を行なう際の C_i からの出力とみなせる。従って、Horn 節集合のプログラムの計算結果は、与えられた Horn 節集合が含む実行文から始まる計算において、実行文が含む変数の他の項への置換の系列の結果である。

Horn 節集合について次の命題が、成立している。

[命題 2.1]^[5] S が充足不能な Horn 節集合ならば、 S からの入力反ばくが存在する。 ■

この命題より Horn 節集合による計算を次のように定義する。
 [定義 2.2] Horn 節集合による計算とは、与えられた Horn 節集合内の一つの節を頂点節とする入力反ばくである。 ■

[4] に入力導出と単位導出の等価性が示されているが、これを拡張して、頂点節を指定した入力反ばくの存在判定を単位導出で行なう手続きが、[6] に示されている。従って、Horn 節集合による計算は、与えられた Horn 節集合の指定された節を含むような単位反ばくとして捉えることができる。

3. リカーシブ図式における計算と Horn 節集合による計算

本章では、Horn 節集合による計算の表現性として、リカーシブプログラムのシミュレーションを検討する。すなわち、リカーシブ図式から Horn 節集合を構成する手順を与え、それによれば、図式の計算の停止性と Horn 節集合による計算の停止性の等価性が成立することを論ずる。

[定義 3.1]^[8] リカーシブ図式 $\mathcal{S}$ のアルファベット $\Sigma_{\mathcal{S}}$ は次の記号の有限部分集合である。

1. 定数: n 変数関数記号 $g_i^n (i \geq 1, n \geq 0)$ 。 g_i^0 を個体定数と呼び a_i と書く
 n 変数述語記号 $f_i^n (i \geq 1, n \geq 0)$ 。 f_i^0 は命題定数と呼ぶ。
2. 変数: 入力変数 $\{x_1, x_2, \dots\}$ 、プログラム変数 $\{y_1, y_2, \dots\}$ 、出力変数
 z 、関数変数 $\{f_1, f_2, \dots\}$

$\Sigma_{\mathcal{S}}$ 上に現われる入力変数 $\{x_1, \dots, x_n\}$ 、プログラム変数 $\{y_1, \dots, y_m\}$ 、関

数変数 $\{f_1, \dots, f_n\}$ をそれぞれ、 $\bar{x}, \bar{y}, \bar{f}$ と書く。 f_i^n, f_j^n, g_k^n の上添字 n は、省略する。 ■

[定義 3.2]^[8] Σ_s 上の量記号なしの命題 π は、 Σ_s 上の g_i, f_i, x_i, y_i, z とから構成される普通の意味での量記号なしの命題である。 Σ_s 上の項とは、 Σ_s 上の g_i, f_i, x_i, y_i, z で構成される普通の意味での項である。 Σ_s 上の条件項は、次のように帰納的に定義される。1. Σ_s 上の項は、条件項である。2. 未定義記号 ω は Σ_s 上の条件項である。3. π が Σ_s 上の量記号なしの命題であり、 τ_1, τ_2 が Σ_s 上の条件項であるとき、 $\text{if } \pi \text{ then } \tau_1 \text{ else } \tau_2$ は条件項である。 Σ_s 上の条件項とは、1, 2, 3 を有限回適用して得られるものだけを言う。^[8] 条件項、関数項、関数変数項、if 項と呼ぶ。 ■

[定義 3.3] 次の定義を行なう。(i) 変数 x_i, y_i , 定数 a_i は基項である。(ii) 関数項 $g_j(\tau_1, \dots, \tau_n)$ があって、すべての τ_i が変数又は定数であるとき、 $g_j(\tau_1, \dots, \tau_n)$ は、基項である。 ■

[定義 3.4]^[8] リカーシブ図式 $\mathcal{S}$ は次のような式の集まりとする。

$$z = \tau_0 < \bar{f}, \bar{x} > \quad \text{where}$$

$$f_1(\bar{x}, \bar{y}) \Leftarrow \tau_1 < \bar{f}, \bar{x}, \bar{y} >$$

$$\vdots$$

$$f_m(\bar{x}, \bar{y}) \Leftarrow \tau_m < \bar{f}, \bar{x}, \bar{y} >$$

$\tau_0 < \bar{f}, \bar{x} >$ は、 $\bar{x}, \bar{f}$ 以外の変数を含まない条件項、 $\tau_i < \bar{f}, \bar{x}, \bar{y} >$ は、 $\bar{x}, \bar{y}, \bar{f}$ 以外の変数を含まない条件項である。

[定義 3.5]^[8] リカーシブ図式 $\mathcal{S}$ の解釈 $\mathcal{I}$ は、次からなる。

1. 空でない集合 ID 。2. τ_i の定数の割当て: (a) 関数記号 g_i^n に ID^n から ID への全域関数を割当てる。個体定数には $a \in ID$ を割当てる。(b) 述語記号 p_i^n に ID^n から $\{\text{真}, \text{偽}\}$ の全域関数を割当てる。■

[定義 3.6]^[8] 組 $IP = \langle \mathcal{S}, \mathcal{I} \rangle$, $\bar{\xi}$ に対し計算は次の 1, 2 によって得られる項の列 $\alpha_0, \alpha_1, \dots$ である。1. α_0 は、 $\tau_i \langle \bar{f}, \bar{\xi} \rangle$ を簡約したものである。2. α_{i+1} ($i \geq 0$) は、 α_i に現われる最も左側の最も内側の $f_i(\bar{\xi}, \tau_i)$ を $\tau_i \langle \bar{f}, \bar{\tau}, \tau_i \rangle$ で置き換え簡約して α_i から得られた項である。簡約とは、次の 1, 2, 3 からなる。1. 任意の量記号なしの命題をその値 (真又は偽) で置き換える。2. $\text{if 真 then } A \text{ else } B$ を A で $\text{if 偽 then } A \text{ else } B$ を B で置き換える。3. $f_i(\tau_1, \dots, \tau_n)$ で各 τ_i が ID の元であるようなものはその値で置き換える。計算列が有限でかつ α_k が最終項である必要十分条件は、 α_k が ID の元であるか、 α_k が ω を含むときである。 α_k が ID の元 s であるとき $\text{val} \langle IP, \bar{\xi} \rangle$ は定義されたと言い、 $\alpha_k = s * \omega$ に対し $s = \text{val} \langle IP, \bar{\xi} \rangle$ と書く。

$\alpha_k = \omega$ のとき $\text{val} \langle IP, \bar{\xi} \rangle$ は未定義である。 ■

[定義 3.7] リカーシブ図式 $\mathcal{S}$ が停止するとは、あらゆる解釈 $\mathcal{I}$ と $\bar{\xi}$ に対し、 $\text{val} \langle \mathcal{S}, \mathcal{I}, \bar{\xi} \rangle$ が定義されることである。 ■

リカーシブ図式 $\mathcal{S}$ を次に示す手順で Horn 節集合 HS に変える。

手順 3.1]

入力: リカーシブ図式 $\mathcal{S}$

出力: Horn節集合 HS

記法: $F_i, Q_{\tau_0}, Q_{g_j}, W, P_i$ は述語記号、 x_i, y_i, z_i, u_i は変数を示す。 $first, second$ は、直積の第一要素、第二要素をとり出す手続きを示す。 $\odot$ で論理和を、 $\cup$ で集合の和を表わす。

procedure HORNSET(S):

begin $HS \leftarrow \phi$;

if τ_0 が w の場合 then $HS \leftarrow \{-W\}$

else if $\tau_0 \langle \bar{f}, \bar{x} \rangle$ が基項 σ の場合

then $HS \leftarrow \{-Q_{\tau_0}(\bar{x}, z)\} \cup \{+Q_{\tau_0}(\bar{x}, \sigma)\}$

else begin $H_0 \leftarrow \text{HORN}(\tau_0 \langle \bar{f}, \bar{x} \rangle, z)$;

$HS \leftarrow HS \cup \{first(H_0)\} \cup second(H_0)$

end;

for 各 $f_i(\bar{x}, \bar{y}) \leftarrow \tau_i \langle \bar{f}, \bar{x}, \bar{y} \rangle$ do

if $\tau_i \langle \bar{f}, \bar{x}, \bar{y} \rangle$ が w の場合

then $HS \leftarrow HS \cup \{+F_i(\bar{x}, \bar{y}, u_i) - W\}$

else begin

$I_i \leftarrow \text{HORN}(\tau_i \langle \bar{f}, \bar{x}, \bar{y} \rangle, u_i)$;

$HS \leftarrow HS \cup \{+F_i(\bar{x}, \bar{y}, u_i) \odot first(I_i)\}$

$\cup second(I_i)$

end;

return HS

end

procedure HORN(τ, v):

begin HS $\leftarrow \phi$;

if τ が w である then $H \leftarrow (-w, \phi)$;

if τ が基項である then $H \leftarrow (-Q_r(\sigma, v), \{+Q_r(\sigma, \sigma)\})$;

if τ が関数変数項 $f_j(\tau_1, \tau_2, \dots, \tau_n)$ である

then begin

$HC \leftarrow -f_j(z_1, \dots, z_n, v)$;

for $i \leftarrow 1$ until n do

if τ_i が基項である

then HC の z_i を τ_i で置き換える

else begin $H_i \leftarrow \text{HORN}(\tau_i, z_i)$;

$HC \leftarrow HC \odot \text{first}(H_i)$;

$HS \leftarrow HS \cup \text{second}(H_i)$;

end;

$H \leftarrow (HC, HS)$

end;

if τ が関数項 $g_j(\tau_1, \tau_2, \dots, \tau_n)$ である

then begin

$HC \leftarrow -Q_{g_j}(z_1, \dots, z_n, v)$; $HS_0 \leftarrow \{+Q_{g_j}(z_1, \dots, z_n, g_j(z_1, \dots, z_n))\}$;

for $i \leftarrow 1$ until n do

if τ_i が基項である

then HC, HS_0 の z_i を τ_i で置き換える

else begin $H_i \leftarrow \text{HORN}(\tau_i, z_i)$;

$HC \leftarrow HC \odot \text{first}(H_i)$;

$HS \leftarrow HS \cup \text{second}(H_i)$

end;

$HS \leftarrow HS \cup HS_0$; $H \leftarrow (HC, HS)$

end;

if τ が (if 項) $\text{if } P_i(\tau_1, \dots, \tau_n) \text{ then } \tau_1 \text{ else } \tau_2$ である

then begin

$HC \leftarrow \{-P_i(\tau_1, \dots, \tau_n, z_p) \odot -P_b(z_p, \bar{x}, \bar{y}, v)\}$;

```

for  $i \leftarrow 1$  until 2 ( $i \neq p_i$ ) do
  if  $\tau_i$  が  $w$  である then  $HS_i \leftarrow \{+P_b(z_p, \bar{x}, \bar{y}, v) \odot -w\}$ 
  else if  $\tau_i$  が基項である
    then  $HS_i \leftarrow \{+P_b(z_p, \bar{x}, \bar{y}, \tau_i)\}$ 
    else begin  $H_i \leftarrow \text{HORN}(\tau_i, v);$ 
                $HS_i \leftarrow \{+P_b(a_i, \bar{x}, \bar{y}, v) \odot \text{first}(H_i)\}$ 
                $\cup \text{second}(H_i)$ 
            end;
   $H \leftarrow (HC, HS_1 \cup HS_2)$ 
end;
return  $H$ 
end

```

(注) $P_i(\tau_{p_1}, \dots, \tau_{p_n}, z_p) \equiv (z_p = \tau_i(\tau_{p_1}, \dots, \tau_{p_n}))$ とする。又 $a_1 \equiv 1, a_2 \equiv 0$ とする。 ■

【命題 3.1】 τ が関数変数や w を含まないとき、 $H \leftarrow \text{HORN}(\tau, v)$ に対し、 $\{\text{first}(H)\} \cup \text{second}(H)$ から、 $\text{first}(H)$ を含んで単位反ばく ($\text{first}(H)$ を頂点節として入力反ばく (2章)) がある。 ■

証明)

1. τ が基項 σ のとき、 $\{\text{first}(H)\} \cup \text{second}(H) = \{-Q_\sigma(\sigma, v), +Q_\sigma(\sigma, \sigma)\}$ によって単位反ばくがある。

2. $\tau = g_j(\tau_1, \dots, \tau_n)$ のとき、各 $H_i \leftarrow \text{HORN}(\tau_i, v_i)$ について $\{\text{first}(H_i)\} \cup \text{second}(H_i)$ から $\text{first}(H_i)$ を含んで単位反ばく ($\text{first}(H_i)$ を頂点節として入力反ばく (2章)) があると仮定する。 $H \leftarrow \text{HORN}(\tau, v)$ に対し、 HS から、 $\text{first}(H) = -Q_{g_j}(x_1, \dots, x_n, v) \odot_x \text{first}(H_i)$ を頂点節とする入力導出で、

$-Q_{gj}(t_1, \dots, t_n, v)$ を導ける。しかも、 $H \text{ is } \exists + Q_{gj}(z_1, \dots, z_n, g_j(z_1, \dots, z_n))$ だから、入力反ばくが存在する。

3. $T = \text{if } \mathcal{F}_i(t_{p_1}, \dots, t_{p_n}) \text{ then } T_1 \text{ else } T_2$ のとき、 $H_i \leftarrow \text{HORN}(T_i, v_i)$ ($i=1, 2$) について、 $\{\text{first}(H_i)\} \cup \text{second}(H_i)$ から、 $\text{first}(H_i)$ を含んで単位反ばく ($\text{first}(H_i)$ を頂点節として入力反ばく (2章)) があると仮定すると、 $H \leftarrow \text{HORN}(T, v)$ について、 H から $\text{first}(H) = -P_i(t_{p_1}, \dots, t_{p_n}, z_p) - P_0(z_p, \bar{x}, \bar{y}, v)$ を頂点節として、 $\text{first}(H_1)$ or $\text{first}(H_2)$ の入力導出があるので、入力反ばくが存在する。1, 2, 3 より命題は証明された。■

[命題 3.2] リカーシブ図式 $\mathcal{S}$ が停止することと H から単位反ばくが存在することと等価である。

証明)

(1) $\bar{x} \Leftarrow 0$ (基項) のときは、命題が成立する。

(2) $\mathcal{S} : \bar{x} \Leftarrow T_0 \langle f_1, f_2, \dots, f_m, \bar{x} \rangle$ where $\{f_i \Leftarrow T_i\}_{i=1, \dots, m}$ に対し、 $\mathcal{S}$ が停止する $\Leftrightarrow H$ から単位反ばくがあると仮定する。

$\mathcal{S} : \bar{x} \Leftarrow T_0 \langle f_1, f_2, \dots, f_m, f_{m+1}, \bar{x} \rangle$ where $\{f_i \Leftarrow T_i\}_{i=1, \dots, m+1}$ とする。

$\mathcal{S}$ は停止する $\Leftrightarrow (\exists i) f_i \Leftarrow T_i$ (T_i は等価的に関数変数や ω を含まない) について、 $\mathcal{S}_1 : \bar{x} \Leftarrow T_0 \langle f_1, f_2, \dots, f_{i-1}, T_i, f_{i+1}, \dots, f_{m+1}, \bar{x} \rangle$ where $\{f_i \Leftarrow T_i\}_{i=1, \dots, m+1}$ が停止する。

T_i が、関数変数や ω を含まないとき、命題 3.1 により、 $I_i \leftarrow \text{HORN}(T_i, u_i)$ に対し、 $\text{first}(I_i)$ を含む $\{\text{first}(I_i)\} \cup$

second(I_i) からの単位反ばくがあるので、 HS からの $+F_i(\bar{x}, \bar{y}, u_i)$ の単位導出演繹 (導出形) が存在する。 ($\exists i$) $f_i \Leftarrow \tau_i$ (τ_i は等価的に関数変数や w を含まない) で、それに対応して、 $+F_i(\bar{x}, \bar{y}, u_i)$ の導出形の存在を仮定すれば、 $\mathcal{S}_1$ に対して、手順 3.1 によって構成される節集合を HS_1 とすると、 $\mathcal{S}_1$ が停止する $\Leftrightarrow HS_1$ からの単位反ばくが存在する。 $\Leftrightarrow HS$ からの単位反ばくが存在する。

$\mathcal{S} : z \Leftarrow \tau_0 \langle f_1, f_2, \dots, f_{m+1}, \bar{x} \rangle$ where $\{f_i \Leftarrow \tau_i\}_{i=1, \dots, m+1}$ に対して、すべての τ_i が関数変数をもつなり w を含むときは、 $\mathcal{S}$ は停止せず、 HS は、 $\{+F_i(\bar{x}, \bar{y}, u_i) \textcircled{1} \text{first}(I_i)\}$ ($I_i \Leftarrow \text{HORN}(\tau_i, u_i)$) の存在により反ばくがない。以上より、命題は、成立する。 ■

[例 4.1] 次のリカーシブ図式 $\mathcal{S}$ を手順 3.1 によって、 HS に変形する。

$\mathcal{S} : z \Leftarrow f(x_1, x_2)$ where

$f(x, y) \Leftarrow \text{if } \mathcal{P}(x) \text{ then } b \text{ else } f(g(x), f(x, y))$

$HS = \{ -F(x_1, x_2, z),$
 $+F(x, y, u) - P(x, z_p) - P_b(z_p, x, y, u),$
 $+P_b(a_1, x, y, b),$
 $+P_b(a_2, x, y, u) - F(g(x), z_2, u) - F(x, y, z_2) \}$

HS から反ばくがなく、 $\mathcal{S}$ は停止しない。 ■

4. Horn 節集合による計算の複雑さ

2章に基き、本章では、Horn 節集合による計算の複雑さを項に関数のネスティングがない Horn 節集合で、各節が高々1変数、各節が高々2変数を含む場合について考察する。

4.1 各節の変数の数が高々1の場合

(a) 各節が1引数関数のみともち関数のネスティングがない場合:

この場合の節集合は、純粹第一階述語論理において冠頭標準形の前置部が、 $\exists x \forall y *$ (*は任意個続く) である Ackermann class (ACK) の論理式の節集合とみなせる。ACKの充足不能性判定は PSPACE-hard である。^[9] [7]での議論に従い、次の命題を得る。

[命題 4.1] 変数が高々1で各節が1引数関数ともち、関数のネスティングがない Horn 節集合の場合、それによる計算の複雑さは、PSPACE-hard である。 ■

[定義 4.1] ACKに属する節集合で、各節中の変数を含むリテラルが高々2個のモナディッククラスを ACK₂ と呼ぶ。 ■

[命題 4.2]^[12] Horn 節集合が、ACK₂ に属する場合、それによる計算の複雑さは、P-complete である。 ■

本命題は、命題論理における Horn 節集合による計算の複雑さが、P-complete であるという結果の拡張となっている。

(b) 一般の場合:

[10]での議論より、次の命題が、成立する。

[命題 4.3] 各節の変数が高々1で関数のネスティングがないHorn節集合の場合、計算の停止性判定は、一般に非可解である。

4.2 各節の変数の数が高々2の場合

(a) 各節が2引数関数のみともち関数のネスティングがない場合:

この場合の節集合は、純粋の第一階述語論理における冠頭標準形の前置部が、 $\exists^* \forall \forall \exists^*$ ($\forall$ は任意個続く)であるGödel class (GDL)の論理式の節集合とみなせる。GDLの充足不能性は可解なので、次の命題が成立する。

[命題 4.4] 各節の変数が高々2で、各節の2引数関数のみともち、関数のネスティングがないHorn節集合の場合、計算の停止性判定は可解である。 ■

(b) 一般の場合:

[命題 4.5] 各節の変数が高々2で1引数関数のみともちHorn節集合の場合、計算の停止性判定は非可解である。各節の変数が高々2で関数のネスティングがないHorn節集合の場合、計算の停止性判定は、一般に非可解である。 ■

6. おまけ

本稿では、Horn節集合による計算の能力上の問題をその表現性と停止性の点から検討した。リカーシブ図式とHorn節集合に変換し、図式の計算をHorn節集合による計算でシミュレートできることを論じ、一斉に、Horn節集合による計算の停止性を

項に関数のネスティングがない場合について論じた。そこでは、命題論理における Horn 節集合による計算が P-complete であるという著しい特徴を拡張した結果を含んでいる。又、本稿での議論は、チューリング機械による計算、部分的帰納関数を Horn 節集合による計算で、シミュレートできるという結果^[2,3]をリカーシブ関式に対し、実証するものである。

〔参考文献〕

- [1] R. Kowalski: "Predicate logic as programming language", IFIP-74, p569-574, 1974.
- [2] S. Å. Tärnlund: "Horn clause computability", BIT, 17, 12, p25-226, 1977.
- [3] J. Sebelik, P. Stepanek: "Horn clause programs suggested by recursive functions", Logic programming Workshop 14-16, 7, p348-359, 1980.
- [4] C. L. Chang, R. C. Lee: "Symbolic Logic and Mechanical Theorem Proving", Academic Press, 1974.
- [5] L. Henshen: "Unit refutability and Horn sets", J. ACM, Oct, Vol 21, No 4, p596-605, 1974.
- [6] 山崎, 堂下: "入力導出を階層化した導出[2]", AL 79-89, 1979.
- [7] J. C. Reynolds: "Transformational systems and algebraic structure of atomic formula", Machine Intelligence, 5, p135-156, 1969.
- [8] Z. Manna: "Mathematical Theory of computation", McGraw-Hill, 1974.
- [9] H. R. Lewis: "Complexity of solvable cases of the decision problem for the predicate calculus", IEEE, 19th Annual Symp. on Foundation of Computer Science, p35-47, 1978.
- [10] 石橋, 山崎, 堂下: "プログラミング言語 PROLOG の解釈機構の複雑さ", AL 80-43, 1980.
- [11] N. D. Jones. et al.: "Complete problems for deterministic polynomial time", Theoretical Computer Science, 3, p105-117, 1977.
- [12] S. Yamasaki. et al.: "The unsatisfiability problems for the Ackermann class and the related properties of a class of flowchart schemas", 信学会 AL 資料, 1981.