数理解析研究所講究録
421巻 1981年 232-246

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、そこでは、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節は高々一

つしかポジィティブリテラルを含まないリテラルの論理和である。ポジィティブリテラルは、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、ポジィティブリテラルは、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、$P, N_1, N_2, \cdots, N_i$は原子論理式、□は空節を表わす。

(i) $+P-N_1-N_2\cdots-N_i$：手続き宣言を意味する。$+P$は手続き名、$-N_1-N_2\cdots-N_i$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) $-N_1-N_2\cdots-N_i$：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) □：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 $A \leftarrow B_1 \wedge B_2 \wedge \cdots \wedge B_m$ において、A

を手続き名，各B_iを手続き呼び出し、そして、包含を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、$B_i \leftarrow C_{i1} \wedge C_{i2} \wedge \cdots \wedge C_{ik} \wedge \cdots \wedge C_{im}$なる手続き宣言の実行が、必要である。この過程は、$A \leftarrow B_1 \wedge B_2 \cdots \wedge B_i \wedge \cdots \wedge B_m$より、$(A \leftarrow B_1 \wedge B_2 \wedge \cdots \wedge C_{i1} \wedge C_{i2} \wedge \cdots \wedge C_{im} \wedge \cdots \wedge B_m) \cdot \theta$（$\theta$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_1と導出を行なう際のC_1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Σ_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$p_i^n(i\geq 1, n\geq 0)$。p_i^0は命題定数と呼ぶ。
2. 変数：入力変数$\{x_1, x_2, \cdots\}$、プログラム変数$\{y_1, y_2, \cdots\}$、出力変数z、関数変数$\{f_1, f_2, \cdots\}$

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

数変数$\{f_1,\cdots,f_\ell\}$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_j, P_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_j, 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、if π then τ_1 else τ_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、$g_j(\tau_1,\cdots,\tau_n)$, $f_j(\tau_1,\cdots,\tau_n)$, if π then τ_1 else τ_2 をそれぞれ、<u>関数項</u>、<u>関数変数項</u>、<u>if項</u>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$g_j(\tau_1,\cdots,\tau_n)$があって、すべての$\tau_i$が変数又は定数であるとき、$g_j(\tau_1,\cdots,\tau_n)$は、基項である。■

[定義 3.4][8] リカーシブ図式Sは次のような式の集まりとする。

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

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

$$\vdots$$

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

$\tau_0 < \bar{f}, \bar{x} >$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$\mathbb{S}$の解釈$\mathbb{I}$は、次からなる。
1. 空でない集合$\mathbb{D}$。2. τ_iの定数の割当て：(a)関数記号g_i^nに$\mathbb{D}^n$から$\mathbb{D}$への全域関数を割当てる。個体定数には$a(\in\mathbb{D})$を割当てる。(b)述語記号p_i^nに$\mathbb{D}^n$から{真,偽}への全域関数を割当てる。■

[定義3.6][8] 組$\mathbb{P}=\langle\mathbb{S},\mathbb{I}\rangle$, $\bar{x}=\bar{\xi}$に対する計算は次の1, 2によって得られる項の列$\alpha_0,\alpha_1,\cdots$である。1. α_0は、$\tau_0\langle\bar{F},\bar{\xi}\rangle$を簡約したもの。2. $\alpha_{i+1}(i\geq 0)$は、α_iに現われる最も左側の最も内側の$f_j(\bar{\xi},\bar{\tau})$を$\tau_j\langle\bar{F},\overline{\ },\bar{\tau}\rangle$で置き換え簡約して$\alpha_i$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. if 真 then A else B をAで if 偽 then A else B をBで置き換える。3. $f_i(\tau_1,\cdots\tau_n)$で各τ_iが$\mathbb{D}$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$\mathbb{D}$の元であるか、α_kがωを含むときである。α_kが$\mathbb{D}$の元ζであるとき$val\langle\mathbb{P},\bar{\xi}\rangle$は定義されたと言い、$\alpha_k=\zeta\neq\omega$に対し$\zeta=val\langle\mathbb{P},\bar{\xi}\rangle$と書く。$\alpha_k=\omega$のとき$val\langle\mathbb{P},\bar{\xi}\rangle$は未定義である。■

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

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

[手順3.1]

入力：リカーシブ図式$\mathbb{S}$

出力：Horn節集合 $\mathbb{HS}$

記法：F_i、Q_{τ_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。Ⓤで論理和を、又 ∪ で集合の和を表わす。

```
procedure HORNSET(S):
 begin HS ← φ;
  if τ0 が ω の場合 then HS ← {−W}
   else if τ0<f̄, x̄> が基項σの場合
      then HS ← {−Qτ0(x̄, z)} ∪ {+Qτ0(x̄, σ)}
      else begin H0 ← HORN(τ0<f̄, x̄>, z);
                 HS ← HS ∪ {first(H0)} ∪ second(H0)
           end;
  for 各 fi(x̄, ȳ) ∈ τi<f̄, x̄, ȳ> do
   if τi<f̄, x̄, ȳ> が ω の場合
     then HS ← HS ∪ {+Fi(x̄, ȳ, ui) −W}
     else begin
            Ii ← HORN(τi<f̄, x̄, ȳ>, ui);
            HS ← HS ∪ {+Fi(x̄, ȳ, ui) Ⓤ first(Ii)}
                                          ∪ second(Ii)
          end;
  return HS
 end
```

```
procedure HORN(τ, v) :
begin HS←φ;
    if τがωである then H←(−W, φ);
    if τが基項σである then H←(−Q_σ(σ, v), {+Q_σ(σ, σ)});
    if τが(関数変数項)f_j(τ_1, τ_2, … τ_n)である
      then begin
            HC←−F_j(z_1, …, z_n, v);
            for i←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 Hi←HORN(τ_i, z_i);
                           HC←HC ⓤ first(Hi);
                           HS←HS ∪ second(Hi):
                     end;
            H←(HC, HS)
            end;
    if τが(関数項)g_j(τ_1, τ_2, …, τ_n)である
      then begin
            HC←−Q_gj(z_1, …, z_n, v); HS_0←{+Q_gj(z_1, …, z_n, g_j(z_1, …, z_n))};
            for i←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 Hi←HORN(τ_i, z_i);
                          HC←HC ⓤ first(Hi);
                          HS←HS ∪ Second(Hi)
                    end;
            HS←HS ∪ HS_0;  H←(HC, HS)
           end;
    if τが(if項) if P_i(τ_p1, …, τ_pn) then τ_1 else τ_2である
      then begin
            HC←{−P_i(τ_p1, …, τ_pn, z_p) ⓤ −P_b(z_p, x̄, ȳ, v)};
```

```
for i←1 until 2 (i≠pi) do
 if τi が ω である then HSi←{+Pb(zp, x̄, ȳ, v) ⓘ -W}
  else if τi が基項である
      then HSi←{+Pb(zp, x̄, ȳ, τi)}
      else begin Hi←HORN(τi, v);
          HSi←{+Pb(ai, x̄, ȳ, v) ⓘ first(Hi)}
                              ∪ second(Hi)
          end;
 H←(HC, HS1∪HS2)
 end;
 return H
end
```

(注) $P_i(\tau_{p1},\cdots,\tau_{pn},z_p)\equiv(z_p=p_i(\tau_{p1},\cdots,\tau_{pn}))$とする。又 $a_1\equiv1$, $a_2\equiv0$ とする。■

「命題 3.1」 τが関数変数やωを含まないとき、H←HORN(τ, v)に対し、{first(H)}∪second(H)から、first(H)を含んで単位反ばく(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、$\{first(H)\}\cup second(H)=\{-Q_\sigma(\sigma,v), +Q_\sigma(\sigma,\sigma)\}$ よって単位反ばくがある。

2. $\tau=g_j(\tau_1,\cdots,\tau_n)$ のとき、各 $H_i\leftarrow HORN(\tau_i,v_i)$ について $\{first(H_i)\}\cup second(H_i)$ から $first(H_i)$ を含んで単位反ばく($first(H_i)$を頂点節として入力反ばく(2章))があると仮定する。H←HORN(τ, v)に対し、HSから、$first(H)=-Q_{g_j}(z_1,\cdots,z_n,v)\underset{i}{ⓤ} first(H_i)$を頂点節とする入力導出で、

$-Q_{gj}(\tau_1,\cdots,\tau_n,v)$ を導ける。しかも、$H\mathfrak{S} \ni +Q_{gj}(z_1,\cdots z_n, g_j(z_1,\cdots,z_n))$ だから、入力反ばくが存在する。

3. $\tau = if\ \mathcal{P}_i(\tau_{p1},\cdots,\tau_{pn})\ then\ \tau_1\ else\ \tau_2$ のとき、$H_i \Leftarrow HORN(\tau_i, v_i)$ $(i=1,2)$ について、$\{first(H_i)\} \cup second(H_i)$ から、$first(H_i)$ を含んで単位反ばく（$first(H_i)$ を頂点節として入力反ばく（2章））があると仮定すると、$H \Leftarrow HORN(\tau,v)$ について、$H\mathfrak{S}$ から $first(H) = -P_i(\tau_{p1},\cdots,\tau_{pn},z_p)-P_b(z_p,\bar{x},\bar{y},v)$ を頂点節として、$first(H_1)$ or $first(H_2)$ 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$\mathfrak{S}$が停止することと$H\mathfrak{S}$から単位反ばくが存在することと等価である。

証明）

(1) $z \Leftarrow \sigma$（基項）のときは、命題が成立する。

(2) $\mathfrak{S}: z \Leftarrow \tau_0\langle f_1,f_2,\cdots,f_n,\bar{x}\rangle$ where $\{f_i \Leftarrow \tau_i\}_{i=1,\cdots,n}$ に対し、$\mathfrak{S}$ が停止する $\Leftrightarrow$ $H\mathfrak{S}$ から単位反ばくがあると仮定する。

$\mathfrak{S}: z \Leftarrow \tau_0\langle f_1,f_2,\cdots,f_n,f_{n+1},\bar{x}\rangle$ where $\{f_i \Leftarrow \tau_i\}_{i=1,\cdots,n+1}$ とする。
$\mathfrak{S}$ は停止する $\Leftrightarrow$ $(\exists i)\ f_i \Leftarrow \tau_i$（$\tau_i$ は等価的に関数変数や ω を含まない）について、$\mathfrak{S}_1: z \Leftarrow \tau_0\langle f_1,f_2,\cdots,f_{i-1},\tau_i,f_{i+1},\cdots,f_{n+1},\bar{x}\rangle$ where $\{f_i \Leftarrow \tau_i\}_{i=1,\cdots,n+1}$ が停止する。

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

second(I_i)からの単位反ばくがあるので、$\mathbb{HS}$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。$(\exists i)\, f_i \Leftarrow \tau_i$（$\tau_i$は等価的に関数変数や$\omega$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、$\mathbb{S}_1$に対して、手順3.1によって構成される節集合を$\mathbb{HS}_1$とすると、$\mathbb{S}_1$が停止する$\Leftrightarrow \mathbb{HS}_1$からの単位反ばくが存在する。$\Leftrightarrow \mathbb{HS}$からの単位反ばくが存在する。

$\mathbb{S}: z \Leftarrow \tau_0 < f_1, f_2, \cdots, f_{n+1}, \bar{x} >$ where $\{f_i \Leftarrow \tau_i\}_{i=1,\cdots,n+1}$に対して、すべての$\tau_i$が関数変数をもつなり$\omega$を含むときは、$\mathbb{S}$は停止せず、$\mathbb{HS}$は、$\{+F_i(\bar{x}, \bar{y}, u_i) \circledcirc first(I_i)\}$（$I_i \Leftarrow HORN(\tau_i, u_i)$）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$\mathbb{S}$を手順3.1によって、$\mathbb{HS}$に変形する。

$$\mathbb{S}: \quad z \Leftarrow f(x_1, x_2) \text{ where}$$
$$f(x, y) \Leftarrow \text{if } p(x) \text{ then } b \text{ else } f(g(x), f(x, y))$$

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

$\mathbb{HS}$から反ばくがなく、$\mathbb{S}$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引数関数のみを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、純粋第一階述語論理において冠頭標準形の前置部が、∃*∀∃*(*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] [9]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引数関数をもち、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引数関数のみを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、$\exists^* \forall \forall \exists^*$ (*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引数関数のみをもち、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引数関数のみをもつ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. むすび

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、一方で、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]をリカーシブ図式に対し、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, p215-226, 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, No4, 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 casses of the decision problem for the predicate calculas", 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", AL80-43, 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.