エルブランの定理にもとずく1階述語論理の論理式の妥当性検証プログラムの1つについて

大芝　猛・永田周郎・宮脇　誠*・舟橋　栄

共通講座教室

（1980年9月3日受理）

A Program for Checking the Validities of Formulas in the First Order Predicate Calculus.

Takeshi Oshiba, Shuro Nagata, Makoto Miyawaki* and Sakae Funahashi

Department of Engineering Sciences

(Received September 3, 1980)

In the first order predicate calculus, as well known, most of the procedures for checking the validity of an arbitrary formula, were supported by Herbrand's theorem. J.A. Robinson has given one of those procedures named as "Resolution Principle".

In this paper, we present an another method for checking the validities of formulas by using Herbrand's theorem and the following proposition:

"There exists an algorithm to determine for an arbitrary open formula $B(y_1, \cdots, y_k)$ whether there are such terms τ_i that B $(\tau_1, \cdots, \tau_k)$ is unsatisfiable, or not."

A program which is an implementation for the above method, is also reported.

0. 概要

1階述語論理の論理式の妥当性を検証する手続きは，エルブランの定理によって保証され，これにもとずく効率的な手続きとしてロビンソンの分解証明法がある。本報告では，ロビンソンの方法とは別に同定理にもとずき更に次の Proposition を援用し，他の1つの妥当性検証手続きを提示した。

『「任意の開論理式 $B(y_1, \cdots, y_M)$ に対し，$B(\tau_1, \cdots, \tau_M)$ を充足不能にする terms τ_i (B のエルブラン領域の項) が存在するか？」を判定するアルゴリズムが存在し，プログラムしやすい形で与えられる。』

また制限付ではあるが FACOM-U-200 により具体的にプログラムを作成し，種々の論理式に適用し，結果を得たので併せて報告する。

1. 妥当性検証手続

エルブランの定理は次の形で述べることができる。

『1階の述語論理の論理式 A に対して，次のような開論理式 $D(x_1, \cdots, x_n)$ を与えうる。

$\vdash A \rightleftarrows \neg A$ は充足不能

$\rightleftarrows \forall x_1 \cdots \forall x_n D(x_1, \cdots, x_n)$ は充足不能

$\rightleftarrows D(\tau_{11}, \cdots, \tau_{1n}) \wedge \cdots \wedge D(\tau_{m1}, \cdots, \tau_{mn})$ はある $\tau_{ij} \in U(D(x_1, \cdots, x_n))$ と $m \geqslant 1$ に対し充足不能』

〔但し$\vdash$は1階述語論理の演繹体系（(LK 等）で証明可能（即ち妥当性と同等）を意味する。また $U(D(x_1, \cdots, x_n))$ は $D(x_1, \cdots, x_n)$ に対するエルブラン領域，即ち D 内の個体記号（なければ a を1つ選ぶ）と関数記号とを用いて作り得るすべての項の集合〕

従って妥当性 $\vdash A$ の検証手続きは Fig. 1 のように表わされる。ここで判定条件＊<□>の検証について m を fix しても $D(x_{11}, \cdots, x_{1n}) \wedge \cdots \wedge D(x_{m1}, \cdots, x_{mn})$ の x_{ij} に代入される項 τ_{ij} の組み合わは，一般には $(D(x_1, \cdots, x_n)$ が関数記号を含む故) 無限個の可能性を持つ。それにもかかわらず条件＊の判定は有限ステップで可能であって，このアルゴリズムをプログラムしやすい□内の形で実現することにより $\vdash A$ の検証手続きは Fig. 2 のようになる。

* 沖電気工業株式会社 Oki Electric Industry Co.

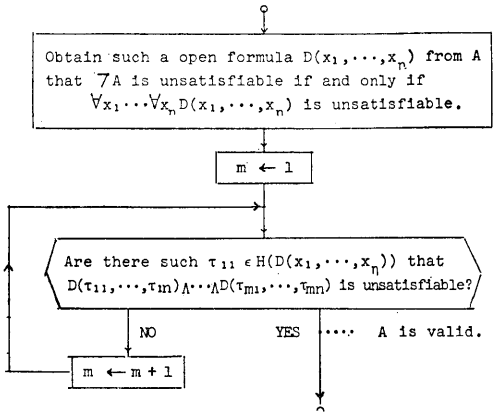
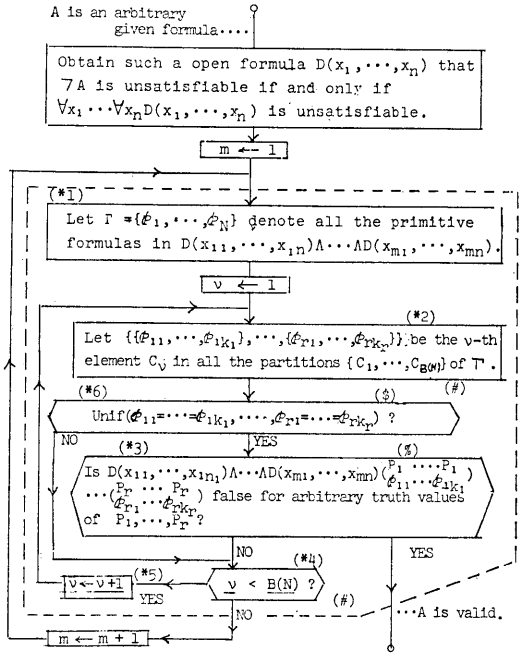


Fig. 1 Procedure for checking the validities of formulas, based on Herbrand's theorem.



(#) B(N) denotes the N-th Bell number($=\sum_{k_0=0}^{N-1}\binom{N-1}{k_0}1^{k_0}\cdot 2^{k_1}\cdot\cdots\cdot(r+1)^{k_r}$).

($) For a system Π of equations on " = ", Unif(Π) means that Π has a solution of terms in H(D(x1,···,xn)), where " = " means the coincidence relation between two figures.

(%) B(P1 ··· Pi / ℘i1 ··· ℘iki) denotes the substitution which replaces ℘i1, ···, ℘iki in B with a propositional variable Pi.

Fig. 2 An implementation of the procedure illustrated in Fig. 1, by using an enumeration of the partitions of an arbitrary finite set.

2. 提起された手続きの正当性

Fig. 1 の判定条件＊を Fig. 2 の [ ] 内のアルゴリズムとして実現しうることは，「有限集合の1つを定めたとき，その分割の全体を数え終える手続き（後述の Fig. 3）が存在する」ことと次の結果により保証される。即ち，アルゴリズムの実現には $E(y_1,\cdots,y_M)$ を $D(x_{11},\cdots,x_{1n})\wedge\cdots\wedge D(x_{m1},\cdots,x_{mn})$ とし，$\tau_1,\cdots,\tau_M$ を $\tau_{11},\cdots,\tau_{ij},\cdots,\tau_{mn}$ とみなして Proposition を適用すればよい。

〔Proposition〕 $E(y_1,\cdots,y_M)$ が開論理式のとき，'$E(\tau_1,\cdots,\tau_M)$ が充足不能となる $\tau_i\in U(E(y_1,\cdots,y_M))$ が存在する。'

$\rightleftarrows$ '$E(y_1,\cdots,y_M)$ の中の原始論理式の全体 $\Gamma=\{p_1,\cdots,p_N\}$ のある分割 $C=\{\{p_{11},\cdots,p_{1k_1}\},\cdots,\{p_{r1},\cdots,p_{rk_r}\}\}$ に対して

① $p_{11}\alpha=\cdots=p_{1k_1}\alpha,\cdots,\ p_{r1}\alpha=\cdots=p_{rk_r}\alpha$ を成立させる統一置換（代入）α が存在し，かつ

② $E(y_1,\cdots,y_M)\begin{pmatrix}P_1\cdots P_1\\ p_{11}\cdots p_{1k_1}\end{pmatrix}\cdots\begin{pmatrix}P_r\cdots P_r\\ p_{r1}\cdots p_{rk_r}\end{pmatrix}$ は恒偽となる。'

（但し各 P_i は命題変数）

（証明）"→part"：$E(y_1,\cdots,y_M)$ の代入 $\alpha=\begin{pmatrix}\tau_1\cdots\tau_M\\ y_1\cdots y_M\end{pmatrix}$ を行った結果の $E(\tau_1,\cdots,\tau_M)$ が充足不能。一方 $E(y_1,\cdots,y_M)$ の原始論理式全体 $\Gamma=\{p_1,\cdots,p_N\}$ に対し，上記 α により同じ形になるものを同値類別し導入した分割を $C_\alpha=\{\{p_{11}\cdots p_{1k_1}\}\cdots\{p_{r1}\cdots p_{rk_r}\}\}$ とする。従って $p_{11}\alpha=\cdots=p_{1k_1}\alpha=R_1,\cdots,p_{r1}\alpha=\cdots p_{rk_r}\alpha=R_r$ とおける。また $E(y_1,\cdots y_M)=H(p_{11},\cdots,p_{1k_1},\cdots,p_{r1},\cdots,p_{rk_r})$ とかける。従って充足不能な $E(\tau_1,\cdots,\tau_M)=E(y_1,\cdots,y_M)\alpha$ は $=H(p_{11}\alpha,\cdots p_{1k_1}\alpha,\cdots,p_{r1}\alpha,\cdots p_{rk_r}\alpha)=H(R_1,\cdots,R_1,\cdots,R_r,\cdots R_r)$ とかける。

従って原始論理式 R_i を命題変数 P_i で置き換えてうる $H(P_1,\cdots,P_1,\cdots,P_r,\cdots,P_r)$ は恒偽である。即ち

$=H(p_{11},\cdots,p_{1k_1},\cdots,p_{r1},\cdots,p_{rk_r})\begin{pmatrix}P_1\cdots P_1\\ p_1\cdots p_1\end{pmatrix}\cdots\begin{pmatrix}P_r\cdots P_r\\ p_{r1}\cdots p_{rk_r}\end{pmatrix}$

$=E(y_1,\cdots,y_M)\begin{pmatrix}P_1\cdots P_1\\ p_{11}\cdots p_{1k_1}\end{pmatrix}\cdots\begin{pmatrix}P_r\cdots P_r\\ p_{r1}\cdots p_{rk_r}\end{pmatrix}$ は恒偽。

"←part"：E の原始論理式全体が $p_{11},\cdots p_{rk_r}$ で，②より

$E(y_1,\cdots,y_M)\begin{pmatrix}P_1\cdots P_1\\ p_{11}\cdots p_{1k_1}\end{pmatrix}\cdots\begin{pmatrix}P_r\cdots P_r\\ p_{r1}\cdots p_{rk_r}\end{pmatrix}$ が恒偽。

故に P_i に $p_{i1}\alpha$ を代入 $(i=1,\cdots,r)$ してうる式 $E(y_1,\cdots,y_M)\begin{pmatrix}p_{11}\alpha\cdots p_{11}\alpha\\ p_{11}\cdots p_{1k_1}\end{pmatrix}\cdots\begin{pmatrix}p_{r1}\alpha\cdots p_{r1}\alpha\\ p_{r1}\cdots p_{rk_r}\end{pmatrix}$ は充足不能，α として①を満す代入をとれば $p_{11}\alpha=p_{12}\alpha=\cdots=p_{1k_1}\alpha,\cdots,p_{r1}\alpha=\cdots=p_{rk_r}\alpha$ ゆえ充足不能な上記の式は

$E(y_1,\cdots,y_M)\begin{pmatrix}p_{11}\alpha\cdots p_{1k_1}\alpha\\ p_{11}\cdots p_{1k_1}\end{pmatrix}\cdots\begin{pmatrix}p_{r1}\alpha\cdots p_{rk_r}\alpha\\ p_{r1}\cdots p_{rk_r}\end{pmatrix}=E(y_1,\cdots,y_M)\cdot\alpha$

$=E(\tau_1,\cdots,\tau_M)$ 自体である。　　（証了）

上記 Proposition により「$E(\tau_1,\cdots,\tau_M)$ が充足不能となる $\tau_i\in U(E(y_1,\cdots,y_M))$ の存否を判定する」＊′には $\Gamma=\{p_1,\cdots,p_N\}$ の分割全体について逐一 "①と②" の成

立いかん？ を調べて行くことに帰着される。

①の「統一置換の存否判定のアルゴリズム」については，Robinson [2], Manna [3] 等にも記述されている。(後述のプログラム作成上独自の手続きを構成したが本質的には上記文献のものと同様であるため説明を省略する。)

②の成立判定は「命題論理の論理式の恒偽性判定」ゆえアルゴリズムである。

以上から Γ の分割全体を取り残しなくトレースし終わるプログラムさえ与えれば，Fig. 1 の ＊の判定と同等な Fig. 2 の [:::] 内の計算はアルゴリズムとして実現される。

3. 分割全体の数え上げプログラム

(1) 一般に N 個の要素からなる集合 $\Gamma=\{p_1,\cdots,p_N\}$ のすべての分割は有限個 ($B(N)$ で表わす) で，これらを取り残しなくカウントするため，次の条件 F を持つ N 進数 $\nu=n_0|n_1|\cdots|n_{N-1}$ を用いる。

$$F(\nu)\overset{\text{def}}{\rightleftarrows} n_0=0\wedge\forall_i(0<i\leqq N-1\rightarrow((\underset{j<i}{\text{Max}}n_j)+1\geqslant n_i))$$

このような N 進数全体 $F=\{\nu|F(\nu),\ \nu$ は N 進数$\}$ が $\Gamma=\{p_1,\cdots p_N\}$ の分割全体を特性づけうる。即ち，index $\nu=n_0|n_1|\cdots|n_{N-1}\in F$ に対する分割 C_ν とは「p_1, p_2, $\cdots$, p_N の属する類の番号を index 順にそれぞれ $n_0, n_1, \cdots n_{N-1}$ と指定した分割」とする。

(例えば $N=8$ のとき $C_{0|1|0|0|1|2|1|2}$ は index の各桁に Γ の要素 $p_1, p_2, \cdots p_8$ が対応し，結果として分割 $\{\{p_1,p_3,p_4\},\{p_2,p_5,p_7\},\{p_6,p_8\}\}$ を表わす。)

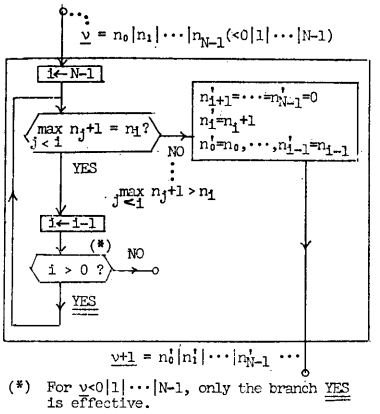
(2) Fの要素間の順序の定義は，$\nu=n_0|n_1|\cdots|n_{N-1}, \mu=m_0|m_1|\cdots|m_{N-1}$ に対し $\nu>\mu\overset{\text{def}}{\rightleftarrows}\exists k: 0\leqq k\leqq N-1$ ($n_0=m_0\wedge\cdots\wedge n_{k-1}=m_{k-1}\wedge n_k>m_k$) とする。…Fig. 2 (＊4)

Fの要素全体 ($B(N)$ 個) を小より大にかけて並べ，その ν ($1\leqq\nu\leqq N$) 番目の index を $\underline{\nu}$ と ν にアンダーラインを付けて表わす。($\underline{1}=0|0|\cdots|0$ であり $\underline{B(N)}=0|1|\cdots|N-1$ である。また $C_1=\{\{p_1,\cdots,p_N\}\}$, $C_{B(N)}=\{\{p_1\},\cdots,\{p_N\}\}$ である。)

(3) Fig. 2. (＊5) $\underline{\nu}\leftarrow\underline{\nu+1}$ 即ち ν 番の index $\underline{\nu}=n_0|n_1|\cdots|n_{N-1}(<B(N))$ が与えられたとき，$\nu+1$ 番目の index $\underline{\nu+1}=n_0'|n_1'|\cdots|n_{N-1}'$ を求める計算は Fig, 3 のフローチャートにより実現される。

4. 計算能率に関する手続の修正

以上説明した論理式の妥当性検証手続 (Fig. 2) につき計算の実際上から修正される点を以下に述べる。また後述の FACOM-U-200 のプログラム作成においては，



(*) For $\nu<0|1|\cdots|N-1$, only the branch YES is effective.

Fig. 3 A procedore for enumerating all the partitions of the set {1, 2, ..., N}.

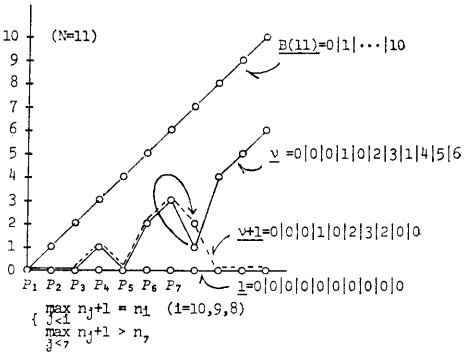


Fig. 4 An example of the relation between $\underline{\nu}$ and $\underline{\nu+1}$, (two successive indeces of partitions).

現在は未修正であるが，修正することによって能率が更に改良される事項についても触れておく。

(1) (＊1), (＊2) のブロックについて

$D(x_{11},\cdots,x_{1n})\wedge\cdots\wedge D(x_{m1},\cdots,x_{mn})$ の中の2つの原始論理式で $p_j=P(\tau_1,\cdots,\tau_p)$, $p_k=Q(\sigma_1,\cdots,\sigma_q)$ のように異なる述語記号で始るものは始めから統一不能 (変数に適当な項を代入しても同じ式にならない) であることは明白で，これらを同じ類に入れるような分割は検証の対象からはずし，(＊1) は次のように修正する。

(＊1′) $D(x_{11},\cdots,x_{1n})\wedge\cdots\wedge D(x_{m1},\cdots,x_{mn})$ の中の述語記号全体を $P_1,\cdots,P_d$ とし P_i で始まる原始論理式全体を $\Gamma_{p_i}=\{P_i(\sigma_{i1}),\cdots,P_i(\sigma_{ik_i})\}$ と indicateする ($i=1,\cdots,d$).

(従って原始論理式全体は $\Gamma=\Gamma_{p_1}\cup\cdots\cup\Gamma_{p_d}$ である) とする。更に (＊2) は次のように変える。

(＊2′) Γ の分割 $C=\{\Gamma_{r1},\cdots,\Gamma_{p_rl}\}$ の細分割全体 : $\{C_1,\cdots,C_L\}$ のうち ν 番目の分割 C_ν を $\{\{p_{11},\cdots,p_{1l_1}\},\cdots,\{p_{r1},\cdots,p_{rk_r}\}\}$ と indicate する

(2) 上記の修正によってもなお検証対象となる分割の数はまだかなり多い。一方ある分割 C_ν について (＊6)

の検査が肯定された上で，(*3) の検査が否定された場合，それ以後の分割の検査から $\mathbf{C}_{\nu}$ より細分された分割はオミットすることが可能である。(なぜなら，分割 $\mathbf{C}_{\nu}$ に対する (*3) 内の式が恒偽でないならば，$\mathbf{C}_{\nu}$ より細分された分割 $\mathbf{C}_{\mu}$ に対して (*3) 内の式も恒偽とはなり得ない。) このような修正を施せば検査対象の分割の数は相当減少し，能率は改良される。但し後述のプログラムについてはこの件は未修正である。

5. Example

以下1例によって，本稿での妥当性検証手続きの説明を行う。

Example: *V.W.O. Quine's paradox.*

$A=\neg\exists z\forall x[P(x,z)\equiv\neg\exists y(P(x,y)\wedge P(y,x))]$

(phase 1) $\neg A$ に対し充足不能性において同等な $\forall x_1\cdots\forall x_n D(x_1,\cdots,x_n)$ を求め次式を得る ($n=2$).

$\forall z\forall x[[\neg P(z,a)\vee\neg P(z,x)\vee\neg P(x,z)]\wedge[P(z,a)\vee P(f(z),z)]\wedge[P(z,a)\vee P(z,f(z))]]$

(phase 2)「$D(\tau_{11},\tau_{12})\wedge\cdots\wedge D(\tau_{m1},\tau_{m2})$ が充足不能となる τ_{ij} が存在するか？」の判定を $m=1,2,\cdots$ について行ない，肯定されるまで続ける。

($m=1$) のときは否定的に判定(説明略)，$m=2$ へ進む。

($m=2$) のときに次のように肯定的に判定され $\vdash A$ が結論される。

『$D(x_1,z_1)\wedge D(x_2,z_2)$ の原始論理式の全体は

$\Gamma=\{P(z_1,a),P(z_1,x_1),P(x_1,z_1),P(f(z_1),z_1),P(z_1,f(z_1))),P(z_2,a),P(z_2,x_2),P(x_2,z_2),P(f(z_2),z_2),P(z_2,f(z_2))\}=\{p_1,\cdots,p_{10}\}$

Γ の分割 $\mathbf{C}_{\nu}$ を $\mathbf{C}_1=\{\{p_1,\cdots,p_{10}\}\}$ から $\mathbf{C}_{B(10)}=\{\{p_1\},\cdots,\{p_{10}\}\}$ に向って

①：Fig. 2 の (*6)
②：Fig. 2 の (*3)

の検証を行ってゆけば，

$\mathbf{C}_{0|0|0|1|2|1|1|1|2|3|4}=\{\{p_1,p_2,p_3\},\ \{p_4,p_6,p_7\},\ \{p_5,p_8\},\{p_9\},\{p_{10}\}\}$ なる分割のときに

① $p_1\alpha=p_2\alpha=p_3\alpha$, $p_4\alpha=p_6\alpha=p_7\alpha$, $p_5\alpha=p_8\alpha$ をみたす統一置換 α の存在が肯定され

〔実際，$\alpha=\begin{pmatrix}a & a & a & f(a)\\ x_1 & z_1 & x_2 & z_2,\end{pmatrix}$，即ち $x_1=z_1=x_2=a$, $z_2=f(a)$〕

② $D(x_1,z_1)\wedge D(x_2,z_2)$ において上記分割の5つの類の論理式をそれぞれ $P_0,\cdots,P_4$ でおきかえてうる命題論理の論理式

$[\neg P_0\vee\neg P_0\vee\neg P_0]\wedge[P_0\vee P_1]\wedge[P_0\vee P_2]\wedge[\neg P_1\vee\neg P_1\vee\neg P_2]\wedge[P_1\vee P_3]\wedge[P_1\vee P_4]$

```
%GIVEN FORMULA
      -#Z@X(P(X,Z)=-#Y(P(X,Y)&P(Y,X)))

    ﾍﾝｽｳ ﾉ ﾅﾏｴ ｵ ﾂｹｶｴﾙ.
      --∃X0@X1(P(X1,X0)=-∃X2(P(X1,X2)&P(X2,X1)))

    ｸﾗﾊﾞｰ,ｸﾄﾞｳﾗ ｵ ｼﾞｮｷｮ.
      --∃X0@X1((-P(X1,X0)V-∃X2(P(X1,X2)&P(X2,X1)))&(--∃X2(P(X1,X2)&P(X2,X1))VP(X1,X0)))

    ¬ ｵ ｳﾗｶﾞﾜ ﾍ ｳﾂｽ.
      ∃X0@X1((-P(X1,X0)V@X2(-P(X1,X2)V-P(X2,X1)))&(∃X2(P(X1,X2)&P(X2,X1))VP(X1,X0)))

    ｿﾝｻﾞｲｷｺﾞｳ ｵ ｼﾞｮｷｮ,
      @X1((-P(X1,A)V@X2(-P(X1,X2)V-P(X2,X1)))&((P(X1,F1(X1))&P(F1(X1),X1))VP(X1,A)))

    ｾﾝｼｮｳｷｺﾞｳ ｵ ﾏｴ ﾆ,
      @X1@X2((-P(X1,A)V(-P(X1,X2)V-P(X2,X1)))&((P(X1,F1(X1))&P(F1(X1),X1))VP(X1,A)))

    V ﾆ ｶﾝｼﾃ & ｵ ﾌﾞﾝﾊﾟｲ.
      @X1@X2((-P(X1,A)V(-P(X1,X2)V-P(X2,X1)))&((P(X1,A)VP(F1(X1),X1))&(P(X1,A)VP(X1,F1(X1)))))

%NORMAL FORM
      -P(X1,A)*-P(X1,X2)*-P(X2,X1)/P(X1,A)*P(F1(X1),X1)/P(X1,A)*P(X1,F1(X1))
      -P0*-P0*-P0/P0*P1/P0*P2
      -P0*-P0*-P1/P0*P2/P0*P3
      -P0*-P1*-P0/P0*P2/P0*P3
      -P0*-P1*-P1/P0*P2/P0*P3
      -P0*-P1*-P2/P0*P2/P0*P1
      -P0*-P1*-P2/P0*P2/P0*P3
      -P0*-P1*-P2/P0*P3/P0*P1
      -P0*-P1*-P2/P0*P3/P0*P4
%END ALL CASES
                                                                  /P(Y1,A)*P(F1(Y1),Y1)
%NORMAL FORM
      -P(X1,A)*-P(X1,X2)*-P(X2,X1)/P(X1,A)*P(F1(X1),X1)/P(X1,A)*P(X1,F1(X1))/-P(Y1,A)*-P(Y1,Y2)*-P(Y2,Y1)
      /P(Y1,A)*P(Y1,F1(Y1))
      -P0*-P0*-P0/P0*P1/P0*P2/-P0*-P0*-P0/P0*P1/P0*P2
      -P0*-P0*-P0/P0*P1/P0*P2/-P0*-P0*-P0/P0*P1/P0*P3
                                  -P0*-P0*-P0/P0*P3/P0*P2

      -P0*-P0*-P0/P0*P1/P0*P2/-P0*-P3*-P4/P0*P5/P0*P6
      -P0*-P0*-P0/P0*P1/P0*P2/-P1*-P1*-P2/P1*P3/P1*P4

%ACCEPT
      -P0*-P0*-P0/P0*P1/P0*P2/-P1*-P1*-P2/P1*P3/P1*P4
```

Fig. 5 An application of Program CHK described in §5 and §6, to "$\neg\exists_x\forall_x(p(x,z)\equiv\neg\exists_y(p(x,y)\wedge p(y,x)))$."

は恒偽であることが肯定される。』

◎次節で説明する FACOM-U-200 による試作プログラムによれば表1のような結果が得られている。

また出力結果はFig. 5 の通りであり phase 2, $m=2$ において，Acctpt されている（妥当な分割について，対応する命題論理の論理式が恒偽でないかぎり，その分割のリストプリントを行っているが途中を切りとってある。

Table 1 Execution-times for the example in §5.

		Running time	
		Case when the listp-rinting is neglected	Case when the listp-rinting is performed
Phase 1		6.5 sec.	6.5 sec
Phase 2	m=1	4.	5.
	m=2	6.	14.
Total		16.5	25.5

Refer to Fig. 5.

6. FACOM-U-200 によるプログラムと適用例

前記妥当性検証手続きを表題の機種のアセンブラ言語を用いて制限付きであるがプログラムを作り，記号論理学の教科書にもある typical な式，太田〔6〕，Ono〔7〕によるラッセルタイプのパラドックスのいくつか，更に前掲のManna〔3〕にのっている諸例に適用したのでその結果を付記する。

ミニコンといわれる機種の（主として）メモリ量の制約等から先づ第1段階として次の条件のもとで，プログラムを作成した。

その際，記号の種類，標準形に展開した時の式の長さ等はあまり自由にせず，前述のような通常よく現れる式については，検証プロセスに入れるよう考慮した。

これらの制限は本質的でなく，制限をゆるめ，あるいは条件を拡張することは比較的容易である。

述語記号は *phase* 1 で6種 J,K,L,M,N,O *phase* 2 で3種 P,Q,R

関数記号は4種 F,G,H,I

変数は10種

定数は10種

○述語記号，関数記号の arguments の数は高々2個。

○*phase* 1 で節形式標準形に変形するとき，リスト構造として展開するためのメモリは1000ワード（2000バイト）。

○*phase* 2 における処理対象の式 $D(x_{11},\cdots,x_{1n})\wedge\cdots\wedge D(x_{m1},\cdots,x_{mn})$ の中の原始論理式の個数は最大16個。

以上の制限のもとでアセンブリ語でカード約2400枚，コアの占有量約12,000バイト（ワーキングエリアを含む）として，第1次のプログラムをまとめた。

なお4 (2) に述べた改良点についてはこの第1プログラムでは未修正である。

以下の実行諸例における時間は1論理式ごとにカード入力を行って検証プロセスに入り，受理されるまでの時分である。但し *phase* 2 におけるリスト（unifiable な分割のみ列挙する）のプリントアウトを押えた場合の時間である。また“?”マークの例は時間的に長くなり処理を打切ったもの，“⊗”印は変形プロセス等で式の長さ原始論理式の個数等が制約をこえたため停止したものを表わす。

○ $[\forall x\exists y P(x,y)\wedge\forall y\exists z Q(y,z)]\supset\forall x\exists y\exists z(P(x,y)\wedge Q(y,z))$　$m=1$, ACCEPT 10秒

○ $\exists x\exists y\forall z[[P(x,y)\supset(P(y,z)\wedge P(z,z))]\wedge[(P(x,y)\wedge Q(x,y))\supset(Q(x,z)\wedge Q(z,z))]]$ 妥当なるも$m=3$で途中打切　?秒

○ $\forall x\forall v\forall y[(P(x)\supset Q(x,y))\supset\forall z[(Q(v,y)\supset R(z))\supset(P(x)\supset R(z))]]$　$m=1$, ACCEPT 7秒

○ $\exists x\exists y[P(x,y)\wedge\forall v\neg Q(x,v)]\vee\forall s\forall t\forall z[\neg P(s,t)\vee Q(t,z)\vee(Q(s,z)\wedge\neg P(t,z))]$　$m=2$ ACCEPT39秒

○ $[\forall x\forall y(P(x,y)\supset P(y,x))\wedge\forall x\forall y\forall z[(P(x,y)\wedge P(y,z))\supset P(x,z)]\wedge\forall x\exists y P(x,y)]\supset\forall x P(x,x)$　$m=1$, ACCEPT 37秒

○ $(\forall x Q(x,x)\wedge\forall x\neg P(x,x))\supset\neg\exists z\forall x[P(x,z)\equiv\neg\forall v(Q(x,v)\supset P(v,v))]$　$m=1$, ACCEPT 9秒

○ $\neg\exists z\forall x[P(x,z)\equiv[\neg\forall s\forall v(P(s,s)\vee(P(s,s)\supset P(v,v)))\vee\neg P(x,x)]]$　$m=2$, ACCEPT 12秒

ラッセルタイプパラドックスの例

○ $\neg\exists z\forall x[P(x,z)\equiv\neg\exists y(P(x,y)\wedge P(y,x))]$　$m=2$, ACCEPT 17秒

○ $\neg\exists z\forall x[P(x,z)\equiv\neg\exists y\exists v(P(x,y)\wedge P(y,v)\wedge P(v,x))]$　$m=2$, ACCEPT 668秒

○ $\neg\exists z\forall x[P(x,z)\equiv\neg\exists y\exists v\exists s(P(x,y)\wedge P(y,v)\wedge P(v,s)\wedge P(s,x))]$　(妥当)　⊗

○ $[\forall y[\forall x(P(x,y)\supset R(x))\supset R(y)]\wedge\forall x(P(x,x)\supset\neg R(x))]\supset\neg\exists z\ \forall x(P(x,z)\equiv R(z))$　$m=2$, ACCEPT 64秒

○ $[[\forall x(\neg P(x,x)\supset R(x))\wedge\forall z[\forall x(R(x)\supset P(x,z))\supset\neg R(z)]]\supset\neg\exists z\forall x(P(x,z)\equiv R(x))$　$m=2$ ACCEPT 68秒

謝辞　本稿に関連して立教大の岩村聯教授，島内剛一

教授，筑波大の西村敏男教授，五十嵐滋教授，広島大の中村昭教授，東京理科大の細井勉教授，東海大の成島弘助教授には種々助言，資料等をいただいたことを感謝します。西村教授には特に今後の改良に関連しての示唆をうけました。

文献

[1] J.R. Shoenfield: Mathematical Logic, (1967), Addison-Wesley.

[2] J.A. Robinson: A Machine-oriented Logic Based on the Resolution Principle, (1965), J. ACM 12 (1).

[3] Z. Manna: Mathematical Theory of Computation, (1974), McGraw-Hill.

[4] 島内剛一：証明のプログラミング，(1963)，数学15巻，p. 48-55.

[5] 西村敏男：定理の証明，(1970)，情報処理 Vol. 11, No. 11, p. 646-651

[6] 太田稔：Russell-type paradox について，数学基礎論シンポジウム，(1965)，p. 12-23.

[7] K. Ono: On Russell-type Paradoxes and Some Related Problems, Lecture Notes on Axiomatic Set Theory, UCLA (1967) IV-G.1-7.