付随式を用いる1階述語論理の妥当性検証手続

大芝 猛・永田周郎・舟橋 栄
共通講座教室
(1984年9月8日受理)

A Procedure for Checking Validities of Formulas by Using Adjoint Formulas

Takeshi OSHIBA, Shuro NAGATA and Sakae FUNAHASHI
Department of Engineering Sciences
(Received September 8. 1984)

In general, any affirmative validity checking procedure by Herbrand's method consists of the following steps.

Firstly, the procedure constructs a certain open formula $\mathring{A}(X_1, \cdots, X_n)$ called the matrix of a given folmula A, and next, it searches a positive integer m and terms $\tau_{11}, \cdots, \tau_{1n}, \cdots, \tau_{m1}, \cdots, \tau_{mn}$ such that $\mathring{A}(\tau_{11}, \cdots, \tau_{1n}) \vee \cdots \vee \mathring{A}(\tau_{m1}, \cdots, \tau_{mn})$ is a tautology. Then the procedure terminates if and only if A is valid.

In this paper, such a validity checking procedure is implemented by using the matrix $\mathring{A}(X_1, \cdots, X_n)$ obtained from the adjoint formula $\tilde{A}\langle X_1, \cdots, X_n\rangle$ of A, through a certain substitution process, where the adjoint formula $\tilde{A}\langle X_1, \cdots, X_n\rangle$ is a certain scheme obtained from A by adding certain term-form indeces at all the quantifiers in A.

Then, such a usage of the adjoint formula of A, makes able to connect an LK-proof drawing algorithm to the above validity checking procedure.

In our impementation, a unification algorithm is also presented as a method for obtaining the general solution for any system of schema-equations : $\tau_1 = \sigma_1, \cdots, \tau_k = \sigma_k$.

0. 概　　要

分解原理で通常用いられる手続は与えられた論理式Aの妥当性検証に対し，否定A（¬A）の充足不能性を調べることとし，しかも¬Aの冠頭標準形から対応する節形式の集合を構成し，それらの項代入形である論理式から矛盾を導びくことを追求する方法によっている。

このような背理法的（refutable）な方法に対し，ここでは肯定的（affirmative）な方法により論理式Aの妥当性検証を行う手続きの1つについて述べる。

その手続の骨子は冠頭標準形への変形を行うことなく，与えられた論理式Aの付随式$\tilde{A}\langle X_1, \cdots, X_n\rangle$なる概念を導入し，次に冠頭標準形の場合のmatrixの拡張ともいえる開論理式$\mathring{A}(X_1, \cdots, X_n)$を対応させ，その項代入形の選言形$\mathring{A}(\tau_{11}, \cdots, \tau_{1n}) \vee \cdots \vee \mathring{A}(\tau_{m1}, \cdots, \tau_{mn})$がトートロジーとなるmと項$\tau_{ij}$とを探すことによって$A$の妥当性の検証を行うという方法によっている。なおこの方法は付随式$\tilde{A}\langle X_1, \cdots, X_n\rangle$が$A$内のquantifierを含めてAの論理式としての構造をそっくり保有していることから，上記妥当性検証手続が肯定的に終了したとき得られるm($\geqq 1$)と項τ_{ij}なる情報を付した形式$\tilde{A}\langle\tau_{11}, \cdots, \tau_{1n}\rangle \vee \cdots \vee \tilde{A}\langle\tau_{m1}, \cdots, \tau_{mn}\rangle$からAの証明図自身を決定論的に試行錯誤なく，下から上へと書き上げることを可能にするというもう一つの特色をもっているものである。即ち，$\tilde{A}\langle\tau_{11}, \cdots, \tau_{1n}\rangle \vee \cdots \vee \tilde{A}\langle\tau_{m1}, \cdots, \tau_{mn}\rangle$自身が$A$の証明図のもつ情報をすべて圧縮してもっている形式となるのである。この後者の証明図作成アルゴリズムについては〔2〕,〔3〕にその詳細が述べられている。

本稿では以下に前者の肯定的妥当性検証手続を具体化するProcedure CHKの内容の概要を述べる。

1. 付随式と対応する開論理式の定義

(1) A内のpositiveな$\exists$'sとnegative な$\forall$'sとを左からすべて列挙し$Q_1x_1, \cdots, Q_nx_n$であったとする。これらを$Q_1x_1(X_1), \cdots, Q_nx_n(X_n)$なるindex ($X_i$) 付のquantifierにおきかえてうる式を$A'$とする。

次にA'内のpositiveな$\forall$'sとnegativeな$\exists$'sを左からすべて列挙し$R_1y_1, \cdots, R_py_p$であったとき，これらをindex付の$R_1y_1[F_1], \cdots, R_py_p[F_p]$でおきかえてうる式を$A$の付随式と呼び$\tilde{A}\langle X_1, \cdots, X_n\rangle$と書く。但し

各 F_j (j=1, …, p) は $F_j=f_j(X_{i_1}, \cdots, X_{i_k})$なる項形式で，各 R_jy_j毎に次のように構成する：f_jは新しい関数記号であり，f_jの arguments は“A'の論理式としての構成順序からみて R_jy_jのあとからほどこされた$Q_ix_i(X_i)$ form の全体を左から $Q_{i_1}x_{i_1}(X_{i_1})$，…，$Q_{i_k}x_{i_k}(X_{i_k})$ とするとき，これらの indeces をとり出して得られる列X_{i_1}, …, X_{i_k}”である。

(example) $A=\forall y_1(\neg\forall x_1((\exists y_2 P(x_1, y_2))\vee P(x_1, y_1))\vee\exists x_2 P(y_1, x_2))$に対し，A の付随式は

$\tilde{A}\langle X_1, X_2\rangle = \forall y_1[f_1](\neg\forall x_1(X_1)((\exists y_2[f_2(X_1)]P(x_1, y_2))\vee P(x_1, y_1))\vee\exists x_2(X_2)P(y_1, x_2))$である。

(2) A に対応してうる開論理式（または拡張された意味での matrix）$\mathring{A}(X_1, \cdots, X_n)$とは付随式 $\tilde{A}\langle X_1, \cdots, X_n\rangle$内の $Qx_i(X_i)(\cdots x_i\cdots)$を$(\cdots X_i\cdots)$でおきかえ，$Ry_j[F_j](\cdots y_j\cdots)$を$(\cdots F_j\cdots)$でおきかえて得られる quantifier-free な（変数 X_1, …, X_nを unbound にもつ）式である。

即ち，一般に式 B 内の $Qx_i(T)(\cdots x_i\cdots)$を$(\cdots x_i\cdots)(\frac{x_i}{T})$なる代入とみなし，$Ry_j[F](\cdots y_j\cdots)$を$(\cdots y_j\cdots)(\frac{y_j}{F})$なる代入とみなしてうる式を sb($B$)とかくことにすれば，

$\mathring{A}(X_1, \cdots, X_n)=\mathrm{sb}(\tilde{A}\langle X_1, \cdots, X_n\rangle)$である。

(example)前例の A に対し，

$\mathring{A}(X_1, X_n)=\neg(P(X_1, f_2(X_1))\vee P(X_1, f_1))\vee P(f_1, X_2)$である。

更に項 τ_1, …, τ_nの代入に関して

$(\mathrm{sb}\tilde{A}\langle X_1, \cdots, X_n\rangle)\left(\begin{smallmatrix}X_1\cdots X_n\\ \tau_1\cdots\tau_n\end{smallmatrix}\right)=\mathrm{sb}(\tilde{A}\langle X_1, \cdots, X_n\rangle\left(\begin{smallmatrix}X_1\cdots X_n\\ \tau_1\cdots\tau_n\end{smallmatrix}\right))$即ち

$\mathring{A}(\tau_1, \cdots, \tau_n)=\mathrm{sb}(\tilde{A}\langle\tau_1, \cdots, \tau_n\rangle)$ が示される。

(example)前例について，

$\mathring{A}(\tau_1, \tau_2)=\neg(P(\tau_1, f_2(\tau_1))\vee P(\tau_1, f_1))\vee p(f_1, \tau_2)$,

$\tilde{A}\langle\tau_1, \tau_2\rangle=\forall y_1[f_1](\neg\forall X_1(\tau_1)((\exists y_2[f_2(\tau_1)]P(x_1, y_2)\vee P(x_1, y_1))\vee\exists x_2(\tau_2)P(y_1, x_2))$,

$\mathrm{sb}(\tilde{A}\langle\tau_1, \tau_2\rangle)=\neg(P(\tau_1, f_2(\tau_1))\vee P(\tau_1, f_1))\vee P(f_1, \tau_2)=\mathring{A}(\tau_1, \tau_2)$

2. 付随式による Herbrand の定理の表現と妥当性検証手続

前述の付随式または拡張された意味での matrix を用いれば，Herbrand の定理は次の形で表わされる。

〝A が LK で証明可能なことの必要十分条件は

$\mathring{A}(\tau_{11}, \cdots, \tau_{1n})\vee\cdots\vee\mathring{A}(\tau_{m1}, \cdots, \tau_{mn})$がトートロジーとなる $m\geqq 1$ と τ_{11}, …, $\tau_{mn}\in H(\tilde{A}\langle X_1, \cdots, X_n\rangle)$が存在することである。〟

但し，ここに $H(\tilde{A}\langle X_1, \cdots, X_n\rangle)$は $\tilde{A}\langle X_1, \cdots, X_n\rangle$ 内に現われる関数記号を用いて得られるあらゆる項の集合である。(もし $\tilde{A}\langle X_1, \cdots, X_n\rangle$ 内に “arguments 数 0 の関数記号” または定数記号がないときは，新しく定数記号 c_0を用意して用いるものとする。)

この形式の Herbrand の定理から，妥当性検証手続は直ちに次のフローチャート (Fig. 1) により表現される。

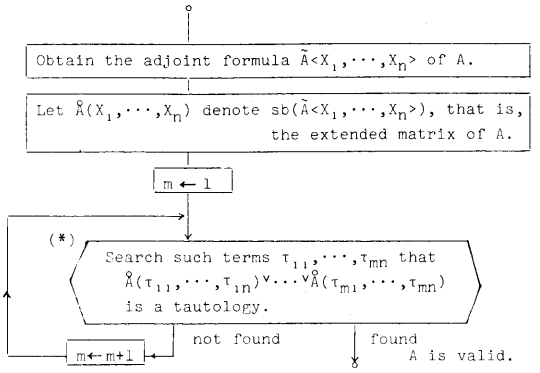

Fig. 1 An affirmative procedure CHK for checking the validities of formulas.

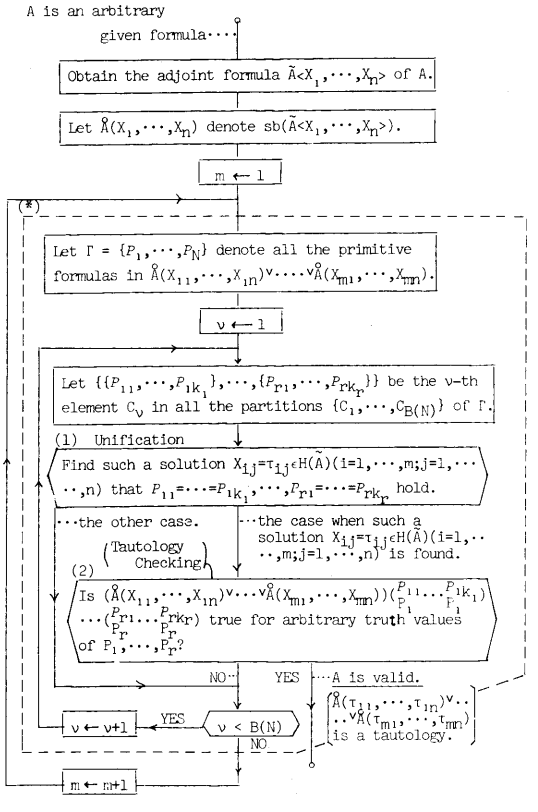

Fig. 2 An implementaion of the procedure CHK.

この妥当性検証手続は更に次の（Fig. 2）の形に具体化される。ここで Fig. 1 の m-fix に対する process ⬭ の部分は開論理式 $\mathring{A}(X_{11}, \cdots, X_{1n}) \vee \cdots \vee \mathring{A}(X_{m1}, \cdots, X_{mn})$ をトートロジーにする解 $X_{11}=\tau_{11}, \cdots, X_{mn}=\tau_{mn}$ を求める手続であり，この⬭の部分は次の Proposition により，アルゴリズムとして実現され，Fig. 2 では[---]で囲まれた部分に相当する。

(Proposition) 開論理式 $E(X_{11}, \cdots, X_{mn})[=\mathring{A}(X_{11}, \cdots, X_{1n}) \vee \cdots \vee \mathring{A}(X_{m1}, \cdots, X_{mn})]$ がトートロジーとなる $X_{ij}=\tau_{ij} \in H(E(X_{11}, \cdots, X_{mn}))$ が存在する。

$\rightleftarrows E(X_{11}, \cdots, X_{mn})$ 内の原始論理式の全体 $\Gamma=\{P_1, \cdots, P_N\}$ のある分割 $C_\nu=\{\{P_{11}, \cdots, P_{1k_1}\}, \cdots, \{P_{r1}, \cdots, P_{rk_r}\}\}$ に対して

(1) $P_{11}\theta=\cdots=P_{1k_1}\theta, \cdots, P_{r1}\theta=\cdots=P_{rk_r}\theta$ を成立させる最も一般な代入 $\theta=\left(\begin{smallmatrix}X_{11}\cdots X_{mn}\\ \tau_{11}\cdots\tau_{mn}\end{smallmatrix}\right)$ があり，かつ

(2) $E(X_{11}, \cdots, X_{mn})\left(\begin{smallmatrix}P_{11}\\ P_1\end{smallmatrix}\cdots\begin{smallmatrix}P_{1k_1}\\ P_1\end{smallmatrix}\right)\cdots\left(\begin{smallmatrix}P_{r1}\\ P_r\end{smallmatrix}\cdots\begin{smallmatrix}P_{rk_r}\\ P_r\end{smallmatrix}\right)$ はトートロジーとなる。

但し P_i は命題変数とする。

この Proposition により，$\mathring{A}(X_{11}, \cdots, X_{1n}) \vee \cdots \vee \mathring{A}(X_{m1}, \cdots, X_{mn})$ がトートロジーとなる $X_{11}=\tau_{11}, \cdots, X_{mn}=\tau_{mn}$ が存在するか否か，更に存在するときはその最も一般の形を求めるところの問題のアルゴリズム⬭は $\Gamma=\{P_1, \cdots, P_N\}$ のすべての分割 $C_\nu(\nu=1, \cdots, B(N))$ について，順次“条件(1)即ち，分割 C_ν の各クラス毎にその原始論理式を等しくする統一代入（unification）の存在の検証と条件(2)即ち分割で同一視される原始論理式を同じ命題変数でおきかえてうる論理式がトートロジーとなることの検証を交互に繰り返す”ことにより実現される。即ち

(Y) もし途中のある分割 $C_\nu=\{\{P_{11}, \cdots, P_{1k_1}\}, \cdots, \{P_{r1}, \cdots, P_{rk_r}\}\}$ で(1)が肯定され，かつその代入 $\theta=\left(\begin{smallmatrix}X_{11}\\ \tau_{11}\end{smallmatrix}\cdots\begin{smallmatrix}X_{mn}\\ \tau_{mn}\end{smallmatrix}\right)$ が得られ，その上で(2)が肯定されたときには，アルゴリズム⬭は肯定的に終了し θ が解を与える。

(N) もしすべての分割について“(1)もしくは(2)が否定される”ならばアルゴリズム⬭は否定的に終了することになる。

(1)，(2)のアルゴリズムの具体的設計について(2)のトートロジーの検証の具体化は自明であるが，(1)に対する unification algorithm としては次のような schema-equations の解法（Fig. 3）の形で与えることが出来る。

3 連立 Schema-equations の解法としての unification algorithm.

一般に，原始論理式の集り $\Gamma=\{P_1, \cdots, P_n\}$ の中に

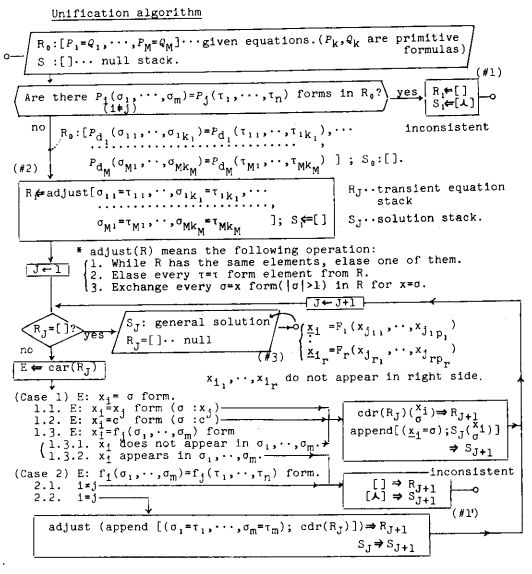

Fig. 3 A unification algorithm as a method for solving schema-equations : $P_1=Q_1, \cdots, P_M=Q_M$.

現われる変数を $X_1, \cdots, X_M$ とする。Γ の1つの分割 $C_\nu=\{\{P_{11}, \cdots, P_{1k1}\}, \cdots, \{P_{r1}, \cdots, P_{rk_r}\}\}$ の各組ごとの統一化を同時に成立させる最も一般的な代入 $\theta=\left(\begin{smallmatrix}X_1\cdots X_M\\ \tau_1\cdots\tau_M\end{smallmatrix}\right)$ を求めることは

$P_{11}=\cdots=P_{1k_1}, \cdots, P_{r1}=\cdots=P_{rk_r}$

なる schema-equation の解 $X_1=\tau_1, \cdots, X_M=\tau_M$ を求めることである。Fig. 3 はかかるアルゴリズムの1つである。

まず，もし $P_{ij}=P_{ij'}$ が $p_k(\sigma_1, \cdots, \sigma_m)=p_l(\tau_1, \cdots, \tau_n)$ $(k\neq l)$ なる形であれば勿論解は存在しない。(但し p_k, p_l は相異なる述語記号) ……Fig. 3(＃1)。

もし，すべての $P_{ij}=P_{ij'}$ が $p_k(\sigma_1, \cdots, \sigma_m)=p_k(\tau_1, \cdots, \tau_m)$ のときは，$\sigma_1=\tau_1, \cdots, \sigma_m=\tau_m$ なるような項の相等に関する連立方程式に整理される。…Fig. 3(＃2)。

この連立方程式系を出発点とし schema-equation としての同等変形により更に小さい項の相等式の集まりに分解して行くことになる。この変形のアルゴリズムは連立1次方程式の消去法と類似の方式で取扱うことが出来る。

即ち unknown としての変数 $X_1, \cdots, X_M$ の各 X_i ごとに，方程式系の中から1文字を残して消去する手続きを逐次行うことにする。そして(途中 $f_i(\sigma_1, \cdots, \sigma_m)=f_j(\tau_1, \cdots, \tau_n)$ のような矛盾式が現われ不能とならないかぎり)遂には一群の unknow $X_{i_1}, \cdots, X_{i_r}$ が他の一群の unknown を含む項で表わされることになる。即ち，不定解を表わす方程式系

$$\begin{cases} X_{i_1}=F_1(X_{j_{11}}, \cdots, X_{j_{1p_1}}) \\ \vdots \qquad\qquad \cdots \text{Fig. 3(\#3)} \\ X_{i_r}=F_r(X_{j_{r1}}, \cdots, X_{j_{rp_r}}) \end{cases}$$

に還元され，当初の方程式系の最も一般な解をうるのである。

4．手続の実際化にともなう留意点

〔1〕本稿による妥当性検証手続きは N 個の対象の集まりの分割のすべてをとり残しなくカウントすることによりアルゴリズム⬭を実行して行くが，このための1つの技法として，あるタイプのN-進数を用いることが可能である。即ち $\Gamma=\{P_1, \cdots, P_N\}$ に対して，その最もあらい分割 $C_1=\{\{P_1, \cdots, P_N\}\}$ には $(\overbrace{0, \cdots, 0}^{N})$ なるN-進数を対応させ，最も細かい分割 $C_{B(N)}=\{\{P_1\}, \cdots, \{P_N\}\}$ には $(0, 1, \cdots, N-1)$ なるN-進数を対応させる。

一般に N-進数 $(n_0, n_1, \cdots, n_{N-1})$ に対し P_k の属するクラス番号が n_{k-1}（k 桁目の数字）であると考えれば，1つのN-進数には1つの分割が一意に対応する。逆に分割の各々を唯1つの N-進数で表現するためには，次の条件をみたすN-進数のみをとればよい：

$$n_0=0 \wedge \forall_i (0<i \leqq N-1 \rightarrow ((\underset{j<i}{\text{Max.}}\ n_j)+1 \geqq n_i)).$$

また，N個の対象の集り Γ の分割のすべてはかかるN−進数のすべてに次の(*)ような自然な順序を与えておき，小から大へ逐一カウントを進めることにより，とり残しなくとり上げることが出来る。

（但しLISP等によりかかるN-進数をデータとして扱う場合は，カウントを進めるための"＋1"の操作，もしくは桁上りのたびに右端の最下位を指摘するのに要する時間は処理時分の合計に大きな影響を与える。従って，このようなN-進数のデータ設計には"左を下位に右を上位にとる"というような注意が必要である。）

〔2〕更にアルゴリズム⬭は一般にすべての $\Gamma=\{P_1, \cdots, P_N\}$ の分割を扱う必要はなく，ある分割 C_ν について(1) C_ν の各クラスの原始論理式を統一化する代入が存在するにかかわらず，(2)対応する論理式がトートロジーとならない場合には，もはや C_ν より細かい分割について，対応する論理式はトートロジーとなり得ないため，このような C_ν より細かいすべての分割についての検証はスキップすることが可能である。⬭の部分のアルゴリズムをこのように修飾することにより処理時分を短縮しうることはすでに文献〔1〕にも述べている。

〔3〕また一方⬭の処理に関して，N個の分割 $C_1, \cdots, C_{B(N)}$ の各々に対する前々節2にのべた"検証アルゴリズム(1)&(2)"をそれぞれ $P_1, \cdots, P_{B(N)}$ とすれば，ある分割 C_ν に対応する検証 P_ν はその他の分割 C_μ に対する検証 P_μ に全く独立に遂行可能である。そこで何らかの分散処理機能をもつ計算システムがあれば，これらの検証プロセス $P_1, \cdots, P_{B(N)}$ をいくつかに分散併行処理させることにより，処理時間を縮めることが出来よう。このような（仮想の）システムの実際化を前提とするとき，前項にのべたスキップしうる分割との関連から，どのように分散併行処理グループを編成するのが適切といえるか，等の問題も生じる。

文　献

〔1〕大芝猛・永田周郎・宮脇誠・舟橋栄：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，名古屋工業大学学報，第32巻，PP. 91−96，1980.

〔2〕T. OSHIBA: A Method for Obtaining Proof Figures of Valid Formulas in the First Order Predicate Calculus, Comm. Math. Univ. St. Pauli, Vol. 30, PP. 49−62, 1981.

〔3〕大芝猛・永田周郎・舟橋栄：付随式を用いる1階述語論理の証明図作成方法，京都大学数理解析研究所講究録，458（形式言語理論とオートマトン理論）PP. 30−39，1982.

(*) $\nu=(n_0, n_1, \cdots, n_{N-1})$, $\mu=(m_0, m_1, \cdots, m_{N-1})$ に対し，
$\nu<\mu \overset{def.}{\Leftrightarrow} \exists k \leqq N-1 (n_0=m_0 \wedge \cdots \wedge n_{k-1}=m_{k-1} \wedge n_k<m_k)$.