述語論理における証明手続について†

——Theorem Prover *TP-I*——

米　澤　明　憲††

Abstract

This paper treats the proof procedure of the first order predicate calculus. At first, a proof procedure (named *BPP*) of Gentzen's *LK* type logic system is shown to be complete. It is also shown that several cases of decision problems in the predicate calculus are solved by using the the completeness of *BPP*. Then an effcient decision procedure (named *DAIM*) for these cases is proposed. Finally Theorem Prover *TP-I* is described, which is the implementation of the combined procedure of *DAIM* and *MPP* (the modified procedure of *BPP*).

1. まえがき

一階述語論理における定理の証明の機械化は，人工知能 system において，人間の推論，検証，計画能力を代替する手段として極めて重要である[7]．また，program の理論的研究においても，program の正当性，停止性，同値性などに関する諸結果が，一階述語論理で記述され[5]，program の自動 debugging，自動作製などが，定理の証明によって可能となることが知られている[6]．

定理の証明手続の基本的なものは，現在まで幾つか提案されている．J. A. Robinson[8),9)] の *Resolution Principle* は，J. Herbrand[3)] が与えた基本定理に基づいて，論理における内挿手続 (Interpolation) を応用したものである．島内[10),11)] による証明手続は，G. Gentzen[2)]の論理体系 *LK* を変形した体系に基づくもので，命題論理を取り扱う場合には H. Wang[14)] の手続とほとんど同じであるが，quantifier を含んだ formula に関して alah variable という概念を導入して，代入すべき free variable の組合せ数の増大を軽減している．

本論文は，まず体系 *TS* 及び証明手続 *BPP* を述べ，次に *BPP* を調べることにより，述語論理における既に知られている幾つかの決定問題の可解性が容易に証明されることを示す．さらに与えられた formula が上記の可解な場合であるとき，その formula が証明可能であるか，証明不可能であるかを判定する効率の良い手続 *DAIM* を与える．最後に *BPP* と *DAIM* を組合せた手続を組み込んだ Theorem Prover *TP-I* の implementation について述べる．

(2 以下で，下線を施された用語はそこではじめて定義されることを示す．)

2. 論理体系 *TS*

以下に述べる論理体系 *TS*† は，G. Gentzen の論理体系 *LK* と同等であるが，その詳しい証明は筆者の文献[12)]を参照されたい．

体系 *TS*

I) Symbol を以下のように用いる．

free variable: a b c a_1 b_1 c_1 …

bound variable: x y z x_1 y_1 z_1 …

述語記号: P Q R P_1 Q_1 R_1 …

論理記号: $\neg$ $\wedge$ $\vee$ $\forall$ $\exists$

補助記号: ( ) , ——

II) formula を以下のように帰納的に定義する．

i)†† 述語記号の後に 0 個またはそれ以上の free variable を並べたものは formula である．

ii) A と B が formula のとき，$\neg(A)$，$(A)\vee(B)$ および $(A)\wedge(B)$ は formula である．

† On a proof procedure of the first order predciate calculus, by Akinori Yonezawa (Faculty of Enginerring, University of Tokyo)

†† 東京大学工学部計数工学科

† 体系 *TS* は島内[10)]における体系 S_3 とほぼ同じものである．

†† 3 において alah variable を含む formula に拡張される．

ⅲ) $A(a)$ を，free variable a を0個またはそれ以上含み且つ bound variable x を含まない formula とする．このとき，$A(x)$ を$A(a)$ の中にあらわれる全ての a を x で置き換えたものとすれば，$\forall x(A(x))$, $\exists x(A(x))$ は formula である．

ⅳ) 上のⅰ) ⅱ) ⅲ) によって構成されるものだけが formula である．(但し，論理記号の結合の強さは ¬, ∀, ∃, ∧, ∨ の順で，誤解の生じないかぎり，カッコは省略してよいものとする．)

述語記号の後に並んだ variable† をその formula の argument という．$\forall x$, $\exists x$ をそれぞれ x を bound vairable とする univesal quantifier, existential quantifier という．formula の構成の順序で最後に導入される論理記号を，最も外側の論理記号という．formula の構成要素となっている formula を sub-formula という．

Ⅲ) sequence とは0個またはそれ以上の formula をコンマで区切って並べたものである．

Ⅳ) 推論とは，S_1, S_2, S を sequence とするとき，Fig. 1 の推論規則 (inference schemata) に従った次のような図形である．

$$\frac{S_1}{S} \quad \text{または} \quad \frac{S_1 \quad S_2}{S}$$

Ⅴ) 証明図とは，1個以上の beginning sequence と呼ばれる Fig. 1 で示される特別な形の sequence からはじめて，sequence を下側につなぎ合せて推論を構成した図形で，循環のないいわゆる木構造をなす．木構造の根に相当する唯一の sequence を end sequence と呼ぶ．

Ⅵ) sequence S が証明可能とは，S を end sequence とする証明図が存在することである．存在し得ないとき証明不可能という．

(この体系では，それぞれ $\neg\neg A$ は A の，$\neg(A\vee B)$ は $\neg A\wedge\neg B$ の，$\neg(A\wedge B)$ は $\neg A\vee\neg B$ の，$\neg\forall x A(x)$ は $\exists x\neg A(x)$ の，$\neg\exists x A(x)$ は $\forall x\neg A(x)$ の省略形と考える．以下では，記述の便宜上，省略形は元の形に直されているものとする．また，ギリシャ文字の大文字は0個またはそれ以上の formula をコンマで区形った図形を表わす．)

○ Beginning sequence $\Gamma, A, \Delta, \neg A, \Lambda$

○ Inference schemata

Ⅰ) "∧"　$\dfrac{A,\Gamma,\Delta \quad B,\Gamma,\Delta}{\Gamma, A\wedge B,\Delta}$

Ⅱ) "∨"　$\dfrac{A,B,\Gamma,\Delta}{\Gamma, A\vee B,\Delta}$

Ⅲ) "∀"　$\dfrac{A(a),\Gamma,\Delta}{\Gamma,\forall x A(x),\Delta}$ (a does not appear in the lower sequence.)

Ⅳ) "∃"　$\dfrac{A(a),\Gamma,\Delta,\exists x A(x)}{\Gamma,\exists x A(x),\Delta}$ (a is an arbitrary free variable.)

Fig. 1 System TS'

† free variable, bound variable および alah variable の総称．

3. 証明手続 *BPP*

証明手続 *BPP*† は，与えられた sequence が体系 *TS* において証明可能か否かを調べるもので，以下 *BPP* の記述のためにⅠ)～Ⅳ) を仮定する．

Ⅰ) formula の argument として，free variable を代入するための場所を確保する目的で，alah variable という概念を導入する．alah variable は，ギリシャ文字の小文字 $\alpha, \beta, \gamma, \alpha_1, \beta_1, \gamma_1, \cdots$ で表わされ，2 の formula の定義 Ⅱ) ⅰ) で free variable の他に alah variable も並べてよいことにする．

Ⅱ) *BPP* で使われる free variable および alah variable は $a_1, a_2, a_3, \cdots$; $\alpha_1, \alpha_2, \alpha_3\cdots$ のように番号づけられ，この順に使われる．

Ⅲ) 使われた alah variable に対して，それに代入できる free variable を書き込む対照表が用意される．

Ⅳ) 与えられた sequence S_0 および途中で生成された sequence は順次 $S_0, S_1, S_2, \cdots$ のように番号づけられ，一列に並べられる．各 sequence は "active" または "inactive" のどちらかの状態を持ち，強制的に "inactive" にされるか，$\Gamma, A, \Delta, \neg A, \Lambda$ の形をしていればただちに "inactive" になるものとする．

〖証明手続 *BPP*〗

〔*Step*-1〕 $A\wedge B$ の形の formula (即ち最も外側の論理記号が ∨ である formula) を持つ "active sequence を番号順に探す．もしなければ，*Step*-2 へ．あれば，その sequence は $\Gamma, A\vee B, \Delta$ の形であるから，それを "inactive" にして，sequence の列の最後に，A, B, Γ, Δ をつけ加えて *Step*-1 へ行く．

〔*Step*-2〕 $\forall x A(x)$ の形の formulaを持つ "active" sequence を番号順に探す．もしなければ，*Step*-3 へ．あれば，その sequence は $\Gamma, \forall x A(x), \Delta$ の形であるから，それを "inactive" にして，sequence の列の最後に $A(a_n)$, Γ, Δ をつけ加えて *Step*-1 へ行く．但し，a_n はそれまでに使われていない番号が最小の free variable である．

† 証明手続 *BPP* は畠内[19]における体系 T_1 とほぼ同じものである．

〔*Step*-3〕 $A\wedge B$ の形の formula を持つ "active" sequence を番号順に探す．もしなければ，*Step*-4 へ．あれば，その sequence は $\Gamma, A\wedge B, \Delta$ の形であるから，それを "inactive" にして，sequence の列の最後に A, Γ, Δ および B, Γ, Δ をつけ加えて *Step*-1 へ行く．

〔*Step*-4〕 全ての "active" sequence について，その中にあらわれる全ての alah variable に，対照表に従って free variable を代入する．ただし，同じ alah variable には同じ free variable を代入する．また alah variable が1つもないときには，*Step*-5 へ行く．同じ alah variable に，1つ以上の代入可能な free vaiable があるから，上のような代入を free variable と alah variable の可能な全ての組合せについて繰り返す．その結果，全ての "active" sequence が同時に $\Gamma, A, \Delta, \rightarrow A, \Lambda$ の形になる代入が1つもなければ，*Step*-5 へ．あれば，S_0 は証明可能である．

〔*Step*-5〕 $\exists xA(x)$ の形の formula を持つ "active" sequence を番号順に探す．もしなければ，*Step*-6 へ．あれば，その sequence は $\Gamma, \exists xA(x), \Delta$ の形であるから，それを "inactive" にして sequence の列の最後に $A(\alpha_m), \Gamma, \Delta, \exists xA(x)$ をつけ加える．そして，$\Gamma, \exists xA(x), \Delta$ の中に free variable があれば，それら全てを，なければ特別な free variable a_0 を対照表の α_m の欄に書き込み，*Step*-1 へ行く．但し，α_m はそれまでに使われていない番号の最小の alah variable である．

〔*Step*-6〕 "active" sequence が残っているか調べる．残っていれば S_0 は証明不可能であり，なければ証明可能である．

(注意-1) 上の手続は，論理記号の間に $\vee, \forall, \wedge, \exists$ の順の優先順位を決めて，順位の高い最も外側の論理記号をもつ formula を含む "active" sequence が残っている間は，それより順位の低い論理記号に対する操作を行なっていない．

BPP に対して次の補題が成立するが証明は省略する．詳しい証明は筆者の文献12) を参照されたい．

〔**補題-I**〕 証明手続 *BPP* は完全性 (completeness) を持つ．すなわち *BPP* で証明可能と判定される sequence は，体系 *TS* において証明可能であり，かつその場合に限られる．

(注意-2) *BPP* は完全性を持つが，証明不可能な sequence に対しては，手続きが有限回で終ることを何ら保証していない．実際，$\exists x\forall y\rightarrow Pxy\vee\forall x\forall y Pxy$ (または，$\forall x\exists yPxy\rightarrow\forall x\exists yPxy$†) に *BPP* を適用すると，*Step*-5 によって生成される sequence には必ず $\exists x\forall y\rightarrow Pxy$ が含まれるので，*Step*-5 が無限に繰り返されることになる．すなわち，この *BPP* ではこのような簡単な場合でも，証明不可能であることが判定できないが，5 の *DAIM* は，この欠点を補うものである．

4. 決定問題の可解性の証明

BPP の完全性および *TS* と *LK* の同等性から，*BPP* によってあるクラスの sequence が証明可能であるか否かを決定できれば，そのクラスに対する決定問題の可解性が証明されたことになる．以下，*BPP* を用いて幾つかの決定問題の可解性についての別証明††を与える．

〔**定理-I**〕 次のような形をした formula に対する決定問題は可解である．

ⅰ) $\forall x_1\forall x_2\cdots\forall x_mA(x_1, x_2, \cdots, x_m)$

ⅱ) $\exists x_1\exists x_2\cdots\exists x_mA(x_1, x_2, \cdots, x_m)$

ⅲ) $\forall x_1\cdots\forall x_m\exists y_1\cdots\exists y_nA(x_1, \cdots, x_m, y_1, \cdots, y_n)$

ⅳ) $\exists x_1\cdots\exists x_m\forall y_1\cdots\forall y_nA(x_1, \cdots, x_m, y_1, \cdots, y_n)$

但し，$m, n\geq 0$，A は $\forall, \exists$ を含まない formula である．

(証明) ⅰ) の formula を S^0 として *BPP* を適用すると，*Step*-2 を m 回繰り返して $A(a_1, a_2, \cdots, a_m)$ が唯一の "active" sequence として残る．その後 *Step*-1 または *Step*-3 によって formula を分解するが，$A(a_1, \cdots, a_m)$ の中の論理記号の数は有限であるから *BPP* は必ず *Step*-6 で終る．

ⅱ) の formula はその否定をとれば ⅰ) の場合に帰着する．ⅲ) は *Step*-2 を m 回繰り返して $\exists y_1\exists y_2\cdots\exists y_nA(a_1, \cdots, a_m, y_1, \cdots, y_n)$ を唯一の "active" sequence として得，*Step*-5 を n 回繰り返して $A(a_0, \cdots, a_m, \alpha_1, \cdots, \alpha_n)$ を含む sequence を得て，さらにこれを *Step*-1 または *Step*-3 で分解したのち *Step*-5 を繰り返すことを続ける．しかし使われた alah variable は全て，それらに代入できる free variable が $a_1, \cdots, a_m$ および A にはじめから含まれていたものに限られる．ゆえ

† $\rightarrow$ は含意 (implication) を示す．

†† 3 の結果は既に知られている．Ackermann[1] を参照されたい．

に，*Step*-4 の alah variable への代入によって調べられる相異なる formula の数は有限である．iv）は否定をとれば，iii）の場合に帰着する．

以下，定理-II の証明に必要な定義，補題を述べる，

○prime formula とは，→ 以外の論理記号を持たない formula のことで，→ を持つときその符号を負，持たないとき符号を正とする．

○*C*-pair とは，符号が異なる以外は全て同一な2つの prime formula の組をいう．例えば $(Pabc, \rightarrow Pabc)$.

○sequence S の σ-列とは，S にあらわれる全ての free variable（もしなければ，特別の free variable a_0）の後に，S に *BPP* を適用して使われる free variable および alah variable を使われる順に一列に並べた列をいう．

○alah variable の間の順序関係 $\ll$ とは，alah variable をそれに代入できる free variable の集合と同一視したとき，ひとつの sequence S に *BPP* を適用したときに使われる alah variable の間に入る，集合の包含関係による順序関係のことで，$\alpha \subseteq \beta$ のとき $\alpha \ll \beta$ で表わす．

○σ-列が λ-property を持つとは，sequence S の σ-列 $\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \cdots\}$ が，任意の $i<j$（i, j は番号）に対して，$\varepsilon_i, \varepsilon_j$ がともに alah variable であるならば，$\varepsilon_i \ll \varepsilon_j$ をみたす場合をいう．

○sequence S の B-pair とは，sequence S に *BPP* を適用して得られる同一の sequence の中にあらわれる，2つの prime formula の組で，述語記号，argument の数は同一で，符号が異なるものをいう．例えば $(Pa\alpha\beta b, \rightarrow P\gamma b\alpha c)$.

○B-pair の clashability とは，sequence S の B-pair が，対照表で許される alah variable への代入によって C-pair となり得るか否かによって決まる値のことをいい，可能なとき1，不可能なとき0と定める．

○B-pair の添字列とは，sequence S の B-pair を $(P\theta_1 \cdots \theta_n, \rightarrow P\theta_{n+1} \cdots \theta_{2n})$ としたとき，各 $\theta_i (1 \leq i \leq 2n)$ に対して，番号 $h(\theta_i)$ を次のように定めたときの番号の列 $\{h(\theta_1), h(\theta_2), \cdots, h(\theta_{2n})\}$ をいう．まず各 θ_i が，S の σ-列にあらわれる順番を $\eta(\theta_i)$ で表わす．$\eta(\theta_i)$ を小さい順に並べた列を s とすると，$\eta(\theta_i)$ が s の j 番目にあらわれるとき $h(\theta_i)=j$ とする．

○branching とは *BPP* の *Step*-3 の適用をいう．

（注意-3） σ-列において，その中の任意の alah variable に，それより前に並んでいる全ての free variable を代入できるとき，その σ-列は λ-property を持つ．

〔**補題-II**〕 sequence S に *BPP* を適用して branching がなければ，S の δ-列は λ-property を持つ．

（証明） branching がないので，*BPP* の *Step*-5 において，S にはじめからある free variable およびそれまでに *Step*-2 で導入された free variable は全て，$\Gamma, \exists x A(x), \Delta$ の中にあらわれている．よって $A(\alpha_m)$, $\Gamma, \Delta, \mathrm{E}x A(x)$ の α_m には，それら全ての free variable が代入可能となる．これは注意-3 によって σ-列が λ-property を持つことを示している．

〔**補題-III**〕 sequence S の任意の2つの B-pair $(P\theta_1 \cdots \theta_n, \rightarrow P\theta_{n+1} \cdots \theta_{2n})$, $(P\pi_1 \cdots \pi_n, \rightarrow P\pi_{n+1} \cdots \pi_{2n})$ は，下の条件のもとで，各々の clashability は一致する．

i）S の σ-列は λ-property を持つ．

ii）θ_i と π_i はともに free variable であるか，またはともに alah variable かである．（但し $1 \leq i \leq 2n$.）

iii）2つの B-pair の添字列が同一である．

i）～iii）をみたす B-pair について，対照表で許される alah variable への代入を行なってみれば補題-III は明らかであろう，以下の定理に現われる $\mathfrak{O}_i\{x_1, \cdots, x_n\}$ は $x_1, \cdots x_n$ を bound vairable とする任意の quantifier の並びである．例えば $(\forall x_5 \forall x_3 \exists x_1 \forall x_1 \exists x_7, \exists x_1 \mathrm{E} x_2 \forall x_3)$，また，$P_i(x_1, \cdots, x_n)$, $Q_i(x_1, \cdots, x_n)$ は prime formula を表わす．但し $\mathfrak{O}_i, P_i, Q_i$ は，添字 i がなくてもよい．

〔**定理-II**〕 次のような形をした formula

i）$\mathfrak{O}\{x_1, \cdots, x_n\}(P(x_1, \cdots, x_n) \vee \mathfrak{O}(x_1, \cdots, x_n))$

ii）$\mathfrak{O}_1\{x_1, \cdots, x_m\}P(x_1, \cdots, x_n) \vee \mathfrak{O}_2\{y_1, \cdots, y_n\}$ $Q(y_1, \cdots, y_m)$ の決定問題は可解である．

（証明） ii）の formula は同値な i）の形に書き換えられることが知られているので i）の形についてのみ示す．

I） $P(x_1, \cdots, x_n)$ と $Q(x_1, \cdots, x_n)$ について，a）述語記号が相異なる．b）符号が同じである．c）argument の数が異なる．a）～b）の場合，*BPP* を適用しても $\Gamma, A, \Delta, \rightarrow A, \Lambda$ の形の sequence は得ら

れないので，証明不可能．

II) a)～b) 以外の場合．定理の formula は $Q(x_1, \cdots, x_n](P_{\theta_1 \cdots \theta_l} \vee \neg P_{\tau_1 \cdots \tau_l})$ の形と考えられる．(但し，θ_i, τ_j は free vairable か bound variable である．) この formula に *BPP* を適用すると branching がないので，補題-II よりその σ-列は λ-property を持つ．一方 "active" sequence の中に B-pair がいくつも取れるが，補題-III より B-pair の添字列が同じものは同一の clashability を持つので，定理の formula が証明可能であるか否かは，添字列の異なる B-pair についてだけ各々の clashability を調べればよい．そのような B-pair の数は有限であり，その中に clashability が 1 となるものがあれば定理の formula は証明可能であり，なければ証明不可能である．

〔**定理-III**〕 次の形をした formula†

i) $\bigvee_{i=1}^{n} \mathcal{O}_i(x_1, \cdots, x_{n_i}](P_{i_1}(x_1, \cdots, x_{n_i}) \vee \cdots \vee P_{i_{m_i}}(x_1, \cdots, x_{n_i}))$

ii) $\bigwedge_{i=1}^{n} \mathcal{O}_i(x_1, \cdots, x_{n_i}](P_{i_1}(x_1, \cdots, x_{n_i}) \wedge \cdots \wedge P_{i_{m_i}}(x_1, \cdots, x_{n_i}))$

の決定問題は可解である．但し $\bigvee_{i=1}^{n} A_i$, $\bigwedge_{i=1}^{n} A_i$ は，それぞれ $A_1 \vee \cdots \vee A_n$, $A_1 \wedge \cdots \wedge A_n$ の略である．

(証明) i) は定理-II から明らかであろう．ii) はその否定をとれば i) の場合に帰着する．

5. 証明手続 *DAIM*

前節の諸定理によって，可解な formula の形がわかったが，実際に証明可能か否かを判定する手続としては，定理-II の証明の中で示したように添字列の異なる全ての B-pair について調べる方法がある．しかしこの方法では quantifier の数が増すと手間が急激に増大する．以下に述べる証明手続 *DAIM* はこの欠点を除くもので，B-pair に相当するものに 1 回適用するだけで良い．*DAIM* の記述のために I)～IV) を仮定する．

I) dalah variable という概念を用いる．これを $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, {}_1\tilde{\alpha}, \tilde{\beta}_1, \tilde{\gamma}_1, \cdots$ で表わす．

II) III) で述べる δ-列，δ-formula を作る際に用いられる free variable および dalah variable は $a_1, a_2, \cdots$: $\tilde{\alpha}_1, \tilde{\alpha}_2, \cdots$ のように番号づけられ，この順に使われる．

III) $\mathcal{O}(x_1, \cdots, x_n]P(x_1, \cdots, x_n)$ に対してその δ-列，δ-formula を定義する．δ-列とは free variable と dalah variable よりなる列で，$P(x_1, \cdots, x_l)$ の中に free variable があればそれらをはじめに並べる．その後に，$\mathcal{O}(x_1, \cdots, x_n]$ を左端から順に調べて universal quantifier なら a_n, existential quantifier なら $\tilde{\alpha}_m$ を順に一列に並べたものである．$a_n, \tilde{\alpha}_m$ はそれぞれ今までに使われていない最小の番号の free varible, dalah variable である．δ-formula とは，$P(x_1, \cdots, x_n)$ の中にあらわれる全ての bound variable を δ-列と同じ規則に従って free variable か dalah variable に置き換えたものをいう．例えば，$\forall x \exists y \exists z P x y z a$ の δ-列は $\{a, a_1, \tilde{\alpha}_1, \tilde{\alpha}_2\}$, δ-formula は $P a_1 \tilde{\alpha}_1 \tilde{\alpha}_2 a$ である．

IV) 記述の簡単化のために，*DAIM* の対象となる formula を $\mathcal{O}_1(x_1, \cdots, x_m]A \vee \mathcal{O}_2(x_1, \cdots, x_n]B$ の形に限る．但し A と B は述語記号，argument の数が同じで，符号の異なる prime formula である．

〖証明手続 *DAIM*〗

〔*Step*-1〕 $\mathcal{O}_1(x_1, \cdots, x_m]A$ と $\mathcal{O}_2(x_1, \cdots, x_n]B$ の δ-列 Σ_1, Σ_2 および δ-formula $P_{\varepsilon_1, \cdots, \varepsilon_l}, \neg P_{\tau_1, \cdots, \tau_l}$ をそれぞれ作り *Step*-2 へ．ただし，$\mathcal{O}(x_1, \cdots, x_m]$, $\mathcal{O}_2(x_1, \cdots, x_n]$ によって導入した free variable, dalah variable は互に異なるものとする．

〔*Step*-2〕 2 つの δ-formula の argument の組 $(\varepsilon_i, \tau_i) 1 \leq i \leq l$ の中から，ε_i と τ_i が互に異なる free variable である組を探す．なければ *Step*-3 へ，あれば証明不可能である．

〔*Step*-3〕 $(\varepsilon_i, \tau_i) 1 \leq i \leq l$ の中で，一方が free variable a で，他方が dalah variable $\tilde{\alpha}$ である組を探す．なければ証明可能である．a が，$\tilde{\alpha}$ を含む δ-列 (Σ_1 または Σ_2 のどちらか) の中で $\tilde{\alpha}$ より後にあれば，証明不可能である．それ以外のとき，2 つの δ-formula の argument の中の全ての $\tilde{\alpha}$ を a で置き換えたものを新しい 2 つの δ-formula として *Step*-2 へ行く．

〔例題-1〕 $\exists x \forall y \exists z \neg P x y z y \vee \forall x \exists y \exists z P x y z z$ 上の formula の δ-列，δ-formula は，それぞれ i)，ii) である．

i) $\{\tilde{\alpha}_1, a_1, \tilde{\alpha}_2\}$, $\{a_2, \tilde{\alpha}_3, \tilde{\alpha}_4\}$.

ii) $(\neg P \tilde{\alpha}_1 a_1 \tilde{\alpha}_2 a_1,\ P a_2 \tilde{\alpha}_3 \tilde{\alpha}_4 \tilde{\alpha}_4)$.

iii) $(\neg P a_2 a_1 \tilde{\alpha}_2 a_1,\ P a_2 \tilde{\alpha}_3 \tilde{\alpha}_4 \tilde{\alpha}_4) : \tilde{\alpha}_1 \leftarrow a_2$

† i) の $n=1$ の場合については J. Herbrand [2].

iv) $(\rightarrow Pa_2a_1\tilde{\alpha}_2a_1,\ Pa_2a_1\tilde{\alpha}_4\tilde{\alpha}_4)$: $\tilde{\alpha}_3 \leftarrow a_1$

v) $(\rightarrow Pa_2a_1\tilde{\alpha}_2a_1,\ Pa_2a_1a_1a_1)$: $\tilde{\alpha}_4 \leftarrow a_1$

vi) $(\rightarrow Pa_2a_1a_1a_1,\ Pa_2a_1a_1a_1)$: $\tilde{\alpha}_2 \leftarrow a_1$ 証明可能.

〔例題-2〕 $\exists x\forall y\rightarrow Pxyx \vee \forall x\exists y Pxyy$

i) $\{\tilde{\alpha}_1, a_1\}$, $\{a_2, \tilde{\alpha}_2\}$ δ-列.

ii) $(\rightarrow P\tilde{\alpha}_1a_1\tilde{\alpha}_1,\ Pa_2\tilde{\alpha}_2\tilde{\alpha}_2)$ δ-formula

iii) $(\rightarrow Pa_2a_1a_2,\ Pa_2\tilde{\alpha}_2\tilde{\alpha}_2)$: $\tilde{\alpha}_1 \leftarrow a_2$.

iv) $(\rightarrow Pa_2a_1a_2,\ Pa_2a_1a_1)$: $\alpha_2 \leftarrow a_1$, a_1 と a_2 が異なる free variable なので証明不可能である.

$\mathcal{O}\{x_1, \cdots, x_n\}(A \vee B)$ の形の formula を扱う場合には, $\mathcal{O}\{x_1, \cdots, x_n\}$ に従って A, B の δ-formula $P_{\theta_1, \cdots, \theta_l}$, $\rightarrow P_{\theta_1', \cdots, \theta_l'}$ を作る. (但し A, B の中にあらわれる同じ bound variable に対しては, 同じ free variable または dalah variable を使う.) そして $\mathcal{O}\{x_1, \cdots, x_n\}$ から作った δ-列 Σ を, 2つの δ-formula に共通な δ-列として *DAIM* を *Step*-2 から適用する. ここで証明不可能となれば, さらに $\mathcal{O}\{x_1, \cdots, x_n\}A \vee \mathcal{O}\{x_1, \cdots, x_n\}B$ という formula に対して *DAIM* を適用すれば良い.

以上の手続きが, 添字列の異なる全ての B-pair を調べる方法と同等であることが示せるが省略する.

6. Implementation (Theorem Prover *TP-I*)

TP-I は, *BPP* を変形した完全性をもつ証明手続 *MPP* に *DAIM* を組合せて定理証明の効率を上昇させると同時に, 証明不可能であると判定できる sequence (formula) の範囲を拡大したものである. その他, 前もって登録された定理, 公理を証明手続の中で利用することが可能である.

MPP は Fig. 2に示す論理体系 *TS'* に基づくものである. *TS'* は *TS* の推論規則に "$\rightleftharpoons$", "$\rightarrow\rightleftharpoons$" "$\rightarrow\exists$", "$\rightarrow\forall$", "$\rightarrow\vee$", "$\rightarrow\wedge$" を独立した論理記号†としてその推論規則をつけ加えたものであるため, *MPP* は *BPP* にこれらの論理記号に対する手続を組み入れたものである. (2 II-ii) の formula の定義は, $(A) \rightleftharpoons (B)$ も formula であると拡張される.) 即ち, *BPP* の *Step*-1 では $A \vee B$ の形の formula のみを扱ったが, *MPP* では, $\rightarrow(A \vee B)$ の形の formula も分解して, $\rightarrow A$, B, Γ, Δ を sequence の列の最後につけ加える. *Step*-2, *Step*-5 においても同様に, それぞれ $\rightarrow\exists x A(x)$, $\rightarrow\forall x A(x)$ についても扱う. また, *Step*-3では $\rightarrow(A \vee B)$, $A \rightleftharpoons B$, $\rightarrow(A \rightleftharpoons B)$ で扱う.

† 例えば, $\rightarrow\rightleftharpoons$ は $\rightarrow(A \rightleftharpoons B)$ のように, $\rightarrow$ と $\rightleftharpoons$ を合せたものを扱うために便宜的に用いている.

○ Beginning sequence　$\Gamma, A, \Delta, \rightarrow A, \Lambda$

○ Infevence schemata

I) "$\wedge$" $\dfrac{A, \Gamma, \Delta \quad B, \Gamma, \Delta}{\Gamma, A \wedge B, \Delta}$　II) "$\rightarrow\vee$" $\dfrac{\rightarrow A, \Gamma, \Delta \quad \rightarrow B, \Gamma, \Delta}{\Gamma, \rightarrow(A \vee B), \Delta}$

III) "$\vee$" $\dfrac{A, B, \Gamma, \Delta}{\Gamma, A \vee B, \Delta}$　IV) "$\rightarrow\wedge$" $\dfrac{\rightarrow A, \rightarrow B, \Gamma, \Delta}{\Gamma, \rightarrow(A \wedge B), \Delta}$

V) "$\rightleftharpoons$" $\dfrac{\rightarrow A, B, \Gamma, \Delta \quad A, \rightarrow B, \Gamma, \Delta}{\Gamma, A \rightleftharpoons B, \Delta}$

VI) "$\rightarrow\rightleftharpoons$" $\dfrac{\rightarrow A, \rightarrow B, \Gamma, \Delta \quad A, B, \Gamma, \Delta}{\Gamma, \rightarrow(A \rightleftharpoons B), \Delta}$

VII)* "$\forall$" $\dfrac{A(a), \Gamma, \Delta}{\Gamma, \forall x A(x), \Delta}$　VIII)* "$\rightarrow\exists$" $\dfrac{\rightarrow A(a), \Gamma, \Delta}{\Gamma, \rightarrow\exists x A(x), \Delta}$

IX)** "$\exists$" $\dfrac{A(a), \Gamma, \Delta, \exists x A(x)}{\Gamma, \exists x A(x), \Delta}$

X)** "$\rightarrow\forall$" $\dfrac{\rightarrow A(a), \Gamma, \Delta, \rightarrow\forall x A(x)}{\Gamma, \rightarrow\forall x A(x), \Delta}$

* In VII) and VIII) free variable a does not appear in the lower sequence.

** In IX) and X) a is an arbitrary free variables.

Fig. 2 System *TS'*

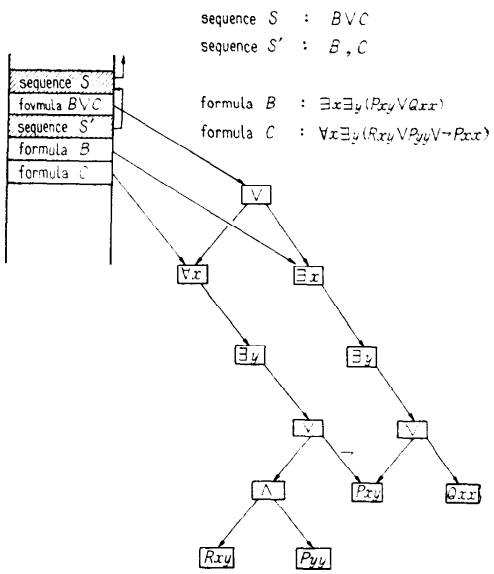

Fig. 3 Internal representation of formulas and decomposition of sequence S into S'

TP-I はこの *MPP* と前節の *DAIM* を中心として Fig. 4 のような構成である.

〖*TP-I* の構成〗

(I) (入力 sequence) 次のような形の formula を, コンマで区切ったものを入力 sequence として許す. Aおよび $\mathcal{O}A$ ($\mathcal{O}$ は任意の quantifier の並び, A は quantifier を含まない formula) から, 命題論理記号 $\rightarrow$, $\vee$, $\wedge$, $\rightleftharpoons$ を用いて帰納的に定義される formula である. 一般に, 任意の formula が同値なこの形の

formula に変換できることが知られている.

【内部表現】 formula に対して Fig. 3 のような tree が free storage 中に作られる. 入力 sequence の内部表現への変換時や, 証明の過程で生成, 分解される formula および sub-formula は, それと同じ形のものが以前に tree として作られていれば新たに tree を作ることはせず, 前の tree の先頭を pointer で指すだけである. Fig. 3 で, sequence S は S' に分解されたとき, formula B, C を指す pointer が作られる. sequence は, その sequence がどの sequence から分解されたかを示す後向きの pointer (Fig. 3 の斜線が施された部分) の次に, その sequence を構成する formula への pointer を並べたものとして表現される.

【定理の利用】 前もって登録してある定理, 公理を登録番号を指定することによって利用できる. 登録された定理, 公理の中にあらわれる述語記号を指定によって自由に変更することが可能で, 変更したものを入力 sequence の最後につけ加えて, 証明手続に入る.

【*DAIM* の適用指定】 指定によって (Ⅲ) から直接 (Ⅶ) に control を移すことができる.

(Ⅱ) 入力 sequence の構文や, 定理, 公理などの指定などに誤りがあれば, その旨 message を出力する.

(Ⅲ) ⅰ) $\vee$, $\rightarrow\wedge$ ⅱ) $\forall$, $\rightarrow\exists$ ⅲ) $\wedge$, $\rightarrow\vee$, $\rightleftharpoons$, $\rightarrow\rightleftharpoons$の論理記号に関してⅰ) ⅱ) ⅲ) の順の優先順位に従って formula を生成, 分解する. (3 の注意-1 参照.) この結果 "active" sequence がなくなれば証明可能で (Ⅷ) へ, $\exists x A(x)$, $\rightarrow\forall x A(x)$ の形の formula を持たない "active" sequence が残っていれば証明不可能で (Ⅸ) へ, その他の場合, *DAIM* に関する指定によって (Ⅳ) または (Ⅶ) へ行く.

(Ⅳ) 各 "active" sequence について, その中に次の形の formula または formula の組があるか調べる.

$\mathcal{O}(A\vee P\vee B\vee P'\vee C)$, $[\mathcal{O}_1(A_1\vee P\vee B_1), \mathcal{O}_2(A_2\vee P'\vee B_2)]$ (但し P と P' は互に述語記号, argument の数が同じで, 符号の異なる prime formula で, A, A_1, A_2, B, B_1, B_2 は任意の formula か空である. quantifier の前に $\rightarrow$ があるときには, de-Morgan の法則を繰返し用いて†, $\rightarrow$ を quantifier の並びの後に入れた形の formula とみなして調べることにする.) 上の 2 つのうち, 少くとも一方を持つ "active" sequence が 1 つでもあれば (Ⅴ) へ, なければ (Ⅶ) へ行く.

† 例えば $\rightarrow\forall x\exists y(Pxy\wedge\rightarrow Qxy)$ は $\exists x\forall y(\rightarrow Px\vee Qxy)$ とみなす.

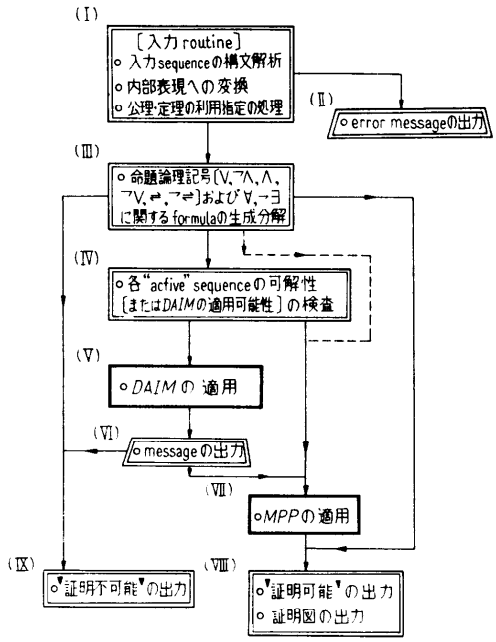

Fig. 4 General control flow of *TP-I*

(Ⅴ) (Ⅵ) で適用可能となった formula または, formula の組全てに対して, 証明手続 *DAIM* を適用する.

(Ⅵ) *DAIM* が適用された formula やその組を含む "active" sequence 全てに対して, *DAIM* の結果に従って, 証明可能, 証明不可能, あるいは不明である旨出力する. (不明となるのは, 例えば (Ⅳ) の formula で *DAIM* によって P と P' からは証明不可能となるが, Aの sub-formula として P または P' があらわれている場合がある.) 1 つでも証明不可能な "active" sequence があれば (Ⅸ) へ, その他の場合は (Ⅶ) へ行く.

(Ⅶ) 全ての "active" sequence について *MPP* を適用する. (Ⅵ) で証明可能となったものでも, 証明図出力のために *MPP* を適用する. その結果, 入力 sequence が証明可能となれば (Ⅷ) へ行く.

(Ⅷ) 入力 sequence が証明可能である旨出力し, 入力 sequence を end sequence とする体系 *TS'* の証明図を出力する.

(Ⅸ) 入力 sequence が証明不可能である旨出力する.

implementation には，TOSBAC 3400-model 31 (24 bit 16 kword) を用いた．

7. あとがき

入力 sequence に *MPP* を適用する前に，その sequence に *DAIM* が適用可能か調べるという方式は，*TP-I* の戦略 (Strategy) の1つと考えられる．*Resolution Principle* で用いられる Set of support strategy も *TP-I* に使うことができるが，その他に定理，公理の利用に関しては，入力 sequence の分解，生成の過程のどの時点で定理，公理を用いるかという問題や，複雑な形をした formula を簡単な形の formula に書き直して，定理，公理の利用が可能か調べる場合の戦略などが考えられる．これらの戦略は，証明手続の完全性を損わないものが重要視され，証明手続が全体として一様 (uniform) なものが今まで研究されている．しかし一様性に関しては，C. Hewitt[4] の問題解決用言語である PLANNER などは，全く逆の新しい方向に進んでいる．PLANNER の中心的な semantics は，*recommendation* に従って data base を探索するという証明手続としては極めて基本的なものがあるが，PLANNER で書かれた program 自身はこの探索の手順を細かく記述したものとなるため，問題自身の記述とそれに対する解決（証明）方法の分離が不明確となる場合が多い．この点に関して人工知能 system を扱う場合に充分な議論が必要であると思われる．

おわりに日頃御指導いただいている，東京大学の森口繁一先生，和田英一先生，立教大学の島内剛一先生に深く感謝する．また御討論いただいている森口研究室の皆様に感謝する．計算機の使用について便宜をはかっていただいた東京女子大の小河原正巳教授に感謝する．

参考文献

1) W. Ackermann: Solvable Cases of Decision Problem, p. 114, North-Holland, Amsterdam (1954).
2) G. Gentzen: Untersuchungen über das Logische Schließen Mathematische Zeitschrift, Bd. 39, SS. 176～210, (1934).
3) J. Herbrand: Recherches sur la théorie de la demonstration, Traveuax de la Société des Sciences et Lattres de Varsovie. Cl. III math-phys., Vol. 33, p. 128 (1930).
4) C. Hewitt: PLANNER: A Language for Manipulating Models and Proving Theorems in Robot, MIT. Project MAC Artificial Intelligence Memo, No. 168, (1968).
5) Z. Manna: Properties of Programs and the First-Order Predicate Calculus, JACM, Vol. 16, No. 2, pp. 244～255. April (1969).
6) Z. Manna, R. J. Waldinger: Toward Automatic Program Synthesis, CACM, Vol. 14, No. 3, pp. 151～165, (1971).
7) J. McCarty, P. J. Hayes: Some Philosophical Problems from the Standpoint of Aritificial Intelligence, Machine Intelligence, Vol. 4, pp. 463～503, (1969).
8) J. A. Robinson: A Machine-oriented Logic-based on the Resolution Principle, JACM, Vol. 12, No. 1, pp. 24～41, (1965).
9) J. A. Robinson: A Review of Automatic Theorem-proving. In Proc. Symposia in Math. App. 1965, Vol. 19. AMS, Providence, R. I., pp. 1～18, (1967).
10) 島内剛一：証明のプログラム，数学，第15巻，pp. 48～55, (1963～1964).
11) 島内剛一：*LK* の証明のプログラム，第1回プログラミングシンポジウム予稿集，(1960).
12) 米澤明憲：On a Proof Procedure of the First-Order Predicate Calculus，東京大学大学院工学系研究科修士論文，(1972).
13) 米澤明憲：Theorem Prover *TP-I*，情報処理学会第12回大会予稿集，pp. 239～240, (1971).
14) H. Wang: Toward Mechanical Mathematics. IBM Journal, Vol. 4, No. 1, pp. 2～22, (1960).

（昭和47年7月20日受付）
（昭和47年8月28日再受付）