数理解析研究所講究録
第 189 巻 1973 年 70-81

A FORMAL SYSTEM OF PROGRAMS

Yutaka Kanayama
University of Electro-Communications

INTRODUCTION There has been a growing sentiment that the use of go to statement is undesirable.[2][5] In this paper we shall propose a programming language system in which the function of go to statements is restricted.

Go to statements are roughly divided into two categories. Forward ones are used for switching the flow of control and backward ones for making loops. In the language described in this paper,

(1) forward go to statements are unnecessary because they are represented by another syntactical structure, and

(2) backward goto statements are expressed by brackets [and].

They are regarded as a label and a go to statement respectively. The power of the both statements in our system is weaker than that in usual programming languages and in the system proposed by Yanov.[7] Because of the restricted use, however, our programs are suitable for mechanical processing.

1. SYNTAX The sets of function symbols $f_0, f_1, \cdots$ and predicate symbols $p_0, p_1, \cdots$ are denoted by F and P. Let $A = F \cup P \cup \{[,]\}$. All members of A are called basic symbols(bs's). We shall treat formulas composed of these symbols, (,),• and +. Well-formed formulas(wf's) of this theory are defined as follows:

(a) Every formula which consists of a single basic symbol is a wf.

(b) If α, β are wfs, then $(\alpha \cdot \beta)$ and $(\alpha + \beta)$ are wfs.

(c) A formula is wf only if it can be a wf on the basis of clauses (a) and (b).

Example 1.1 $((((f_1 \cdot p_1) \cdot]) + (([\cdot p_2) \cdot (f_2 \cdot f_2))) + ((] \cdot p_1) \cdot]))$ is a wf.

The phrase structure of every wf α is unambiguous. If α and β_i are arbitrary formula for $i=1,\cdots,n$ and $c_1,\cdots,c_n$ occurrences of basic symbols (in γ), $\alpha^{c_1 \cdots c_n}_{\beta_1 \cdots \beta_n}$ denotes a formula obtained from α by the substitution of $\beta_1,\cdots,\beta_n$ for $c_1,\cdots,c_n$ in α respectively. If $\beta_1,\cdots,\beta_n$ and α are wf, then so is $\alpha^{c_1 \cdots c_n}_{\beta_1 \cdots \beta_n}$. We write $\alpha \equiv \beta$ if α and β are identical formulas. If β, γ are wfs, c is an occurrence of a bs (in γ) and $\alpha \equiv \gamma^{c}_{\beta}$, then β is a subformula of α.

We shall employ some abbreviations:

(a) The outermost parentheses may be omitted,

(b) $\alpha \cdot \beta + \gamma$ is understood to mean $(\alpha \cdot \beta) + \gamma$, and

(c) the dots(•) may be omitted.

The leftmost occurrence c of a bs in a wf is called an entrance

$C(\alpha)$ denotes the set of all occurrences of bs's in α. For every wf α, $e_+(\alpha) \subseteq C(\alpha)$ is defined as follows:

$$\begin{cases} e_+(a) = \{\text{the occurrence of } a \text{ itself}\}, & \text{if } a \varepsilon A; \\ e_+(\alpha\cdot\beta) = e_+(\beta), \\ e_+(\alpha+\beta) = e_+(\alpha) \cup e_+(\beta). \end{cases}$$

If $c \;\varepsilon\; e_+(\alpha)$, c is called a <u>+exit</u> of α. Every α has at least one +exit.

For each wf α, a relation $t_+(\alpha) \subseteq C(\alpha)^2$ is defined as follows:

$$t_+(\alpha) = \{(c_1,c_2) \mid (E\beta)(E\gamma)[(\beta\gamma) \text{ is a subformula of } \alpha \;\&\; c_1 \varepsilon e_+(\alpha) \;\&\; c_2 \text{ is the entrance of } \gamma]\}.$$

This relation is called a <u>+transition relation</u>.

<u>Lemma 1.1</u> For any $c \;\varepsilon\; C(\alpha)$, exactly one of the following conditions holds:

(i) $c \;\varepsilon\; e_+(\alpha)$,

(ii) $(E!c')[(c,c') \;\varepsilon\; t_+(\alpha)]$.

<u>Example 1.2</u> In the examples below, underlines indicate + exits and arrows + transition relations.

$$\alpha_1 \equiv ((a_1 a_2)\underline{a}_3 + (a_4 a_5)\underline{a}_6) + (a_7 a_8)\underline{a}_9,$$

$$\alpha_2 \equiv (((a_1+a_2)+a_3)((a_4+a_5)+a_6))((\underline{a}_7+\underline{a}_8)+\underline{a}_9),$$

where $a_1,\cdots,a_9 \;\varepsilon\; A$.

Analogously, the set $e_-(\alpha)$ of <u>- exit</u> of α and the <u>- transition relation</u> $t_-(\alpha)$ are defined as follows:

$$\begin{cases} e_-(a) = \{\text{the occurrence of } a \text{ itself}\}, & \text{if } a \;\varepsilon\; A; \\ e_-(\alpha+\beta) = e_-(\beta), \\ e_-(\alpha\cdot\beta) = e_-(\alpha) \cup e_-(\beta). \end{cases}$$

$$t_-(\alpha) = \{(c_1,c_2) \mid (E\beta)(E\gamma)[(\beta+\gamma) \text{ is a subformula of } \alpha \;\&\; c_1 \;\varepsilon\; e_-(\beta) \;\&\; c_2 \text{ is the entrance of } \gamma]\}.$$

<u>Lemma 1.2</u> For any $c \in C(\alpha)$, exactly one of the following conditions holds:

(i) $c \in e_-(\alpha)$,

(ii) $(E!c')[(c,c') \in t_-(\alpha)]$.

<u>Example 1.3</u> -exits and - transition relations are indicated as in example 1.2.

$$\alpha_1 \equiv ((a_1a_2)a_3 + (a_4a_5)a_6) + (\underline{a}_7\underline{a}_8)\underline{a}_9,$$

$$\alpha_2 \equiv (((a_1+a_2)+\underline{a}_3)((a_4+a_5)+\underline{a}_6))((a_7+a_8)+\underline{a}_9).$$

Now for every wf α, an integer $q(c)$ is assigned to each occurrence c of a bs in α. If c is the i th occurrence of a bs from left in α, then $q(c) = i$. If c is the entrance, $q(c) = 1$. The number $q(c)$ is called the <u>states</u> of c.

Let $Q(\alpha)$ denote the set of all states of occurrences of bs's in α. E_+, E_-, T_+, T_- are defined as follows:

$E_\pm(\alpha) = \{q(c) | c \in e_\pm(\alpha)\}$,

$T_\pm(\alpha) = \{(q(c_1), q(c_2)) | (c_1, c_2) \in t_\pm(\alpha)\}$.

<u>Example 1.4</u> Let α_1, α_2 be those given in example 1.2 and 1.3.

$E_+(\alpha_1) = E_-(\alpha_2) = \{3,6,9\}$,

$E_-(\alpha_1) = E_+(\alpha_2) = \{7,8,9\}$,

$T_+(\alpha_1) = T_-(\alpha_2) = \{(1,2),(2,3),(4,5),(5,6),(7,8),(8,9)\}$,

$T_-(\alpha_1) = T_+(\alpha_2) = \{(1,4),(2,4),(3,4),(4,7),(5,7),(6,7)\}$.

Note the duality among operators, exits and relations.

<u>Lemma 1.3</u> For any α, q_1, q_2, q_3 and q_4, if $(q_1,q_2),(q_3,q_4) \in t_+(\alpha)$ or $t_-(\alpha)$ and $q_1 < q_3$, then $q_2 \leqslant q_3$ or $q_4 \leqslant q_2$.

This fact indicates the restriction of the power of "forward go to statements" permitted by this system.

Functions sbl(symbol) and lbl(label) are defined as follows:

sbl(α,q) = the basic symbol which occurs at the state q in the wf α, if q ε Q(α).

If all the occurrence of brackets in a wf are properly nested, then we can find a corresponding one for each one. Such an occurrence is called an <u>image</u> of the other.

$$\mathrm{lbl}(\alpha,q) = \begin{cases} r, & \text{if } \mathrm{sbl}(\alpha,q) = \,] \text{ and it has an image of which state equals } r; \\ q, & \text{otherwise.} \end{cases}$$

<u>Example 1.5</u> Let $\alpha_3 \equiv ((((((p_1])[)f_1)p_2)])]) + f_2$. Then

$\mathrm{sbl}(\alpha_3,2) = \mathrm{sbl}(\alpha_3,6) = \mathrm{sbl}(\alpha_3,7) = \,]$,

$\mathrm{lbl}(\alpha_3,2) = 2$, $\mathrm{lbl}(\alpha_3,6) = 3$, $\mathrm{lbl}(\alpha_3,7) = 7$.

2. SEMANTICS

An <u>interpretation</u> I is a triple (D,v_1,v_2), where D is a non-empty set and v_1,v_2 are functions: $F \to D^D$ and $P \to \{T,F\}^D$ respectively. C^B denotes the set of all total functions: $B \to C$. We neglect the possible fine structure of D, v_1 and v_2 in this paper.

Assume a wf α and an interpretation I are given. $\xi = (q,x)$ is called a <u>description with respect to α and I</u> if $q \;\varepsilon\; Q(\alpha) \cup \{\omega\}$ and $x \;\varepsilon\; D$. The set of all descriptions $(Q(\alpha) \cup \{\omega\}) \times D$ is denoted by H_+. Furthermore H denotes $Q(\alpha) \times D$.

A relation $\xrightarrow[\alpha,I]{} \subseteq H_+ \times H$ is defined as follows:

$$(q,x) \xrightarrow[\alpha,I]{} (r,y) \longleftrightarrow$$

$$\{(sbl(\alpha,q) \varepsilon F \ \&\ y=v_1(sbl(\alpha,q))(x)) \bigvee (sbl(\alpha,q) \not\varepsilon F \ \&\ y=x)\} \ \&$$

$$\{(sbl(\alpha,q)=] \ \&\ r=lbl(\alpha,q))$$

$$\bigvee(sbl(\alpha,q)\varepsilon P \ \&\ v_2(sbl(\alpha,q))(x)=F \ \&\ q\varepsilon E_-(\alpha) \ \&\ r=q)$$

$$\bigvee(sbl(\alpha,q)\varepsilon P \ \&\ v_2(sbl(\alpha,q))(x)=F \ \&\ (q,r)\varepsilon T_-(\alpha))$$

$$\bigvee((sbl(\alpha,q)\varepsilon P \ \&\ v_2(sbl(\alpha,q))(x)=T) \bigvee sbl(\alpha,q)\varepsilon F \cup \{[\}) \ \&\ q\varepsilon E_+(\alpha) \ \&\ r=\omega)$$

$$\bigvee((sbl(\alpha,q)\varepsilon P \ \&\ v_2(sbl(\alpha,q))(x)=T) \bigvee sbl(\alpha,q)\varepsilon F \cup \{[\}) \ \&\ (q,r)\varepsilon T_+(\alpha))\}.$$

For any α, I and $\xi \varepsilon H$, there exists a unique $\eta \varepsilon H_+$ such that $\xi \xrightarrow[\alpha,I]{} \eta$. We write $\xi \xrightarrow[\alpha,I]{+} \eta$ if there exists a sequence $\xi_0,\cdots,\xi_t$ $(t>0)\ \varepsilon\ H_+$ such that $\xi=\xi_0$, $\xi_m \xrightarrow[\alpha,I]{} \xi_{m+1}$ for $m=0,\cdots,t-1$ and $\xi_t=\eta$.

We say ξ <u>converges in α under</u> I if there exists $y \varepsilon D$ such that $\xi \xrightarrow[\alpha,I]{+} (\omega,y)$.

Floyd states that if there exists some well-ordering on D with respect to a program, then it converges.[3] Later Manna gave a sufficient condition by assigning a well-ordered set to each vertex of a flowchart.[6] Next theorem is a stronger result because it gives a necessary and sufficient condition for the convergence.

<u>Theorem 2.1</u> Any $\xi \varepsilon H$ converges in α under I iff there exists a relation $\cdot\!\!> \subseteq H^2$ such that $\cdot\!\!>$ is a well-founded ordering and $(\xi)(\eta)[(\xi \xrightarrow[\alpha,I]{} \eta) \rightarrow \xi \cdot\!\!> \eta]$.

The proof is parallel to that for Theorem 1 in [4].

A partial function $\Psi(\alpha,I)$ is defined as follows:

$$\Psi(\alpha,I)(x) = \begin{cases} y, & \text{if } (1,x) \xrightarrow[\alpha,I]{+} (\omega,y); \\ \text{undefined}, & \text{if } (1,x) \not\xrightarrow[\alpha,I]{+} (\omega,y) \text{ for any } y. \end{cases}$$

We write $\alpha = \beta$ if $\Psi(\alpha,I) = \Psi(\beta,I)$ for every interpretation I. Furthermore, we write $\alpha =_s \beta$ (α is <u>strongly equivalent to</u> β), if for every wf γ and an occurrence $c \in C(\gamma)$, $\gamma^c_\alpha = \gamma^c_\beta$. By the following result, we can omit more parentheses.

<u>Lemma 2.2</u> (a) $(\alpha)(\beta)(\gamma)[(\alpha\beta)\gamma =_s \alpha(\beta\gamma)]$,

(b) $(\alpha)(\beta)(\gamma)[(\alpha+\beta)+\gamma =_s \alpha+(\beta+\gamma)]$.

An interpretation I is called <u>standard</u> if $v_1(f_0)$ = the identity function. Hereafter every I will be assumed to be standard.

<u>Example 2.1</u> Consider a pseudo algol program for Euclidean algorithm.

aa: <u>if</u> $x_1 = x_1 \div x_2 \times x_2$ <u>then</u> <u>goto</u> bb;

$(x_1,x_2) := (x_2,x_1 - x_1 \div x_2 \times x_2)$;

<u>goto</u> aa;

bb: ....

When the control goes to bb, the value of x_2 is the GCD of the initial values of x_1 and x_2. This program is expressed in our system under "conventional" standard interpretation as follows:

$$\alpha \equiv \underset{1\,2}{[}p_1 \underset{3}{f_1} \underset{4}{]} + \underset{5}{f_0},$$

where p_1 and f_1 has the meanings,

p_1: $x_1 > x_1 \div x_2 \times x_2$,

f_1: $(x_1,x_2) := (x_2, x_1 - x_1 \div x_2 \times x_2)$.

If $x_1=3178$ and $x_2=434$, then the computation is

$(1,(3178,434)) \to (2,(3178,434)) \to (3,(3178,434)) \to (4,(434,140)) \to$
$(1,(434,140)) \to (2,(434,140)) \to (3,(434,140)) \to (4,(140,14)) \to$
$(1,(140,14)) \to (2,(140,14)) \to (5,(140,14)) \to (\omega,(140,14))$.

A flowchart of this wf is shown in Figure 1.

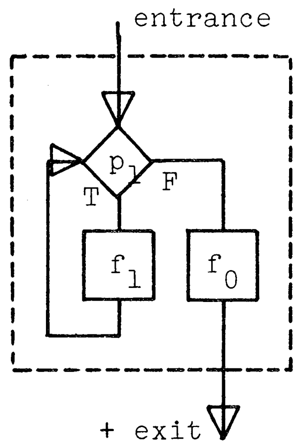


Figure 1.

Example 2.2 Next one is a Turing machine Z taken from example 3.7 in [1]. It contains 23 quadruples and computes the function: $(x,y) \mapsto (x+1)(y+1)$.

$$Z = \left\{\begin{array}{llll}
q_1\ 1\ 0\ q_1 & q_4\ 1\ R\ q_3 & q_6\ 0\ 0\ q_{10} & q_9\ 0\ L\ q_9 \\
q_1\ 0\ R\ q_2 & q_4\ 0\ L\ q_5 & q_7\ 1\ R\ q_7 & q_9\ \eta\ 1\ q_5 \\
q_2\ 1\ \varepsilon\ q_3 & q_5\ 1\ L\ q_6 & q_7\ 0\ R\ q_8 & q_{10}\ 1\ L\ q_{10} \\
q_3\ \varepsilon\ R\ q_3 & q_5\ 0\ L\ q_6 & q_8\ 1\ R\ q_8 & q_{10}\ 0\ L\ q_{10} \\
q_3\ 1\ R\ q_3 & q_6\ 1\ \eta\ q_6 & q_8\ 0\ 1\ q_9 & q_{10}\ \varepsilon\ 0\ q_1 \\
q_3\ 0\ R\ q_4 & q_6\ \eta\ R\ q_7 & q_9\ 1\ L\ q_9 &
\end{array}\right\}$$

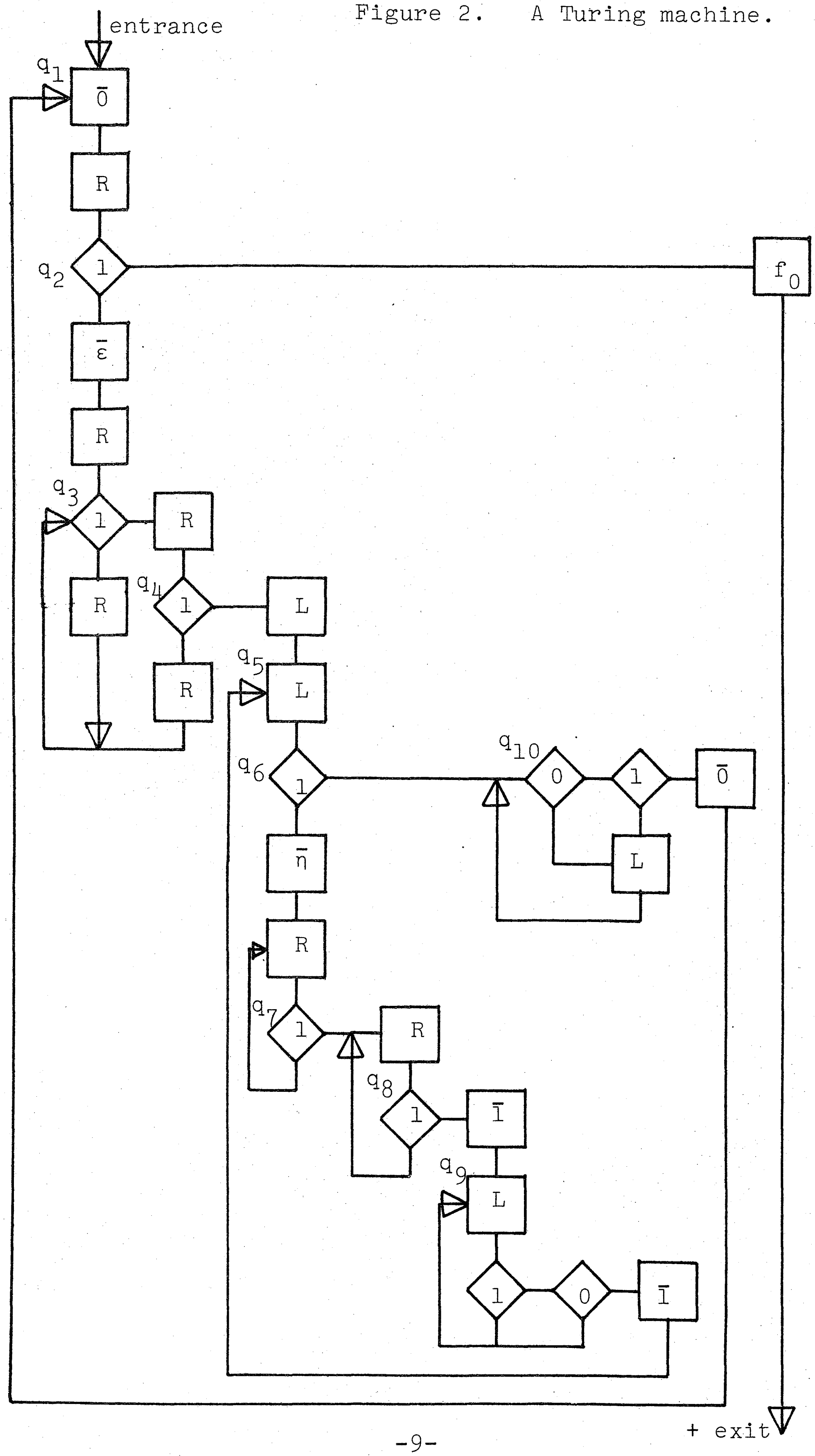
This algorithm can be expressed by the following wf:

$[\bar{0}R1(\bar{\varepsilon}R[[1R]+R1R]+L[L1(\bar{\eta}[R1]+[R1]+\bar{1}[L(1+0)]+\bar{1}])+[(0+1)L]+\bar{0}])+f_0$.

The symbols 0 and 1 are predicate symbols.

0(resp. 1) means a predicate that the letter being read is 0(resp. 1).

Figure 2. A Turing machine.



The symbols $\bar{0},\bar{1},\bar{\varepsilon},\bar{\eta}$,R and L are function symbols.

$\left\{\begin{array}{l}\bar{0}\text{(resp. }\bar{1},\bar{\varepsilon},\bar{\eta}\text{) means that "write 0(resp. }1,\varepsilon,\eta\text{)", and} \\ \text{R(resp. L) means that "move right(resp. left)".}\end{array}\right.$

A flowchart of this wf is shown in Figure 2.

3. NORMAL FORMS

A wf is called <u>neutral</u> if all the brackets in it have their images.

<u>Lemma 3.1</u> For any wf α, there exists a neutral β such that $\alpha = \beta$.

A wf is called <u>- exit-free</u>(or <u>mef</u>) if it contains no - exits which are occurrences of predicate symbols.

<u>Lemma 3.2</u> For any wf α, there exists a mef β such that $\alpha = \beta$.

A wf is called <u>independent</u> if it is neutral and mef. Independent wfs have several desirable properties for formal manipulation. Wfs in example 2.1 and 2.2 are independent.

<u>Lemma 3.3</u> If α,β are independent and $\alpha = \beta$, then $\alpha =_s \beta$.

<u>Lemma 3.4</u> If (a) a wf α is independent, (b) $c_1,\cdots,c_k (k\geq 0)$ are occurrences of ['s in α, (c) $c_{k+1},\cdots,c_{2k}$ are occurrences of]'s in α, (d) images of $c_1,\cdots,c_k$ are $c_{2k},\cdots,c_{k+1}$ respectively, (e) states of $c_1,\cdots,c_k$ are $1,\cdots,k$ respectively, (f) no +'s occur between c_1 and c_k, and (g) $c_{k+1},\cdots,c_{2k}$ are +exits in α,then

$$\alpha = \alpha^{c_1 \cdots c_k c_{k+1} \cdots c_{2k}}_{f_0 \cdots f_0 \, \alpha \; \cdots \; \alpha} .$$

A wf is called <u>loop-normalized</u> if all of its occurrences of]'s are in its + exits. Next proposition shall be a "normal form theorem" in this theory if it can be proved.

<u>Conjecture 3.5</u> For any wf α, there exist $n(>0), \alpha_1, \cdots, \alpha_n$, $\beta_1, \cdots, \beta_n$ such that

$$\begin{cases} \alpha = \alpha_1, & \\ \alpha_i = \beta_i \, {}^{c_1 \cdots c_{m(i)}}_{\alpha_{j_1} \cdots \alpha_{j_{m(i)}}} & \text{for } i=1,\cdots,n, \end{cases}$$

where $m(i) \geqslant 0$ for $i=1,\cdots,n$,

$1 \leqslant j_1, \cdots, j_{m(i)} \leqslant n$ for $i=1,\cdots,n$,

$\alpha_1, \cdots, \alpha_n$ are independent and loop-normalized,

$\beta_1, \cdots, \beta_n$ contain no brackets, and

$c_1, \cdots, c_{m(i)}$ are + exits in β_i for $i=1,\cdots,n$.

If this is provable, we can construct a system of equations from $\beta_1, \cdots, \beta_n$ such that α is the minimal solution of it.

REFERENCES

[1] Davis,M., Computability and Unsolvability, McGRAW-HILL Inc., New York,(1958).

[2] Dijkstra,E.W., " Go to statement considered harmful", CACM, vol.11, pp147-148(1968).

[3] Floyd,R.W., "Assigning meaning to programs", Proceedings of Symposia in Applied Mathematics, Am. Math. Soc., vol. 19, pp. 19-32(1967).

[4] Kanayama,Y., "Algebraic Properties of Programs", Proc. 1st USA-Japan Computer Conference, pp. 380-383(1972).

[5] Knuth,D.E.and R.W. Floyd, "Notes on Avoiding "go to" statements", Information Proc. Letters, vol. 1, pp. 23-31 (1971).

[6] Manna,Z., "Termination of Programs Represented as Interpreted Graphs", Proc. SJCC, pp. 83-89(1970).

[7] Yanov,Y.I., "The Logical Schemes of Algorithms", Problems of Cybernetics, vol. 1, pp.82-140, Pergamon Press, New York(1960).