REINFORCED LOGICS

KATUZI ONO

Introduction

The device of representing a predicate by a universal sentence[1] gives rise to logics of higher order. To represent an object constructed by a series of certain steps, we usually use terms. To represent a predicate constructed by a series of certain steps, we now use sentences in general. We can now substitute special predicates represented by sentences for predicate variables of a sentence just as we have been able to substitute special objects represented by terms for object variables. It is remarkable that we can always distinguish predicates from objects, proceeding in this way.

The logic proper for this kind of sentences would be a logic of higher order. Two kinds of variables, object variables and predicate variables, both admitting quantification would be used in the logic. In the present paper, I will introduce a series of logics of this kind, which I call REINFORCED LOGICS. The reinforced logic of a logic **LX** is denoted by **LXR**.

I will introduce a series of reinforced logics **LXR** in (1). Broadly speaking, any reinforced logic can be regarded as a type-theoretical logic assuming only two types and having the axiom of reducibility.[2] The logic **LKR** must be closely connected with Takeuti's **GLC**.[3] The reinforced logic **LOR** would be a basic logic for a series of logics **LXR** in similar manner as that **LO** is basic for a series of logics such as **LK**, **LJ**, **LM** etc.[4]

Received September 13, 1965.

1) This device has been one of the leading ideas of my preceeding work, Ono [1]. For example, $u<v$ can be regarded as a special case (u for x and v for y) of the universal sentence $(x)(y)x<y$. I represent the binary predicate "$<$" by the universal sentence $(x)(y)x<y$, so $u<v$ can be expressed also by $((x)(y)x<y)(u, v)$.

2) See Whitehead and Russell [1].

3) See Takeuti [1].

4) See Ono [2] and [3]. I regard the primitive logic **LO** basic because popular logics such as the lower classical predicate logic **LK**, the intuitionistic predicate logic **LJ**, Johansson's minimal predicate logic **LM**, etc. can be faithfully interpreted in it. As for details, see Ono [2].

We can give a formalism for **LOR** similar to that for **LO** given in my preceding paper.[5] **FLO** refers to the formalism for **LO** and **FLOR** refers to the new formalism for **LOR** which will be introduced in the present paper. Formal material of **FLOR** are the same as that of **FLO**. Namely, only one kind of variables together with the head- and tail-brackets and the comma are used in **FLOR**.

FLOR is so designed that any sequence of variables can be regarded as an index and any sequence of variables and brackets can be regarded as a sentence as far as it contains at least one bracket. Most remarkable difference between **FLOR** and **FLO** lies in interpretation of sentences. Formal exposition of **FLOR** is given in **(2)**.

In **(3)**, I will mention my view about future developments and possible applications of the reinforced logics.

(1) Informal description of reinforced logics

In this section, we describe reinforced logics by dealing with two kinds of variables, object and predicate variables. The former are denoted by small Latin letters and the latter are denoted by capital Latin letters. I use also usual logical constants such as IMPLICATION, UNIVERSAL and EXISTENTIAL QUANTIFICATIONS together with some other auxiliary meta-logical symbols.

Any elementary sentence is a series of symbols of the form $P(R, \dots, T; x, \dots, z)$ in which any predicate variable can be replaced by any sentence. $P(R, \dots, T; x, \dots, z)$ can be interpreted as the predicate P having $R, \dots, T$ as parameters holds for $x, \dots, z$. Both series $R, \dots, T$ and $x, \dots, z$ may be vacant. Any sentence whose elementary sub-sentences are all of the form $P(; x, \dots, z)$ is called STANDARD. In the following, $P(; x, \dots, z)$ is expressed simply as $P(x, \dots, z)$ to harmonize with the usual expression.

Any sentence of the form $\mathfrak{P}(R, S, \dots, T ; x, y, \dots, z)$ *i.e.* any sentence having free predicate variables $R, S, \dots, T$ and free object variables $x, y, \dots, z$ requires no explanation proper to the reinforced logics unless $\mathfrak{P}$ is a sentence of the forms $(^*u)\mathfrak{H}$ or $(^*U)\mathfrak{H}$. Here "$(*\ )$" stands for either "$(\ )$" or "$(\exists\ )$". In this case, $\mathfrak{P}(R, S, \dots, T ; x, y, \dots, z)$ means $\mathfrak{H}^*(R, S, \dots, T ; y, \dots, z)$

[5] Ono [1].

in the former case and $\mathfrak{H}^{**}(S, \ldots, T;\ x, y, \ldots, z)$ in the latter case, where $\mathfrak{H}^*$ and $\mathfrak{H}^{**}$ are sentences obtained from $\mathfrak{H}$ on replacing all the variables u or U bound to the quantifying variable u or U standing at the top of $({}^*u)\mathfrak{H}$ or $({}^*U)\mathfrak{H}$ by x or R, respectively. Notice that it does not matter here whether the quantifier at the top is universal or existential. In $\mathfrak{P}(R, S, \ldots, T;\ x, y, \ldots, z)$, any sentences $\mathfrak{R}$, $\mathfrak{S}$, $\ldots$, $\mathfrak{T}$ can be substituted for $R, S, \ldots, T$, respectively.

Inference rules of **LXR** for the primitive logic or for any **J**- or **K**-series logics **LX**[6] are the same as in **LX** except for quantifications. So, I will write down only inference rules for quantifications.

(**A**) For any sentence $\mathfrak{P}$, $(x)\mathfrak{P}$ holds if $(x)\mathfrak{P}(u)$ holds for every object variable u (*i.e.* $\forall u!$), and $(X)\mathfrak{P}$ holds if $(X)\mathfrak{P}(U;)$ holds for every predicate variable U (*i.e.* $\forall U!$).

(**B**) For any sentence $\mathfrak{P}$, $(x)\mathfrak{P}$ implies $(x)\mathfrak{P}(u)$ for any variable u which does not occur in $\mathfrak{P}$ as bound variables. For any sentence $\mathfrak{P}$, $(X)\mathfrak{P}$ implies $(X)\mathfrak{P}(\mathfrak{R};)$ for any *standard* sentence[7] $\mathfrak{R}$ which has no free variables occurring in $\mathfrak{P}$ as bound variables.

The following inference rules are assumed only in the reinforced logic **LXR** of a logic **LX** which admits existential quantifiers.

(**C**) For any sentence $\mathfrak{P}$ and any variable u, $(\exists x)\mathfrak{P}(u)$ implies $(\exists x)\mathfrak{P}$, assuming that u does not occur as bound in $\mathfrak{P}$. For any sentence $\mathfrak{P}$ and any *standard*[8] sentence $\mathfrak{R}$, $(\exists X)\mathfrak{P}(\mathfrak{R};)$ implies $(\exists X)\mathfrak{P}$, assuming that any free variable occurring in $\mathfrak{R}$ does not occur in $\mathfrak{P}$ as bound.

---

6) I call **LJ**, **LM**, and the positive predicate logic **LP** altogether **J**-SERIES LOGICS. **LK**, **LN**, and **LQ** obtained by fortifying **LJ**, **LM**, and **LP** by Peirce's rule ("$(\mathfrak{A}\rightarrow\mathfrak{B})\rightarrow\mathfrak{A}$ implies $\mathfrak{A}$") are called altogether **K**-series logics. See Ono [2].

7), 8) These restrictions are necessary for avoiding trivial paradoxes of Russell-type. Namely, if we remove these restrictions in the modified logic **LMR*** of **LMR**, for example, then we can prove in **LMR*** (but not in **LMR**)

$$(\exists Q)(X)(Q(X;)\equiv\neg X(X;))$$

from the easily provable sentence

$$(\exists Q)(X)(Q(X;)\equiv\neg X(X;))((X)\neg X(X;);).$$

Now, by assumption, we can take a Q satisfying (this can be done in **LMR** too)

$$(X)(Q(X;)\equiv\neg X(X;)),$$

so we can prove in **LMR*** (but not in **LMR**)

$$(X)(Q(X;)\equiv\neg X(X;))(Q;)\ \textit{i.e.}\ Q(Q;)\equiv\neg Q(Q;),$$

which surely leads to contradiction in the usual way.

(D) If $(\exists x)\mathfrak{P}$ is assumed for any sentence $\mathfrak{P}$, we can take a new variable u for which we can assume $(\exists x)\mathfrak{P}(u)$ *i.e.* $\exists u!\ (\exists x)\mathfrak{P}(u)$. Similarly, if we assume $(\exists X)\mathfrak{P}$, we can take a new variable U for which $(\exists X)\mathfrak{P}(U;)$ can be assumed *i.e.* $\exists U!\ (\exists X)\mathfrak{P}(U;)$.

Before concluding the first part of this section, I would like to exhibit a simple example proof in **LOR**, which can not be followed in **LO**.

Namely, we can define $a=b$ in **LOR** by $(P)(P(a)\to P(b))$. To prove symmetricity of the equality thus defined *i.e.* to prove $R(b)\to R(a)$ for every predicate variable R by assuming $(P)(P(a)\to P(b))$, we have only to adopt the sentence $(x)(R(x)\to R(a))$ in place of P. Because $P(a)\to P(b)$ is standard, we can prove $R(b)\to R(a)$ in **LOR**.

Although it is very practical to use two kinds of letters distinctly for expressing object variables and predicate variables, it is not necessary to do so. Even when we denote object variables and predicate variables by the same kind of letters, we can distinguish from the context whether a variable in a sentence is an object variable or a predicate variable except for quantifying variables.

Quantification itself can be easily interpreted so as to match with our unification. For example, $(u)\mathfrak{P}$ can be interpreted as "For all objects u and for all predicates u holds $\mathfrak{P}$", *i.e.* $(u)(U)\mathfrak{P}^*$ according to the original notation denoting predicate variables by capital letters.

We have only to interpret substitutions such as $(x)\mathfrak{H}(u)$, which must be originally a sentence of the form $(x)\mathfrak{H}^*(u)$ or $(X)\mathfrak{H}^{**}(U)$. For this purpose, we had better adopt a new way of expressing propositions. As a typical form of elementary sentences, we adopt now $P(u, U; \cdots; w, W : x, \cdots, z)$, denoting predicate variables by capital letters.

Accordingly, the same proposition is expressed as $p(u, u; \cdots; w, w : x, \cdots, z)$ if we denote object variables as well as predicate variables by small letters. Any elementary sentence has the form $p(u, r; \cdots; w, t : x, \cdots, z)$ and every variable in the sentence is distinguished by the scheme

$$P(O, P; \cdots; O, P : O, \cdots, O)$$

(O for object variables and P for predicate variables).

Object variables can be replaced by variables only, but predicate variables can be replaced by variables as well as by sentences.

Quantifying variables can be regarded as variables of both characters, object-like as well as predicate-like. Namely,

$$(x)\mathfrak{h}(u, \mathfrak{r}\ ;\ v, \mathfrak{s}\ ;\ \cdots\ ;\ w, \mathfrak{t} : x, \cdots, z)$$

as well as

$$(\exists x)\mathfrak{h}(u, \mathfrak{r}\ ;\ v, \mathfrak{s}\ ;\ \cdots\ ;\ w, \mathfrak{t} : x, \cdots, z)$$

means $\mathfrak{h}^*(v, \mathfrak{s}\ ;\ \cdots\ ;\ w, \mathfrak{t} : x, \cdots, z)$, where $\mathfrak{h}^*$ is the sentence obtained from $\mathfrak{h}$ on replacing every object variable x in $\mathfrak{h}$ bound to the quantifying variable x standing at the top of the sentence by u and every predicate variable x in $\mathfrak{h}$ bound to the same quantifying variable x by $\mathfrak{r}$.

Any sentence $\mathfrak{s}$ can be called STANDARD if and only if every sub-sentence of $\mathfrak{s}$ is a sentence of the form $\mathfrak{p}(: x, \ldots, z)$. Inference rules for quantifications can be stated as follows:

(a) For any sentence $\mathfrak{p}$, $(x)\mathfrak{p}$ holds if $(x)\mathfrak{p}(u, u :)$ holds for every u (*i.e.* $\forall u!$).

(b) For any *standard*[9] sentence $\mathfrak{p}$, any variable u, and any sentence $\mathfrak{r}$, including the case where $\mathfrak{r}$ is expressed by a single variable, $(x)\mathfrak{p}$ implies $(x)\mathfrak{p}(u, \mathfrak{r} :)$ as far as neither u nor any free variable in $\mathfrak{r}$ occurs in $\mathfrak{p}$ as bound variables.

(c) For any sentence $\mathfrak{p}$, any variable u, and any *standard*[10] sentence $\mathfrak{r}$, including the case where $\mathfrak{r}$ is expressed by a single variable, $(\exists x)\mathfrak{p}(u, \mathfrak{r} :)$ implies $(\exists x)\mathfrak{p}$ as far as neither u nor any free variable in $\mathfrak{r}$ occurs in $\mathfrak{p}$ as bound variables.

(d) If we assume $(\exists x)\mathfrak{p}$, we can take a new variable u for which $(\exists x)\mathfrak{p}(u, u :)$ can be assumed *i.e.* $\exists u!$ $(\exists x)\mathfrak{p}(u, u :)$.

(2) Formalism FLOR for the reinforced primitive logic LOR

Symbols of **FLOR** are the same as that of **FLO**. Namely, only one class of VARIABLES are used together with the HEAD- and TAIL-BRACKETS ("[" and "]") and the COMMA. Any sequence of variables is called INDEX. Any sequences of variables and brackets is called SENTENCE as far as it contains at least one bracket. Any sequence of variables, brackets, and commas is called PROOF as far as it contains at least one comma.

---

9), 10) Compare Foot-note 7) and 8).

Any proof $\mathfrak{p}$ is divided into a sequence $\mathfrak{f}$ of indices and sentences by commas just as in **FLO**, and we can normalize the proof by deleting some commas and also some symbols at the top. We can define INTRODUCTORY INDICES in the sequence $\mathfrak{f}$, and we can divide the sequence again by its introductory indices into a sequence $\mathfrak{l}$ of LINES, each line beginning with an introductory index except for the first line. We regard the first line as beginning with the null-sequence introductory index. All these can be done just as in **FLO**.[11]

Thus far, we can proceed in parallel with **FLO**. We have to change our notions, however, concerning FREE and BOUND VARIABLES, because we would like to introduce QUANTIFICATIONS with respect to predicate variables. Also, we have to change our notion FORMAL EQUIVALENCE, because we have taken

$$x[---]yz[---][---]t[---]uvw,$$

for example, as

$$(x[---]yz[---][---]t[---])(uvw)$$

in **FLO**, but now in **FLOR**, we take it as

$$(x[---])(y(z[---][---])t[---])(uvw),$$

parentheses here being employed as auxiliary meta-logical symbols.

To explain more formally, we denote the i-th term (a symbol or a symbol series) of a sequence $\mathfrak{a}$ by $\mathfrak{a}_i$ and the sub-sequence from its i-th term to its j-th term by $\mathfrak{a}_{ij}$.

Now, let $\mathfrak{s}$ be a sentence as a sequence of symbols. Then, $\mathfrak{s}$ is called NORMAL if and only if the number of head-brackets in $\mathfrak{s}$ is equal to the number of tail-brackets in it and the number of head-brackets in $\mathfrak{s}_{1i}$ is no less than the number of tail-brackets in $\mathfrak{s}_{1i}$ for every i. We can normalize any sentence by adding some head-brackets at its top and some tail-brackets at its end. In any normal sentence, any head- (tail-) bracket is coupled with a partner tail- (head-) bracket, head-brackets always preceding their partner tail-brackets in the sentence.

Any variable $\mathfrak{s}_i$ in a sentence $\mathfrak{s}$ is called HEAD-VARIABLE (or TAIL-VARIABLE) if and only if it is immediately followed by a head- (tail-) bracket

11) See Ono [1].

in "$\mathfrak{s}$" skipping over variables. Thus far is quite the same as in **FLO**.

Any-head-variable in $\mathfrak{s}$ except those following just after a tail-bracket in "$\mathfrak{s}$" is called QUANTIFYING VARIABLE. Any tail-variable in $\mathfrak{s}$ following just after a head-bracket in "$[\mathfrak{s}$" is called PREDICATE-VARIABLE. Any variable which is neither quantifying variable nor predicate variable is called OBJECT-VARIABLE.[12]

To divide a normal sentence $\mathfrak{s}$ into three blocks as shown in the preceeding example, the FIRST BLOCK $\mathfrak{s}_{1i}$ is defined by that i is the smallest number satisfying the following conditions: $\mathfrak{s}_i$ is a tail-bracket, $\mathfrak{s}_{i+1}$ is not a head-bracket, and $\mathfrak{s}_{1i}$ is a normal sentence. The SECOND BLOCK $\mathfrak{s}_{i+1,j}$ is defined by that $\mathfrak{s}_j$ is the last tail-bracket in $\mathfrak{s}$. The first block of any sentence can never be void, while the second and third blocks may be void.

The second block is usually divided into its sub-blocks, each consisting of an object variable at its top and a normal sentence following after it. Namely, the first block of the sentence $\mathfrak{s}_{h+1,j}$ is the $(k+1)$-th sub-block of the second block of $\mathfrak{s}$ assuming that the k-th sub-block of the second block of $\mathfrak{s}$ ends with $\mathfrak{s}_h$ $(h<j)$.

As we wish that the range of any quantifying variable in a block or in a sub-block should not extend over the end of the block or sub-block, I define the notion "BOUND TO" as follows: Let $\mathfrak{s}_i$ be a quantifying variable and $\mathfrak{s}_j$ be a variable after $\mathfrak{s}_i$ $(i<j)$ in $\mathfrak{s}$. Then, $\mathfrak{s}_j$ is said to be BOUND TO $\mathfrak{s}_i$ if and only if

1) $\mathfrak{s}_j$ is not a quantifying variable and $\mathfrak{s}_j$ is denoted by the same letter as $\mathfrak{s}_i$.

2) For any tail-bracket $\mathfrak{s}_k$ in $\mathfrak{s}_{ij}$ $(i<k<j)$, the number of head-brackets in $\mathfrak{s}_{i,k+1}$ really exceeds the number of tail-brackets in it.

3) No quantifying variable $\mathfrak{s}_h$ in $\mathfrak{s}_{ij}$ of the same letter as $\mathfrak{s}_i$ satisfies the above condition for $\mathfrak{s}_i$.

Any variable in a sentence is called BOUND if and only if it is bound to some quantifying variable in the sentence. Any variable in a sentence is called FREE if and only if it is not BOUND.

---

12) In **FLO**, quantifying variables are dealt with as if they are object variables. In **FLOR**, however, they are dealt with as if they are variables of both characters, object-like as well as predicate-like. See Ono [1].

The following example would be a help to understand our rules. To make comparison easier, I dealt with the same normal sentence in **FLOR** and in **FLO**. Arrows show the BOUND-TO relation of variables. Single underline indicates the second block of the whole sentence. Q, P, and O under the variables indicate that the variables just above them are quantifying variables, predicate variables, or object variables, respectively.

In **FLOR**: $xy[x[xy][yx]y[xx][yy]xy]\underline{xy[xy][yx]x[xy]}yxy$

$QQ[Q[PO][PO]O[PO][PO]OO]OQ[PO][PO]O[PO]OOO$

In **FLO**: $xy[x[xy][yx]y[xx][yy]xy]xy[xy][yx]x[xy]\underline{yxy}$

$QQ[Q[PO][PO]Q[PO][PO]OO]QQ[PO][PO]Q[PO]OOO.$

Now, I will define formally BRACKET-TRANSFORMATION, SUBSTITUTION, and FORMAL EQUIVALENCE of sentences as follows:

1) If $\mathfrak{a}$ is a sentence, any one of "$\mathfrak{a}$", "$\mathfrak{a}]$", and "$[\mathfrak{a}$" can be transformed to each other. If p is a predicate variable in a sentence $\mathfrak{a}p\mathfrak{b}$, the sentence "$\mathfrak{a}p\mathfrak{b}$" can be transformed into "$\mathfrak{a}[p]\mathfrak{b}$", and *vice versa.* If $\mathfrak{a}$ is a series of quantifying variables and $\mathfrak{b}$ is a non-void series of normal sentences of the form $[\cdots]$[13], any sentence of the form "$\mathfrak{c}\mathfrak{a}\mathfrak{b}\mathfrak{d}$" can be transformed into the sentence "$\mathfrak{c}[\mathfrak{a}\mathfrak{b}]\mathfrak{d}$" and *vice versa,* unless $\mathfrak{d}$ begins with a head-bracket. These transformations are called BRACKET TRANSFORMATION.

2) Let $\mathfrak{z}$ be a series of normal sentences of the form $[\cdots]$, and $x\mathfrak{a}y\mathfrak{z}\mathfrak{b}$ be a normal sentence whose first block is $x\mathfrak{a}$ and whose second block has $y\mathfrak{z}$ as its first sub-block. Then any sentence of the form "$\mathfrak{c}[x\mathfrak{a}y\mathfrak{z}\mathfrak{b}]\mathfrak{d}$" can be transformed into "$\mathfrak{c}[\mathfrak{f}\mathfrak{b}]\mathfrak{d}$" and *vice versa*, where $\mathfrak{f}$ is the sentence obtained from $\mathfrak{a}$ on replacing every object variable x and every predicate variable x of $\mathfrak{a}$ bound to the quantifying variable x standing at the top of $x\mathfrak{a}$ by y and $\mathfrak{z}$, respectively, assuming that the variable y as well as any free variables in $\mathfrak{z}$ do not occur in $\mathfrak{a}$ as bound

13) Expressions of this kind sound surely ambiguous. To speak more exactly, we should say that the sentence is of the form $[\cdots]$ and the brackets at both ends are coupled together.

variables. These transformations are called SUBSTITUTIONS.

3) Two sentences are called FORMALLY EQUIVALENT if and only if the one can be transformed into the other by a finite number of steps of BRACKET TRANSFORMATIONS and SUBSTITUTIONS.

Any normal sentence $\mathfrak{s}$ is called STANDARD if and only if every sequence of symbols between any coupled head- and tail-brackets in "[$\mathfrak{s}$]" is a sentence of the form $\mathfrak{abc}$, $\mathfrak{a}$ and $\mathfrak{c}$ being series of variables and $\mathfrak{b}$ being a series of sentences of the form [---].

Now, let $\mathfrak{p}$ be any proof arranged in a series of LINES, each line beginning with an introductory index except for the first line. Each introductory index is followed by a single comma or double commas according as it refers to sentences of assertion character or of assumption character. We attach the null series index followed by a single comma to the first line. We assume a certain order between letters for variables. For any index $\mathfrak{a}$ of the length n, the class of all the indices of the form $\mathfrak{a}_{1i}l$ $(i<n)$, l standing before $\mathfrak{a}_{i+1}$ with respect to the order of letters, called GROUND[14] of $\mathfrak{a}$. Also, the class of all the indices of the form $\mathfrak{a}l$ is called FRAME-WORK[15] of $\mathfrak{a}$.

Each line can be considered as consisting of three parts, an introductory index at its top, a series of sentences coming next to it, and lastly a series of indices which we call REFERENCE INDICES.

Any proof $\mathfrak{p}$ is called VALID in **FLOR** if and only if it satisfies the following conditions:

1) Lines are arranged in the lexicographic order of their introductory indices with respect to the order of letters.

2) Any line having an introductory index $\mathfrak{a}$ followed by double commas has no reference indices at its end, and the frame-work of $\mathfrak{a}$ is vacant.

3) Every reference indices in any line beginning with an introductory index $\mathfrak{a}$ followed by a single comma and having vacant frame-work is an index in the ground of $\mathfrak{a}$. For any sentence $\mathfrak{s}$ in the line, such reference indices $\mathfrak{c}, \ldots, \mathfrak{e}$, and $\mathfrak{f}$ can be found out in the line that: firstly, a *standard*[16] sentence $\mathfrak{xp}\cdots\mathfrak{rt}$ is in the line beginning with $\mathfrak{f}$, $\mathfrak{x}$ being a series of variables, $\mathfrak{p}, \ldots, \mathfrak{r}$, and $\mathfrak{t}$ being normal sentences of the form [---], and $\mathfrak{s}$ is formally equivalent to a

14), 15) See Ono [1] and [3]. Ground of $\mathfrak{a}$ has been originally called BASIS of $\mathfrak{a}$ in [3].
16) Compare Foot-notes 7), 8), 9) and 10).

normal sentence of the form $\mathfrak{x}\mathfrak{t}\mathfrak{y}$, $\mathfrak{y}$ being a series of normal sentences of the form $\mathfrak{z}[\cdots]$, no free variable of $\mathfrak{y}$ occurring in $\mathfrak{x}\mathfrak{p}\cdots\mathfrak{r}$ as bound variable; secondly any one of $\mathfrak{x}\mathfrak{p}\mathfrak{y}, \ldots, \mathfrak{x}\mathfrak{r}\mathfrak{y}$ is formally equivalent to a sentence in some line beginning with one of $\mathfrak{c}, \ldots, \mathfrak{e}$.

4) Reference indices of any line beginning with an introductory index $\mathfrak{a}$ having non-void frame-work are two indices $\mathfrak{a}h$ and $\mathfrak{a}k$ (possibly coincide) such that $\mathfrak{a}h$ is the only introductory index in the frame-work of $\mathfrak{a}$ which is followed by double commas. In the line beginning with $\mathfrak{a}$ lies only one sentence which is formally equivalent to a sentence of the form $\mathfrak{x}\mathfrak{p}\cdots\mathfrak{r}\mathfrak{t}$, $\mathfrak{x}$ being a series of variables of the length n and $\mathfrak{p}, \ldots, \mathfrak{r}$, and $\mathfrak{t}$ being normal sentences of the form $[\cdots]$. The series of sentences in the line beginning with $\mathfrak{a}h$ are sentences of the form $\mathfrak{x}\mathfrak{p}\mathfrak{y}, \ldots, \mathfrak{x}\mathfrak{r}\mathfrak{y}$, where $\mathfrak{y}$ is a series of symbols of the form $\mathfrak{z}_1[\mathfrak{z}_1]\cdots\mathfrak{z}_n[\mathfrak{z}_n]$ for series $\mathfrak{z}$ of mutually distinct n variables which do not occur in any sentence in the lines beginning with $\mathfrak{a}$ or with an index belonging to the ground of $\mathfrak{a}$. In the line beginning with $\mathfrak{a}k$, lies a sentence formally equivalent to the sentence $\mathfrak{x}\mathfrak{t}\mathfrak{y}$.

Essentially, the logic **FLOR** introduced in this way can be regarded as a formalism for **LOR**, although I do not go through detailed discussion on the matter.

(3) Concluding remarks

In the future development of reinforced logics, one of the most important tasks must be to prove the conjecture that the reinforced logic **LXR** for any **J**- or **K**-series logic **LX** can be faithfully interpreted in the reinforced primitive logic **LOR**. Although this conjecture is highly plausible, it is not trivial at all. In fact, it is not clear at present even how to define FAITHFUL INTERPRETATION of a logic in a reinforced logic.

Most remarkable merit of reinforced logics would be that the word "FINITE STEPS OF" can be taken up into our vocabulary, which is so familiar in mathematical reasoning. For example, to express that y can be attained by stepping forward from x in a finite number of steps of procedure from u to v satisfying the relation $R(u, v)$, we have only to take up the expression

$$(P)((u)(v)(R(u, v) \to (P(u) \to P(v))) \to (P(x) \to P(y))).$$

Needless to say, reasonings of this kind are essentially the same as reasonings employing complete induction.

As for the restriction that some inference rules can be used only for sentences being equivalent to some standard sentences, we have no need to worry about much. I dare not say that reinforced logics can cover such extensive theories as set or class theories. However, I believe, the logics can be recognized very useful for establishing popular mathematical theories such as number theory or even analysis.

References

[1] Ono, K. [1] A formalism for primitive logic and mechanical proof-checking, Nagoya Math. J., vol. **26** (1966), 195–203.

[2] Ono, K. [2] On universal character of the primitive logic, Nagoya Math. J., vol. **27-1** (1966), 331–353.

[3] Ono, K. [3] A certain kind of formal theories, Nagoya Math. J., vol. **25** (1965), 59–86.

[4] Takeuti, G. [1] On a generalized logic calculus, Jap. J. Math., vol. **23** (1953), 39–96 .

[5] Whitehead, A. N. and Russell, B. [1] Principia mathematica I–III, 2nd ed. Cambridge (1925–1927).

Mathematical Institute
Nagoya University