Nomalizing Procedure of Sequences over Ordinary Logical Constants

By Hiroakira ONO

Department of Physics, Faculty of Science, University of Tokyo

(Communicated by A. Nozaki)

(Received February 1, 1967)

1. Introduction

In the ordinary formalism of logic, an arbitrary sequence of primitive symbols such as $A\neg\vee B$ or $C\wedge\exists D$ do not always make sense as a logical formula. We must therefore define a class of "well-formed formulas" to exclude these senseless sequences among the whole class of sequences of primitive symbols. On the other hand, K. Ono proposed a new powerful formalism for a primitive logic LO, after which all sequences of primitive symbols can be regarded as well-formed[1].

The above-mentioned logic LO, introduced by K. Ono himself, is an intuitionistic predicate logic having only two logical constants "implication" and "universal quantifier". Primitive symbols in the formalism are head- and tail-brackets ([,]) together with symbols in an infinite list $\{x, y, -\}$. These symbols in the same list are used to denote both predicate variables and individual variables. For example, a formula $R(x, y)$ in the ordinary formalism is denoted by $[rxy]$. The formulas representing "A implies B" and "for all x, $R(x)$ holds" are denoted by $[a][b]$ and $x[rx]$, respectively.

Theoretically, this formalism for LO seems sufficient to represent many logical systems, since the lower classical predicate logic LK as well as the intuitionistic predicate logic LJ, etc. is interpreted faithfully in LO[2]. For practical purposes, however, it would be much more convenient if we can use other logical constants such as "conjunction", "disjunction", "negation", "existential quantifier", etc..

In this paper, we shall show another formalism in which these logical constants are allowed as primitive symbols and still any sequences of primitive symbols are construed as representing some logical propositions. More precisely, we shall construct an algorithm to transform any sentence (a finite sequence of primitive symbols) into a normal sentence (a well-formed formula in the ordinary sense.)

The following examples may be helpful in understanding the main difference of our formalism from ordinary ones. The primitive symbols of our formalism are head- and tail-brackets ([,]), negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), existential quantifier ($\exists$) and an infinite list of variables[3]. In the following list, we shall illustrate the interpretation of the NORMAL sentences in our formalism.

1.	$[rx_1x_2—x_n]$ (r and x_i's are variables.)	: predicate $R(x_1, x_2, —, x_n)$.
2.	$[\ ]$	: any fixed proposition.
3.	$[\neg[A]]$	: not "A". ("A" denotes the interpretation of A.)
4.	$[x[A]]$	: For any x, "A" holds.
5.	$[\exists x[A]]$	: There exists x such that "A" holds.
6.	$[\vee[[A_1][A_2]—[A_n]]]$	: "A_1" or "A_2" or—or "A_n".
6.1.	$[\vee[[A]]]$	: "A".
7.	$[\wedge[[A_1][A_2]—[A_n]]]$	: "A_1" and "A_2" and—and "A_n".
7.1.	$[\wedge[[A]]]$	: "A".
8.	$[[A_1][A_2]—[A_n][A]]$	: "A" is deducible from "A_1", "A_2", $\cdots$ and "A_n".
9.	$[x[A]y]$	: formula obtained by substituting y for x in "A".

For instance, $(x)(p\supset F(x))\supset(p\supset(x)F(x))$ in the ordinary formalism is expressed as $[[x[[p][fx]]][[p][x[fx]]]]$. $[xx]$ denotes a predicate $X(x)$. $[x[xx]]$ and $[x[xx]y]$ denote a predicate meaning "for all x, $X(x)$" and a predicate obtained by substituting y for x in $X(x)$, respectively. $[x[\ ]]$ denotes "for all x, A^* holds". (A^* denotes any fixed proposition.)

2. Normal Form of a Sentence

2.1. Proper and Normal Sentences

We define PROPER and NORMAL sentences. Hereafter, the following notations are adopted for convenience.

1) variables: x, y, z, u (u denotes any fixed variable.).
2) meta-logical variables standing for any single primitive symbol or the null sequence: a, b, c, d, a_i, b_i .
3) meta-logical variables for any sentence (any sequence of symbols): A, B, C, D, A_i, B_i .
4) the null sentence: $\sharp$.

In any sentence of the form $AxB]C$, the variables x is called a TAIL-VARIABLE if B consists of variables only. Any variable in a sentence which is not a tail-variable is called a HEAD-VARIABLE. We will next explain the notion "pairing of head- and tail-brackets in a sentence", which is used in Definition 1. Any pair of a head- and a tail- brackets in a sentence is called PAIRED if and only if it satisfies the following conditions: 1) The head-bracket stands before the tail-bracket. 2) The number of head-brackets between them is equal to the number of tail-brackets between them. 3) If the sentence between them is of the form $a_1a_2—a_m$, then in any sentence $a_1a_2—a_i$ for each $1\leqq i\leqq m$, the number of head-brackets is not smaller than the number of tail-brackets.

If a head-bracket c and a tail-bracket d are paired, then c (or d) is called the PARTNER of d (or c).

DEFINITION 1. Any sentence $a_1a_2—a_n$ is called PROPER if and only if it satisfies the following conditions.

1.1. $a_1=[$ and $a_n=]$.
1.2. a_1 and a_n are paired.
2.1. If a_i is $\neg$, then a_{i-1} and a_{i+1} are head-brackets.
2.2. If a_i is $\wedge$ or $\vee$, then a_{i-1}, a_{i+1} and a_{i+2} are all head-brackets.
2.3. If a_i is a head-variable, then a_{i-1} is $\exists$ or [, and a_{i+1} is [.
2.4. If a_i is $\exists$, then a_{i-1} is [, a_{i+1} is a variable, and a_{i+2} is [.
3.1. If a_i is $\neg$, and $a_{i+1}=[$ and $a_k=]$ are paired, then a_{k+1} is a tail-bracket whose partner is a_{i-1}.
3.2. If a_i is $\wedge$ or $\vee$, and $a_{i+1}=[$ and $a_k=]$ are paired, then a_{k+1} is a tail-bracket whose partner is a_{i-1}.
3.3. If a_i is a head-variable, a_{i-1} is [, and $a_{i+1}=[$ is the partner of $a_k=]$, then either a_{k+1} is a tail-bracket whose partner is a_{i-1}, or a_{k+1} is a variable and a_{k+2} is a tail-bracket whose partner is a_{i-1}.
3.4. If a_i is $\exists$ and $a_{i+2}=[$ and $a_k=]$ are paired, then a_{k+1} is a tail-bracket whose partner is a_{i-1}.
4. If $a_i=[$, $a_k=]$ and a_i and a_k are paired, then either both a_{i-1} and a_{k+1} are variables, or a_{k+1} is [or].

Any pair of brackets in a proper sentence are really paired in the ordinary sense. Obviously, any bracket in a proper sentence has its partner.

DEFINITION 2. Any sentence is called NORMAL if and only if it is proper and satisfies the following condition 5.

5. If $a_i=[$, $a_{i+1}=[$, and a_{i+1} and a_k are paired, then $a_{k+1}\neq]$ unless a_{i-1} is $\wedge$ or $\vee$ and $a_{i+2}\neq[$.

For instance, $[\vee[[xy][zu]]]$ is normal. $[[\vee[[xy][zu]]]]$ and $[\vee[[[xy][zu]]]]$ are proper but not normal. $\vee[[xy][zu]]$ and $[\vee[xy][zu]]$ are not proper. Roughly speaking, a normal sentence is a kind of proper sentence which contains no redundant pairs [[——]]. We denote normal sentences as $\bar{A}$, $\bar{B}$, $\bar{D}$, etc..

2.2. Transformations

We introduce here the list of transformations. Suppose that there is a transformation $C_1\longleftrightarrow C_2$ in the list and that a sentence AC_1B is given. If A and B satisfy the conditions of transformation $C_1\longleftrightarrow C_2$, then AC_1B can be transformed into a sentence AC_2B. We denote this as $AC_1B\longrightarrow AC_2B$. We define that every transformation is symmetric, *i.e.*, $C_1\longleftrightarrow C_2$ implies $C_2\longleftrightarrow C_1$. Hence, $A\longrightarrow B$ implies $B\longrightarrow A$. If $A_1\longrightarrow A_2$, $A_2\longrightarrow A_2$,—— $A_{n-1}\longrightarrow A_n$, then we denote this as $A_1\Longleftrightarrow A_n$. Obviously, transformability $\Longleftrightarrow$ is an equivalence relation. In the following list, we assume $a\neq[$ and $b\neq]$.

1.1. $\neg a\longleftrightarrow\neg[a$.

1.2. $ca \longleftrightarrow c[[a$,

$c[a \longleftrightarrow c[[a$ (for $c=\wedge, \vee$).

1.3. $\exists xa \longleftrightarrow \exists x[a$,

$\exists[\longleftrightarrow \exists u[$ (u is a fixed variable),

$\exists c \longleftrightarrow \exists[c$ (for $c=\neg, \wedge, \vee, \exists,]$).

2. $ac \longleftrightarrow a[c$ (for $c=\neg, \wedge, \vee, \exists$).

3. If there is a tail-bracket in $\#A]$ which has no partner, then $\#A] \longleftrightarrow \#[A]$. If there is a head-bracket in $[A\#$ which has no partner, then $[A\# \longleftrightarrow [A]\#$.

4. If $a_1=[$, $a_n=]$, and a_1 and a_n are not paired, or if $a_1 \neq [$or $a_n \neq]$, then $\#a_1a_2 —— a_n\# \longleftrightarrow \#[a_1a_2 —— a_n]\#$.

5. If $[A]$ is proper, then

5.1. for $C=x, \exists, \exists x$ (If $C=x$, then $a\neq\exists$ and b is not a variable.),

$aC[A]b \longleftrightarrow a[C[A]]b$,

$[CA]b \longleftrightarrow [C[A]]b$,

$aC[A]] \longleftrightarrow a[C[A]]$.

5.2. if $a\neq\exists$, then

$ax[A]yb \longleftrightarrow a[x[A]y]b$,

$[x[A]yb \longleftrightarrow [x[A]y]b$,

$ax[A]y] \longleftrightarrow a[x[A]y]$.

5.3. $a\neg[A]b \longleftrightarrow a[\neg[A]]b$,

$[\neg[A]b \longleftrightarrow [\neg[A]]b$,

$a\neg[A]] \longleftrightarrow a[\neg[A]]$.

5.4. for $c=\wedge, \vee$,

$ac[B]b \longleftrightarrow a[c[B]]b$,

$[c[B]b \longleftrightarrow [c[B]]b$,

$ac[B]] \longleftrightarrow a[c[B]]$.

(If A is of the form $a_1a_2 - a_n$ and $a_1=[$, then $B=A$. Otherwise $B=[A]$.)

Suppose that any given sentence is of the form CAD. Then any one of the transformations 5 can be applied to A, only after all transformations 5 are applied to C as far as possible.

6. If $[A]$ is proper and c is not a variable, then

$$c[A]x \longleftrightarrow c[A][x .$$

When this transformation is applied to any sentence B in the direction from left to right (*i.e.*, $c[A]x \longrightarrow c[A][x)$, it must be applied only after transformations 5 is applied to B from left to right.

7. If A is proper, then $[A] \longleftrightarrow A$ unless A is of the form $[a_1a_2 - a_n]$, where $a_1 \neq [$ and either $\wedge$ or $\vee$ stands just before A.

If x is a tail-variable in any sentence A, then x is a tail-variable also in the sentence A' which is obtained from A by any one of these transformations[6].

We introduce some notations about transformations. $T(i)$ (or $T'(i)$) means applying the i-th transformation from left to right (or from right to left) as

far as possible. $T_1(i)$ means applying the i-th transformation from left to right only once. $T(i)T(j)$ denotes the combined operation of $T(i)$ and $T(j)$ (first $T(j)$ and next $T(i)$). $T(i_m)-T(i_1)=T(j_n)-T(j_1)$ means that for any sentence A, the sentence obtained from A by operating $T(i_m)-T(i_1)$ is the same sentence obtained from A by operating $T(j_n)-T(j_1)$. Obviously, $T(k_p)-T(k_1)T(i_m)-T(i_1)=T(k_p)-T(k_1)T(j_n)-T(j_1)$ for any $T(k_p)-T(k_1)$ as far as $T(i_m)-T(i_1)=T(j_n)-T(j_1)$. We show next an example of these transformations. Sentence $\exists[x[zx$ is transformed into $\exists u[x[zx$ by 1.3 (we express it as $\exists[x[zx \xrightarrow{1.3} \exists u[x[zx$), and $\exists u[x[zx \xrightarrow{2} [\exists u[x[zx \xrightarrow{3} [\exists u[x[zx] \xrightarrow{3} [\exists u[x[zx]]$
$\xrightarrow{3} [\exists u[x[zx]]]$.
So $\exists[x[zx$ is transformed into a normal sentence $[\exists u[x[zx]]]$. In the next section 2.3, we will show that any sentence is transformed uniquely into a normal sentence, independently of the order of transformations applied.

2.3. Normalizing Process and Normal Form

In this section, we prove our main theorem that any sentence can be transformed into a normal sentence. We define an operation NP $= T(7)T(3)T(6)T(5)T(4)T(3)T(2)T(1)$ and call NP, NORMALIZING PROCESS. It can be shown that NP yields really normal sentence. In the following, $C(i)$ denotes the i-th condition of a normal sentence.

LEMMA 1. Normalizing process without $T(7)$ makes any sentence proper.

LEMMA 2. Normalizing process makes any sentence normal.

Proof. We will show that every condition of proper (or normal) sentence is satisfied after we transform any given sentence by $T(3)T(6)-T(1)$ (or NP). Notice that once any condition $C(j)$ holds at a part of a given sentence, whatever operation $T(i)$ we may apply, $C(j)$ also holds at that part unless $i=6$. Because if $i\neq 7$, then each transformation $T(i)$ only adds some symbols to a given sentence, and each condition $C(j)$ is concerned with either some symbols before or after a symbol such as $\exists$, or head-(or tail-) part of a given sentence, or before or after a proper part[7].

By $T(1,1)$ and $T(2)$, $C(2.1)$ holds and by $T(1.2)$ and $T(2)$, $C(2.2)$ holds. By $T(1.3)$ and $T(2)$, $C(2.4)$ holds. By $T(3)$, every bracket has a partner. By $T(6)$, $C(1.1)$ holds. Extending proper part by applying $T(5)$ successively, we can obtain at last a sentence in which $C(2.3)$ and $C(3)$ hold. Next we apply $T(6)$. In this step, if any given sentence is of the form $C[A]xB$, then B consists of only variables and some tail-brackets following these variables. Because if B contains $\neg$ or $\wedge$ or $\vee$ or $\exists$, then by $T(2)$, B must contain head-brackets, and if B contains head-brackets, then $C[A]xB \longrightarrow C[A][xB'$ by $T(5)$. So B must consist of only variables followed by some tail-brackets. Now $C[A]xB \longrightarrow C[A][xB$ by $T(6)$. (Notice that we have only to apply $T(6)$ once.) $C[A][xB$ is not proper. So we must apply $T(3)$ once more. Then $C(4)$ and $C(1.1)$ hold. We have not applied $T(7)$ yet and can see that all conditions from $C(1)$ to

$C(4)$ are satisfied. Thus Lemma 1 holds. Let us now apply $T(7)$, then it is easily shown that $C(5)$ holds. Hence Lemma 2 holds.

Theorem

a) For any sentence A, there exists at least one normal sentence $\bar{B}$ such that $A \Longleftrightarrow \bar{B}$.

b) For any pair of normal sentences $\bar{B}$ and $\bar{C}$ satisfying $A \Longleftrightarrow \bar{B}$ and $A \Longleftrightarrow \bar{C}$, $\bar{B} = \bar{C}$ holds.

Proof.

a) By Lemma 2, we can make any sentence normal.

b) Let us assume $A \Longleftrightarrow \bar{B}$ and $A \Longleftrightarrow \bar{C}$. Then $\bar{B} \Longleftrightarrow \bar{C}$ and there is a sequence $(A_1, A_2, -, A_n)$ such that $A_1 = \bar{B}$, $A_n = \bar{C}$ and for $1 \leqq i \leqq n-1$ A_i is transformed into A_{i+1} by applying only one transformation. Let $\bar{D}$ be the normal sentence obtained from D by NP. Now we will prove $\mathrm{NP} = \mathrm{NP}T_1(i) = \mathrm{NP}T_1'(i)$ for any i. If we succeed in it, then we can also prove $\bar{B} = \bar{C}$. For $\bar{B} = \bar{A}_2$, $\bar{A}_2 = \bar{A}_3$, — and $\bar{A}_{n-1} = \bar{C}$ follow immediately from this fact. (See Fig. 1.)

$$\begin{tikzcd}
\bar{B} \arrow[r, Leftrightarrow] \arrow[d, Leftrightarrow, "\mathrm{NP}"'] & A_2 \arrow[r, Leftrightarrow] \arrow[d, Leftrightarrow] & A_3 \arrow[r, Leftrightarrow] \arrow[d, Leftrightarrow] & - \arrow[r, Leftrightarrow] & A_{n-1} \arrow[r, Leftrightarrow] \arrow[d, Leftrightarrow] & \bar{C} \arrow[d, Leftrightarrow, "\mathrm{NP}"] \\
\bar{B} \arrow[r, phantom, "="] & \bar{A}_2 \arrow[r, phantom, "="] & \bar{A}_3 \arrow[r, phantom, "="] & - \arrow[r, phantom, "="] & \bar{A}_{n-1} \arrow[r, phantom, "="] & \bar{C}
\end{tikzcd}$$

Fig. 1

1. Transformations 1.1, 1.2 and 1.3 are obviously commutative.
   $T(1)T_1(1) = T(1)$ by the definition. Hence, $\mathrm{NP}\,T_1(1) = \mathrm{NP}$.
   For 1.1, $\neg[a \xrightarrow{1.1} \neg a \xrightarrow{1.1} \neg[a$. Hence $T(1)T_1'(1.1) = T(1)$.
   For 1.2 and 1.3, the similar propositions hold. Thus $\mathrm{NP}\,T_1'(1) = \mathrm{NP}$.
2. Let c be $\wedge$ or $\vee$. (See transformation 2.)
   1. $bca \xrightarrow{2} b[ca \xrightarrow{1,2} b[c[[a$,
      $bca \xrightarrow{1} bc[[a \xrightarrow{2} b[c[[a$, so $T(2)T(1)T_1(2) = T(2)T(1)$.
      Hence, $\mathrm{NP}T_1(2) = \mathrm{NP}$.
   2. $b[ca \xrightarrow{2} bca \xrightarrow{1} bc[[a \xrightarrow{2} b[c[[a$,
      $b[ca \xrightarrow{1,2} b[c[[a$. Hence $\mathrm{NP}T_1'(2) = \mathrm{NP}$.
   $\mathrm{NP}T_1(2) = \mathrm{NP}T_1'(2) = \mathrm{NP}$ can be proved similarly in other cases $c = \neg, \exists$.
3. Case 1, where the given sentence A contains no $\wedge, \vee, \exists, \neg$.

Suppose that A is expressed in the form $C]D$, and in $C]$, there is a tail-bracket which has no partner.

$$A \xrightarrow{3} [C]D \xrightarrow{1,2} [C]D \xrightarrow{3} [C']D',$$
$$A \xrightarrow{1,2} A \xrightarrow{3} [C]D \xrightarrow{3} [C']D'. \quad \text{Hence, } \mathrm{NP}T_1(3) = \mathrm{NP}.$$

$\mathrm{NP}T_1(3) = \mathrm{NP}$ can be proved similarly also in the case where A is of the form $C[D$.

Case 2, where A contains at least one $\wedge$ or $\vee$ or $\exists$ or $\neg$.
Suppose that $A=C]D$ and C contains m head-brackets having no partner and n tail-brackets having no partner $(n>0)$. If $m\neq 0$, the tail-bracket in question must have a partner. We consider therefore only the case $m=0$.

$$C]D \xrightarrow{3} [C]D \xrightarrow{1,2} [C'']D''$$

Suppose that C'' contains p more head-brackets which are generated by $T(2)T(1)$, q of them have their partners in C'' and r of them have their partners in $D''(q+r\leqq p$ and $q\leqq n)$.
Let s (or t) be the number of the tail- (or head-) brackets in D'' which have no partners. Then,

$$[C'']D'' \xrightarrow{3} \underbrace{[\cdots}_{n-q+1}[C'']D'']\underbrace{\cdots]}_{p-q-r} \xrightarrow{3} \underbrace{[\cdots}_{n-q+1+s}[C'']D'']\underbrace{\cdots]}_{p-q-r+t}$$

$$\overset{\text{def.}}{=}[b_1b_2-b_n] .$$

On the other hand, if $p=q$, then $C]D \xrightarrow{1,2} C'']D'' \xrightarrow{3} [C'']D''$.
Thus $T(2)T(1)T_1(3)=T_1(3)T(2)T(1)$, therefore $\mathrm{NP}T_1(3)=\mathrm{NP}$.

If $p>q$, then $C]D \xrightarrow{1,2} C'']D'' \xrightarrow{3} \underbrace{[\cdots}_{n-q}[C'']D'']\underbrace{\cdots]}_{p-q-r-1} \xrightarrow{3} \underbrace{[\cdots}_{n-q+s}[C'']D'']\underbrace{\cdots]}_{p-q-r+t-1}$

$$=b_1b_2-b_n .$$

(a) If b_1 and b_n are not paired and the head-bracket before b_1 and the tail-bracket after b_n are paired,
then

$$[b_1-b_n] \xrightarrow{4} [b_1-b_n] , \text{ and } b_1-b_n \xrightarrow{4} [b_1-b_n] .$$

Hence, $\mathrm{NP}T_1(3)=\mathrm{NP}$.
(b) 1. If b_1 and b_n are paired, then

$$[b_1-b_n] \longrightarrow [b_1-b_n]=[[b_2-b_{n-1}]] , \text{ and}$$

$$b_1-b_n \xrightarrow{4} b_1-b_n=[b_2-b_{n-1}] .$$

2. If b_1 and b_n are not paired and the head-bracket before b_1 and the tail-bracket after b_n are also not paired, then

$$[b_1-b_n] \xrightarrow{4} [b_1-b_n]] , \text{ and } b_1-b_n \xrightarrow{4} [b_1-b_n] .$$

Let $[c_1-c_p]$ be either $[b_2-b_{n-1}]$ or $[b_1-b_n]$, then $[[c_1-c_p]] \xrightarrow{5,6,3} [[c_1'-c_q']]$, and $[c_1-c_p] \xrightarrow{5,6,3} [c_1'-c_q]$.
Since $[c_1'-c_q']$ as well as $[[c_1'-c_q']]$ is proper by Lemma 1,

$$[[c_1'-c_q']] \xrightarrow{7} [c_1'-c_q'] . \text{ Hence, } \mathrm{NP}T_1(3)=\mathrm{NR} .$$

$\mathrm{NP}T_1(3)=\mathrm{NP}$ can be proved similarly in the case where A is of the form $C[D$. $\mathrm{NP}T_1'(3)=\mathrm{NP}$ can be proved also similarly.

4. Case 1, where the given sentence $a_1a_2-a_k$ contains n head- and m tail-brackets which have no partner $(m, n>0)$.

$$a_1{-}a_k \xrightarrow{4} [a_1 - a_k] \xrightarrow{1,2} [a_1'{-}a_l'] \xrightarrow{3} \underbrace{[\cdots}_{m-q}[a_1'{-}a_l']\underbrace{\cdots]}_{n+p-q}\,.$$

(Suppose that $[a_1'{-}a_l']$ contains p more brackets than $[a_1{-}a_k]$ and q of them have their partners.)
On the other hand;
If $a_1 \neq \neg, \wedge, \vee, \exists$, then

$$a_1 - a_k \xrightarrow{1,2} a_1' - a_l' \xrightarrow{3} \underbrace{[\cdots}_{m-p}[a_1'{-}a_l']\underbrace{\cdots]}_{n+p-q}\,.$$

Hence, $\mathrm{NP}T_1(4)=\mathrm{NP}$.
If $a_1 = \neg$ or $\wedge$ or $\vee$ or $\exists$, then

$$a_1{-}a_k \xrightarrow{1,2} [a_1' - a_l' \xrightarrow{3} \underbrace{[\cdots}_{m-q+1}[a'{-}a_l']\underbrace{\cdots]}_{n+p+1-q}\,.$$

Now, we can prove $\mathrm{NP}T_1(4)=\mathrm{NP}$ just as we have proved Case 2 of 3.

Case 2, where $m=0$ or $n=0$. $\mathrm{NP}T_1(4)=\mathrm{NP}$ can be proved similarly in this case. $\mathrm{NP}T_1'(4)=\mathrm{NP}$ can be proved similarly.

5. Let c be either $\wedge$ or $\vee$, and B be the proper sentence which is defined in the list of transformations.

$$c[B] \xrightarrow{5} [c[B]] \xrightarrow{1\sim5} [c[B]]\text{ , and } c[B] \xrightarrow{1,2} [c[B] \xrightarrow{3,4,5} [c[B]]\,.$$

Hence, $\mathrm{NP}T_1(5)=\mathrm{NP}$.

$$[c[B]] \xrightarrow{5} c[B] \xrightarrow{1,2} [c[B] \xrightarrow{3,4,5} [c[B]]\text{ , and}$$

$$[c[B]] \xrightarrow{1\sim5} [c[B]]\,.$$ Hence, $\mathrm{NP}T_1'(5)=\mathrm{NP}$.

$\mathrm{NP}T_1(5)=\mathrm{NP}T_1'(5)=\mathrm{NP}$ can be proved similarly in other cases.

6. We will prove $\mathrm{NP}T_1'(6)T(5)=\mathrm{NP}$, since $T(6)$ must be applied after applying $T(5)$. Suppose that the given sentence is of the form $Bdc[A]xc$, $A=a_1a_2{-}a_n$ and $[A]$ is proper.

Case 1, where $c=[\,,\,]$. $\mathrm{NP}T_1(6)T(5)=\mathrm{NP}$ can be proved easily.

Case 2, where $c=\wedge, \vee$.

If $d\neq[\,,\,]$, $\neg$, and $a_1=[$, then
(a) d is not a variable;

$$dc[A]x \xrightarrow{5} d[c[A]]x \xrightarrow{6} d[c[A]][x \xrightarrow{1\sim6} [D[c[A]]][x,\text{ and}$$
$$dc[A]x \xrightarrow{1,2} D[c[A]x \xrightarrow{3\sim5} [D[c[A]]]x \xrightarrow{6} [D[c[A]]][x\,.$$

(Here we define $D=\exists u$ if $d=\exists$, $D=d[$ if $d=\wedge, \vee$.)
(b) d is a variable;

$$dc[A]x \xrightarrow{5,6} [d[c[A]]x] \xrightarrow{1\sim6} [d[c[A]]x]\text{ , and}$$
$$dc[A]x \xrightarrow{1} d[c[A]x \xrightarrow{2\sim5} d[c[A]]x \xrightarrow{5} [d[c[A]]x]\,.$$

Hence, $\mathrm{NP}T_1(6)T(5)=\mathrm{NP}$. The same proposition can be proved in other cases and the case wherein $c=\neg$ or $\exists$. $\mathrm{NP}T_1'(6)=\mathrm{NP}$ can be proved similarly.

7. Let A be proper.

(a) $[A]\xrightarrow{7}A\xrightarrow{1\sim 6}A$, and $[A]\xrightarrow{1\sim 6}[A]\xrightarrow{7}A$.
Hence, $\mathrm{NP}T_1(7)=\mathrm{NP}$.

(b) $A\xrightarrow{7}[A]\xrightarrow{1\sim 6}[A]\xrightarrow{7}A$, and $A\xrightarrow{1\sim 7}A$.
Hence, $\mathrm{NP}T_1'(7)=\mathrm{NP}$.

Q.E.D.

By our Theorem, there exists only one sentence $\bar{B}$ for any sentence A such that $A\Longleftrightarrow\bar{B}$. So, we call this $\bar{B}$ the NORMAL FORM of A. Now, we can say that any sentence has one and only one normal form of it. For example, suppose that a sentence $[[\vee\, xu]x[zy$ is given.

$[[\vee\, xu]x[zy \xrightarrow{1} [[\vee[[xu]x[zy \xrightarrow{2,3} [[\vee[[xu]x[zy]]]] \xrightarrow{4,5} [[\vee[[xu][x[zy]]]]] \xrightarrow{6,3}$

$[[\vee[[xu][x[zy]]]]]$ (proper) $\xrightarrow{7} [\vee[[xu][x[zy]]]]$ (normal). Thus, $[\vee[[xu][x[zy]]]]$ is a normal form of $[[\vee\, xu]x[zy$ and therefore $[[\vee\, xu]x[zy$ means that $X(u)$ or for any x, $Z(y)$ holds[8].

3. Conclusions

In this paper, we have discussed the case in which primitive symbols are [,] , $\neg$, $\wedge$, $\vee$, $\exists$ and an infinite number of variables. If we add "implication ($\supset$)" to primitive symbols, we should express "A implies B" as $[\supset[[A][B]]]$. But in this case it would be necessary for us to construct a transformation so that there exist only two normal sentences between the head-bracket just after " $\supset$ " and its partner. It does not seem to be too difficult to do so.

Our formalism can be used for any logic which has these primitive symbols, such as LK or minimal logic LM. But if we eliminate negation "$\neg$" from primitive symbols, 1.1 of our transformations, 2.1 and 3.1 of our definition of normal sentence, we can obtain a formalism of positive logic LP. In this case, if we regard [ ] as constantly false proposition $\curlywedge$ and define "not A" by "A implies $\curlywedge$", then we obtain another formalism for LM or LK.

If we restrict primitive symbols to [,] and an infinite number of variables, transformations to 3, 4, 5.1, 5.2, 6, 7 and definition of normal sentence to 1, 2.3, 3.3, 4, 5 with obviously necessary modifications, we obtain a formalism for primitive logic LO. Thus, we can say that our normalizing procedure gives a formalism which makes any sentence "well-formed" for most logics.

References

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

[2] ——— , A certain kind of formal theories, *Nagoya Math. J.*, **25** 59-86, (1965).

[3] ——— , On universal character of the primitive logic, *Nagoya Math. J.*, **27** 331-353, (1966).

1) See K. Ono [1], [2]. The author would like to express his gratitude to K. Ono and A. Nozaki for their many suggestions which have helped greatly in preparing this paper.
2) See K. Ono [3].
3) As for "implication", see section 3 of this paper.
4) See section 3 of this paper.
5) When we interpret any given sentence of the form 8, we must verify that it is not the part of the form 6 or 7.
6) The similar proposition does not hold with respect to a head-variable. For instance, $\exists xy$ is transformed into $[\exists x[y]]$. y is a head-variable in $\exists xy$ but a tail-variable in $[\exists x[y]]$.
7) It is easily verified that if a given sentence A is normal, then $T(i)A$ (sentence obtained from A by applying $T(i)$) is equal to A for any i and that if A is proper and $i \neq 7$, then $T(i)A=A$.
8) As for interpretation of normal sentences, refer to section 1.