TSUKUBA J. MATH.
Vol. 12 No. 1 (1988). 97—114

A TWO-STORIED UNIVERSE OF TRANSFINITE MECHANISMS

Dedicated to Professor Shôkichi Iyanaga on his eightieth birthday

By

Mariko YASUGI

Introduction.

In our previous article [4], we developed a theory of *methods* for our specific objective: to clarify the functional structure of ordinal diagrams. This theory was symbolized by HP (hyper-principle) for the reason that it rendered the foundations of a well-ordered structure far beyond ε_0. (For the structure of order type ε_0 we presented CP, the construction principle; see [2]). The theory HP can be regarded, however, as a general theory of *transfinite mechanisms* independent of ordinal diagrams, and it is of quite an interest in so far as a *mechanism* in our notion is a computational system which produces an object combined in one, given as an input a (transfinite) sequence of already produced objects. This mechanism is again disposed into the universe of objects of our concern. This is the idea behind HP.

Incidentally, what were called *methods* previously are here called *mechanisms*. The reason for this as well as a delicate distinction between methods and mechanisms will be explained in the sequel to this paper.

Now, in this article, we propose an extension of HP, symbolized by TM[2] (a *two-storied theory* of *transfinite mechanisms*), which is obtained from HP by allowing substitutions of term-forms of HP for free variables in type-forms, and hence the new universe (of mechanisms) is *two-storied*. That is, the objects in the universe of HP are regarded as the mechanisms belonging to the *first floor*, so to speak, and play the role of parameters to determine type-forms of the mechanisms *upstairs*. The function variables appearing in the original type-forms are then regarded as living in the basement.

In the sequel to this, an application of TM[2] will be presented; an extension of the transfinite definitions in [4] will be interpreted in TM[2].

Received October 17, 1986.

§ 1. First-floor-term-forms.

Let $\boldsymbol{I}$ be a primitive recursive scheme such that for each $l=1, 2, \cdots$, $\boldsymbol{I}(l)$ represents the pair of a set and its order, say $(I_l, <^l)$, which admits an (a concrete) accessibility proof. (See the introduction of [3] for the meaning of the *accessibility proof.*)

DEFINITION 1.1. 1) Language $\mathcal{L}_0(\boldsymbol{I})$. The language $\mathcal{L}_0(\boldsymbol{I})$ is the quantifier-free part of the language of HA, Heyting arithmetic, augmented by function variables. It may contain some special constants.

2) Terms and (quantifier-free) formulas of $\mathcal{L}_0(\boldsymbol{I})$ are defined as usual. Terms may contain function variables.

Note. 1) In subsequent sections, we deal with the cases where there are just two accessible sets I_1 and I_2. We do not lose anything essential by this simplification.

2) We shall constantly refer to Part II of [4], and hence we cite some definitions and their consequences therefrom with the asterisk affixed. For instance, Definition 2.1* stands for Definition 2.1 in Part II of [4]. The theory we are to introduce is, in its essence, an extension of HP which admits substitutions of term-forms (of HP) of *at*-type or *fn*-type for the free variables in type-forms.

3) The terms which are free of function variables are said to be $\mathcal{L}_0(\boldsymbol{I})$-recursive.

DEFINITION 1.2. 1) The language $\mathcal{L}_{tp}(\boldsymbol{I})$ for type-forms is much the same as the language $\mathcal{L}_{tp}$ in Definition 1.1*. It is based on the language of $\mathcal{L}_0(\boldsymbol{I})$-terms and $\mathcal{L}_0(\boldsymbol{I})$-formulas (which are quantifier-free), and the previous T and $\mathfrak{R}$ be replaced respectively by $\mathfrak{R}^1$ and $\mathfrak{R}^2$.

2) Type-forms, the variables in them and their reduction rules are the same as those in the definition cited above, save that $\mathfrak{R}^1$ and $\mathfrak{R}^2$ need some care.

2.1) Let $t=t(\Xi)$ be an expression in the language $\mathcal{L}_{tp}(\boldsymbol{I})$ free of $\mathfrak{R}^1$ and $\mathfrak{R}^2$ with parameter Ξ, let s be a type-form without $\mathfrak{R}^1$ and $\mathfrak{R}^2$, and let i stand for an $\mathcal{L}_0(\boldsymbol{I})$-term of *at*-type which is supposed to be an element of I_1. Define $\boldsymbol{M}(s, t; i)$ and $\boldsymbol{N}(s, t; i)$ as follows.

$\boldsymbol{M}(s, t; o)=s$, where o is the initial element of I_1;

$$\boldsymbol{N}(s, t; i)=\Lambda j C[j<^1 i; \Lambda\xi_1 \cdots \Lambda\xi_t \boldsymbol{M}(s, t; j), ept],$$

where $\xi_1, \cdots, \xi_t$ are some free variables in $\boldsymbol{M}(s, t; j)$ different from j;

$$\boldsymbol{M}(\mathfrak{s},\, \mathfrak{t}\,;\, i)=\mathfrak{t}(\boldsymbol{N}(\mathfrak{s},\, \mathfrak{t}\,;\, i)) \quad \text{if} \quad o<^1 i\,.$$

If $\boldsymbol{M}(\mathfrak{s}, \mathfrak{t}\,; i)$ is a type-form for all i in I_1, then $\mathcal{R}^1[\mathfrak{s}, \mathfrak{t}\,; i]$ is a type-form. Define

$$\boldsymbol{N}(\mathcal{R}^1\,;\, \mathfrak{s},\, \mathfrak{t}\,;\, i)=\Lambda j\mathcal{C}[j<^1 i\,;\, \Lambda\xi_1 \cdots \Lambda\xi_t\mathcal{R}^1[\mathfrak{s},\, \mathfrak{t}\,;\, j],\, ept]\,.$$

$\mathcal{R}^1[\mathfrak{s}, \mathfrak{t}\,; i]$ is accompanied by the reduction rule:

$$\mathcal{R}^1[\mathfrak{s},\, \mathfrak{t}\,;\, i] \Longrightarrow \mathcal{C}[i=o,\, o<^1 i\,;\, \mathfrak{s},\, \mathfrak{t}(\boldsymbol{N}(\mathcal{R}^1\,;\, \mathfrak{s},\, \mathfrak{t}\,;\, i)),\, ept]\,.$$

The reduct will be abbreviated to $\mathcal{R}^1(\mathfrak{s}, \mathfrak{t}\,; i)$. The variables in $\mathcal{R}^1[\mathfrak{s}, \mathfrak{t}\,; i]$ are the ones in $\mathfrak{s}$, $\mathfrak{t}$ and i.

2.2) $\mathcal{R}^2[\mathfrak{r}, \mathfrak{u}\,; k]$ and its reduction rule are defined similarly, but with slight modifications; $\mathfrak{r}$ and $\mathfrak{u}$ be free of $\mathcal{R}^2$ (but possibly with $\mathcal{R}^1$), I_1 and $<^1$ are respectively replaced by I_2 and $<^2$ and k is supposed to range over I_2.

3.3) Let $\mathfrak{s}$ be a type-form and let ϕ be an $\mathcal{L}_0(\boldsymbol{I})$-term (of *at*-type or *fn*-type). Then $\Pi(\mathfrak{s}\,; \phi)$ is a type-form. If $\mathfrak{s}$ is of the form $\Lambda x t(x)$ and ϕ is of the same type as x, then we place the reduction rule;

$$\Pi(\mathfrak{s}\,;\, \phi) \Longrightarrow \mathfrak{t}(\phi)\,.$$

Note. 1) Type-forms of the form $\Pi(\mathfrak{s}\,; \phi)$ (in 2.3) above) are not necessarily meaningful. Such a waste will be adjusted later.

2) The T-type in [4] is a special case of the $\mathcal{R}^1$ here.

3) The treatment of T and $\mathcal{R}$ in [4] was somewhat inarticulate. We are correcting it subsequently.

DEFINITION 1.3. The complexity of a type-form $\mathfrak{r}$, denoted by $d(\mathfrak{r})$ here, is a notion similar to the γ-degree defined in § 11 of [1]. Here we call it simply the degree.

1) We first define an extension of I_p, $*I_p$, $p=1, 2$.

$$I_p^{\sim}=\{i^{\sim}\,;\, i\in I_p\}\,,$$

$$I_p{}^*=I_p\cup I_p^{\sim}\,,$$

$$*I_p=I_p{}^*\cup\{\infty_p\}\,,$$

$$i<^p{}_* i^{\sim}<^p{}_* j \ \text{ for any } i \text{ and } j \text{ in } I_p \text{ with } i<^p j\,,$$

$$\infty_p \text{ is the maximal element in } {}^*I \text{ with respect to } <^p{}_*\,.$$

The order type of $*I$ is thus $2|I|+1$. We assume the elements of $*I_1$ and $*I_2$ are coded so that the two sets of codes are disjoint. Define

$$I_0=*I_1\cup *I_2\,,$$

where the order $<_0$ of I_0 is induced from $<^1{}_*$ and $<^2{}_*$ and, if $i \in *I_1$ and $k \in *I_2$, then $i <_0 k$.

2) Next define $r(\mathfrak{R}^p ; \mathfrak{t})$, the rank of $\mathfrak{R}^p$ in a type-form $\mathfrak{t}$, which is an element of I_0.

2.1) Suppose $\mathcal{C}[(\mathcal{A}), (u)]$ occurs in $\mathfrak{t}$, where A_l is the condition $j <^p i$ and u_l contains $\mathfrak{R}^p[s, t ; j]$, and where $i \in I_p$ and j is either a variable or a numeral, and for the latter case either $j \notin I_p$ or $i \leqq^p j$. Then, for this occurrenc of $\mathfrak{R}^p$, $r(\mathfrak{R}^p ; \mathfrak{t}) = i$.

2.2) Suppose $\mathfrak{R}^p[s, t ; j]$ occurs in $\mathfrak{t}$, where $j \in I_p$ and 2.1) is not the case. Then for this occurrence of $\mathfrak{R}^p$, $r(\mathfrak{R}^p ; \mathfrak{t}) = j^{\sim}$.

2.3) For any occurrence of $\mathfrak{R}^p$ in $\mathfrak{t}$ which does not fit either of above, $r(\mathfrak{R}^p ; \mathfrak{t}) = \infty_p$.

3) Let $\mathfrak{t}_0$ and $\mathfrak{t}$ be type-forms where $\mathfrak{t}$ is a part of $\mathfrak{t}_0$. We define $d(\mathfrak{t} ; \mathfrak{t}_0)$, the degree of $\mathfrak{t}$ relative to $\mathfrak{t}_0$, to be an element of ω^{I_0}. (Let the order of ω^{I_0} be denoted by $<_\omega$.)

3.1) $d(\mathfrak{t} ; \mathfrak{t}_0) = 1$ if $\mathfrak{t}$ is atomic.

3.2) $d(\Lambda x t ; \mathfrak{t}_0) = d(t ; \mathfrak{t}_0) + 1$.

3.3) $d(s \to t ; \mathfrak{t}_0) = d(s ; \mathfrak{t}_0) \# d(t ; \mathfrak{t}_0)$.

3.4) $d(\Pi(s ; \phi) ; \mathfrak{t}_0) = d(s ; \mathfrak{t}_0) + 1$.

3.5) $d(\mathfrak{R}^p[s, t ; i] ; \mathfrak{t}_0) = d(s ; \mathfrak{t}_0) \# \omega^{r(\mathfrak{R}^p ; \mathfrak{t}_0)}$

3.6) $d(\mathcal{C}[(\mathcal{A}) ; (t)] ; \mathfrak{t}_0) = d(t_1 ; \mathfrak{t}_0) \# \cdots \# d(t_{m+1} ; \mathfrak{t}_0)$.

4) Define $d(\mathfrak{t})$ to be $d(\mathfrak{t} ; \mathfrak{t})$.

PROPOSITION 1.1. *If $s \Rightarrow t$ for hyper-types s and t, then $d(s) <_\omega d(t)$.*

PROPOSITION 1.2. *The notion of normality can be defined as in Definition 1.2* except for the general cases of $\Pi(s ; \phi)$, which is to be settled in the subsequent proof. The normalization theorem on hyper-types can be proved by transfinite induction on d. (See Proposition 1.2* and Corollary*.)*

PROOF. Apply Proposition 1.1. In case of $\Pi(s ; \phi)$ where s is not of the form $\Lambda x t(x)$, consider the normal form of s, say s_0, which exists by the induction hypothesis. If s_0 is of the form $\Lambda x t(x)$ and ϕ is of the same type as x, then we define the reduction rule

$$\Pi(s ; \phi) \Longrightarrow t(\phi) ;$$

otherwise $\Pi(s ; \phi)$ will be said to be irrelevant, and regard this itself as normal. By virtue of Proposition 1.1, $d(t(\phi)) <_\omega d(s_0) <_\omega d(s) <_\omega d(\Pi(s ; \phi))$. This settles the adjustment problem in 2) of the note to Definition 1.2.

DEFINITION 1.4. We can define the *objects* of respective non-irrelevant hyper-type (which will be called hyper-mechanisms) as in Definition 1.3* by transfinite induction on the degree. Due to the involvement of $\mathfrak{R}^1$ and $\mathfrak{R}^2$, they are of transfinite character.

DEFINITION 1.5. (See Definition 2.1*.) 1) The language $\mathcal{L}_{tm}(\boldsymbol{I})$ (for term-forms) is $\mathcal{L}_{tp}(\boldsymbol{I})$ augmented by the variable-forms of associated type-form s, X_n^s, for all n and s, and special constant symbols μ, $\mathcal{B}$, λ, Π, $\boldsymbol{C}$.

2) The term-form of a certain type-form, free and bound variables and variable-forms in it and the associated variables (in type-forms) are defined as before.

3) A term-form which does not have associated free variables will be called a hyper-term, and a hyper-term which does not have free variables of variable-forms will be called a hyper-functional.

4) The constructional complexity of a term-form Φ, denoted by $*(\Phi)$ here, is defined as before.

5) For each hyper-functional Φ, we introduce a (functional) symbol Q_Φ (or Q for short), which is to be interpreted to be the object represented by Φ, say J_Φ (which will also be written as J_Q). Q_Φ is *not* an *official* term in the language.

6) The type-form of a term-form Φ will be written as $[\Phi]$.

Note. The variables (of *at*-ype or of *fn*-type) in the original language are to be distinguished from variable-forms of types respectively N_0 and $N_0 \to N_0$, although they are treated the same way in the formations of term-forms.

PROPOSITION 1.3. 1) *Proposition* 2.1* *holds.*

2) *If Φ and Ψ are identical save for some bound variable-forms and, if Q and R respectively correspond to Φ and Ψ, then $J_Q = J_R$ (the same object).*

DEFINITION 1.6. 1) First assignments. Let $\boldsymbol{x} \equiv x_1, \cdots, x_l$ be a finite sequence of distinct variables and let a_k be a closed $\mathcal{L}_0(\boldsymbol{I})$-term of the same type as x_k, $1 \leqq k \leqq l$. Put $\boldsymbol{a} \equiv a_1, \cdots, a_l$. Then

$$\boldsymbol{a}\boldsymbol{x} \equiv (x_1/a_1, \cdots, x_l/a_l)$$

will denote the (first) assignment of $\boldsymbol{a}$ to $\boldsymbol{x}$.

2) Let Φ be any formal object. $\boldsymbol{a}\Phi$ will denote the result of replacement of x_k by a_k in Φ, presuming that x_k be not bound in Φ and that there be no clashes of variables. If $\boldsymbol{x}$ exhaust the free variables in Φ, then $\boldsymbol{a}$ will be said to be complete for Φ.

3) If the Φ above is a type-form ι, then $\boldsymbol{a}\iota$ will become a hyper-type under

a complete assignment $\boldsymbol{a}$.

4) If the Φ in 2) is a term-form, then $\boldsymbol{a}\Phi$ will become a hyper-term under a complete assignment $\boldsymbol{a}$.

5) Second assignments. Let $\boldsymbol{y}\equiv y_1, \cdots, y_m$ be a finite sequence of distinct variable-forms of hyper-types, and let $\boldsymbol{b}\equiv b_1, \cdots, b_m$ be a finite sequence of functional symbols such that b_k is of the same hyper-type as y_k, $1\leqq k\leqq m$. Then

$$\boldsymbol{by}\equiv(y_1/b_1, \cdots, y_m/b_m)$$

will denote the (second) assignment of $\boldsymbol{b}$ to $\boldsymbol{y}$.

6) Let Φ be any formal object, and let $\boldsymbol{a}$ be complete for Φ. $\boldsymbol{ba}\Phi$ will denote the result of replacement of y_k by b_k in $\boldsymbol{a}\Phi$. If $\boldsymbol{y}$ covers all the free variable-forms in Φ, then $\boldsymbol{b}$ will be said to be complete for $\boldsymbol{a}\Phi$.

7) If Φ is a term-form and $\boldsymbol{a}$ is complete for Φ, then $\boldsymbol{ba}\Phi$ can be defined according to 4) and 6). If $\boldsymbol{b}$ is complete for $\boldsymbol{a}\Phi$, then $\boldsymbol{ba}\Phi$ will become a hyper-functional (in the extended language).

COLLOLLAAY. 1) *Corollary* holds; added is* $\boldsymbol{a}\mathfrak{R}^p[{}_s, {}_t\,; i]=\mathfrak{R}^p[\boldsymbol{a}_s, \boldsymbol{a}_t\,; \boldsymbol{a}i]$.

2) $\boldsymbol{ba}\Pi(\Phi\,;\Psi)=\Pi(\boldsymbol{ba}\Phi\,;\boldsymbol{ba}\Psi)$;
$\boldsymbol{ba}\lambda X\Phi=\lambda\boldsymbol{a}X\boldsymbol{ba}\Phi$; $\boldsymbol{ba}\mathcal{C}[(\mathcal{A})\,;(\Phi)]=\mathcal{C}[\boldsymbol{a}(\mathcal{A})\,;\boldsymbol{ba}(\Phi)]$.

Note. As was noted above, free variables and variable-forms of type N_0 and $N_0\rightarrow N_0$ are distinguished, and so the variables are relevant to first assignments, while variable-forms are relevant to second assignments. $\mathcal{L}_0(\boldsymbol{I})$-terms are of course term-forms, and hence they can be substituted for variable-forms if types are appropriate.

DEFINITION 1.7. 1) CNPR in § 3* (the continuity principle) will be assumed.

2) The interpretation of a term-form Φ at a complete assignment $\boldsymbol{ba}$, $\boldsymbol{J}(\Phi, \boldsymbol{b}, \boldsymbol{a})$, can be defined as in Definition 3.2*.

(1) Closed $\mathcal{L}_0(\boldsymbol{I})$-terms can be interpreted naturally.

(2) Φ is a variable-form X^s. $\boldsymbol{ba}$ determines a functional symbol Q of hyper-type $\boldsymbol{a}_s$. Let $\boldsymbol{J}(\Phi, \boldsymbol{b}, \boldsymbol{a})$ be J_Q.

(3) $\boldsymbol{J}(\Pi(\Phi\,;\Psi), \boldsymbol{b}, \boldsymbol{a})$ can be defined inductively as before.

(4) Consider $\boldsymbol{J}(\lambda X\Phi\,; \boldsymbol{b}, \boldsymbol{a})$, where $[X]$ (the type-form of X)$={}_s$ and $[\Phi]={}_t$, ${}_t$ being free of X. For each Q a functional symbol of $\boldsymbol{a}_s$,

$$M_Q=\boldsymbol{J}(\Phi, (\boldsymbol{b}, \boldsymbol{a}X/Q), \boldsymbol{a})$$

has been defined as a hyper-mechanism of $\boldsymbol{a}_t$. Let M be the hyper-mechanism (of $\boldsymbol{a}_s\rightarrow\boldsymbol{a}_t$) which associates with J_Q for any such Q the M_Q above. Let $\boldsymbol{J}(\lambda X\Phi\,; \boldsymbol{b}, \boldsymbol{a})$ be this M.

(5) Consider $\boldsymbol{J}(\lambda X\Phi\,;\,\boldsymbol{b},\,\boldsymbol{a})$ as above where X occurs (free) in ι. Then $\lambda X\Phi$ is of type-form $\Lambda X\iota$. For each ϕ a closed $\mathcal{L}_0(\boldsymbol{I})$-term of the same type as X, put $\boldsymbol{c}=(\boldsymbol{a},\,X/\phi)$. Then

$$M_\phi=\boldsymbol{J}(\Phi,\,\boldsymbol{b},\,\boldsymbol{c})$$

has been defined as a hyper-mechanism of $\boldsymbol{c}\iota$. Define $\boldsymbol{J}(\lambda X\Phi,\,\boldsymbol{b},\,\boldsymbol{a})$ to be the mechanism M which associates with each ϕ the M_ϕ above. M is of hyper-type $\Lambda X\boldsymbol{a}\iota$.

(6) $\mathcal{C}[(\mathcal{A}),\,(\Phi)]$ can be interpreted as before.

(7) $\Pi(\mathcal{B}\,;\,\mathcal{Z})$ can be defined as before, and from this $\boldsymbol{J}\mathcal{B}$ will be defined. The S in $\mathcal{Z}$ is a variable (of fn-type) in $\mathcal{L}_0(\boldsymbol{I})$. (See (11) in Definition 3.2*.)

PROPOSITION 1.4. *The* $\boldsymbol{J}$ *above is well-defined.*

PROOF. We can follow the proof of Proposition 3.1*. CNPR and an informal reasoning of the bar theorem are used. For the reader's convenience, CNPR (the continuity principle) is written out below.

$$\mathrm{CNPR}(L,\,S):\ \forall S'(S'\restriction ap(L\,;\,S)=S\restriction ap(L\,;\,S)\restriction ap(L\,;\,S')=ap(L\,;\,S)),$$

where L is an arbitrary term-form of type-form $(N_0\to N_0)\to N_0$, S is an arbitrary $\mathcal{L}_0(\boldsymbol{I})$-term of fn-type, S' is a variable of fn-type and $ap(L\,;\,S)$ represents the application of L to S and $S\restriction n$ represents the restriction of S to length n.

DEFINITION 1.8. The objects and expressions defined in this section (type-forms, term-forms and hyper-mechanisms) will be said to first-floor.

§2. Second-floor-term-forms.

DEFINITION 2.1. 1) We assume henceforth the properties of first-floor-objects defined in the first section.

2) Let A be any $\mathcal{L}_0(\boldsymbol{I})$-formula. If B is obtained from A by replacing some free variables by first-floor-term-forms (of appropriate type), then B can be interpreted in the semantics of first-floor-term-forms (see Definition 1.7). We shall call such $B\mathcal{L}_{tm}(\boldsymbol{I})$-recursive.

3) Second-floor-type-forms and the variables and (first-floor-) variable-forms in them as well as the associated variables are defined below. We denote the underlying language of second-floor-type-forms by $2\text{-}\mathcal{L}_{tp}(\boldsymbol{I})$. We shall omit the adjective "second-floor-" when confusion is not likely.

3.1) An expression obtained from a first-floor-type form by replacing some free variables by first-floor-term-forms (of appropriate type) is a second-floor-type-form. As a special case, a first-floor-type-form is a type-form in the extended sense.

3.2) Second-floor-type-forms are closed under the formation rules of first-floor-type-forms except for $\mathcal{R}^1$ and $\mathcal{R}^2$. In $\mathcal{R}^p[s', t'; i']$ as a second-floor-type-form, s', t' and i' are respectively obtained from s, t and i which are first-floor-objects by replacement of some variables by first-floor-term-forms of appropriate type; that is $\mathcal{R}^p[s', t', i']$ is admitted only through 3.1) above. $\mathcal{L}_{tm}(\boldsymbol{I})$-recursive formulas are admitted for the $(\mathcal{A})$ in $C[(\mathcal{A}); (t)]$, and, in the reduction rule for $C[(\mathcal{A}); (t)]$, the truth value of A_t (under assignments) can be evaluated according to the semantics of first-floor-term-forms; see 2) above.

3.3) Let t be a type-form in which X a first-floor-variable-form is not bound. Then ΛXt is a (second-floor-) type-form. Let ϕ be a first-floor-term-form of the same type-form as X whose variable-forms are not bound in t, and let t' be obtained from t by replacing all occurrences of X by ϕ. Then $\Pi(\Lambda Xt; \phi)$ is a type-form with the reduction rule

$$\Pi(\Lambda Xt; \phi) \Longrightarrow t'.$$

The (first-floor-) variable-form in this are those in ΛXt and in ϕ.

3.4) For any s which is not of the form ΛXt, $\Pi(s; \phi)$ is still defined; see 2.2) in Definition 1.2 and the note to it.

3.5) The associated variables in a type-form t are those in the first-floor-type-forms of the variable-forms in t.

4) A second-floor-type-form is a second-floor-hyper-type if it contains no free first-floor-variable-forms (variables and associated variables inclusive).

Note. (1) In 3.3) above, if X a variable-form occurs in t, then it is in the form $\phi(X)$, where $\phi(X)$ is a first-floor-term-form of *at*-type or *fn*-type. So, for each complete assignment to $\phi(X)$, $\phi(X)$ can be evaluated according to the semantics of first-floor-term-forms.

(2) Here, too, $\Pi(s; \phi)$ is not necessarily meaningful. Adjustment will be made later.

(3) As was stated in 3.2) above, $\mathcal{R}^p[s', t'; i']$ is of a special form. We do *not* form this for arbitrary second-floor-type-forms s' and t'; that is, s' and t' do not contain ΛXt. It is possible, however, that an expression in them is of (first-floor-) type-form which involves $\mathcal{R}^1$ or $\mathcal{R}^2$.

PROPOSITION 1.1* *holds for second-floor-type-forms if appropriately modified; in* 3.3) *above, if* ΛXt *and* ϕ *are hyper-types, then the immediate reduct of* $\Pi(\Lambda Xt; \phi)$ *is also.*

DEFINITION 2.2. 1) The normality of a hyper-type is defined as in Definition 1.2* with the following modifications.

1.1) For every variable-form in a hyper-type whose associated first-floor-type-form $\mathfrak{s}$ is a (first-floor-) hyper-type, reduce $\mathfrak{s}$ to its normal form (which uniquely exists by Proposition 1.2).

1.2) If 1.1) has been executed, then $\Lambda X t(X)$ is normal.

1.3) For $\Pi(\mathfrak{s}\,;\phi)$ which does not have the reduction rule, the normalization problem will be settled subsequently.

2) We define $\varepsilon(\mathfrak{t})$, the constructional complexity of $\mathfrak{t}$ a second-floor-type-form relative to first-floor-objects.

2.1) $\varepsilon(\mathfrak{t})=1$ if $\mathfrak{t}$ is free of ΛX (X a properly first-floor-variable-form). In the subsequent cases we assume 2.1) is not the case.

2.2) $\varepsilon(\Lambda X t)=\varepsilon(t)+1$

2.3) $\varepsilon(\mathfrak{s}\rightarrow t)=\varepsilon(\mathfrak{s})+\varepsilon(t)$

2.4) $\varepsilon(\Pi(\mathfrak{s}\,;\phi))=\varepsilon(\mathfrak{s})+1$

2.5) $\varepsilon(C[(\mathcal{A})\,;(t)])=\varepsilon(t_1)+\cdots+\varepsilon(t_{m+1})$

Note. Since $\mathfrak{R}^p[\mathfrak{s}', t'\,; i']$ does not contain ΛX, this fits the case 2.1).

PROPOSITION 2.1. 1) *If* $\mathfrak{s}$ *and* t *are hyper-types such that* $\mathfrak{s}\Rightarrow t$ *and* $\varepsilon(\mathfrak{s})>1$, *then* $\varepsilon(t)<\varepsilon(\mathfrak{s})$.

2) 2) *Suppose* t' *is obtained from* t *by substitution of a first-floor-term-form for a variable-form. Then* $\varepsilon(t)=\varepsilon(t')$.

3) *The normalization theorem on hyper-types can be proved by induction on* ε, *under the assumption of the semantics of first-floor-term-forms.*

PROOF. 3) First execute 1.1) in the definition above. Suppose first for a hyper-type $\mathfrak{t}$ $\varepsilon(\mathfrak{t})=1$ holds. It $\mathfrak{t}$ is $C[(\mathcal{A})\,;(t)]$, then the truth value A_l is determined in the semantics of first-floor-term-forms, and hence the reduct is uniquely determined. If $\mathfrak{t}$ is $\mathfrak{R}[\mathfrak{s}, t\,; i]$ and i contains first-floor-term-forms, then i can be evaluated in the semantics of first-floor-term-forms. If $\mathfrak{t}$ is $\Pi(\mathfrak{s}\,;\phi)$ and ϕ is a first-floor-hyper-functional, then ϕ can be evaluated. With these facts at our disposal, we can follow the proof of Proposition 1.2 (relying on transfinite induction on d).

Suppose next $\varepsilon(\mathfrak{t})>1$. $\Lambda X t$ itself is normal. Consider $\Pi(\Lambda l t(l)\,;\phi)$, which is reduced to $t(\phi)$. If a first-floor-type-form $\mathfrak{s}(l)$ is the associated type-form of a first-floor-variable-form occurring in t becomes $\mathfrak{s}(\phi)$, then reduce this (if necessary) to the normal form. Consider next $\Pi(\Lambda X t(X)\,;\phi)$ as $\mathfrak{t}$, which is reduced to $t(\phi)$. ϕ belongs to first-floor, and hence

$$\varepsilon(t(\phi))>\varepsilon(\mathfrak{t}) .$$

In case of general $\Pi(\mathfrak{s}\,;\phi)$, consider the normal form of $\mathfrak{s}$ as before.

DFFINITION 2.3. The objects of respective (second-floor-) hyper-type can be defined as before. If $\mathfrak{t}$ is a hyper-type with $\varepsilon(\mathfrak{t})=1$, the definition is similar to that in Definition 1.3* by virtue of the semantics of first-floor-term-forms. Otherwise the desired objects can be defined by induction on ε, based upon the first-floor-sementics. An object of hyper-type $\Lambda Xt(X)$ with $[X]=\mathfrak{s}$ is a mechanism to associate with each J_{ϕ}, where ϕ is a first-floor-hyper-functional of hyper-type $\mathfrak{s}$, an object of (second-floor-) hyper-type $t(\phi)$. This is well-defined, since $\varepsilon(t(\phi))<\varepsilon(\Lambda Xt)$.

DEFINITION 2.4. 1) We assume first-floor-term-forms and (second-floor-) type-forms. We are to define second-floor-term-forms (which will simply be called term-forms when confusion is no1 likely) of certain type-forms, free and bound variables and variable-forms (of first-floor and second-floor) in them, the associated variables (of first grade) in first-floor-type forms occurring in the first-floor-variable-forms which constitute type-forms and the associated first-floor-variable-forms (of second grade) in type-forms. The underlying language will be denoted by $2\text{-}\mathcal{L}_{tm}(\boldsymbol{I})$. For any term-form Φ, its type-form will be denoted by $[\Phi]$.

Let n be a natural number. The second-floor-variable-form of the associated type-form $\mathfrak{s}$, written as $Y^{\mathfrak{s}}_{n}$, is prepared for every $\mathfrak{s}$.

(1) Each first-floor-term-form is a (second-floor-) term-form whose (first-floor-) variable-forms are those occurring in it and whose associated variables of first grade are those in its type-forms.

(2) Each variable-form $Y^{\mathfrak{s}}$ is an atomic term-form of type-form $\mathfrak{s}$. It is free in itself. The associated variables of first grade are those in type-forms of the variable-forms occurring in $\mathfrak{s}$ and the associated variable-forms of second grade are the variable-forms in $\mathfrak{s}$.

(3) If Φ is a term-form of type-form $\mathfrak{s}\rightarrow t$ and if Ψ is a term-form of type-form $\mathfrak{s}$, then $\Pi(\Phi\,;\Psi)$ is a term-form of type-form t.

(4) If Φ is a term-form of type-form $\Lambda Xt(X)$ with $[X]=\mathfrak{s}$ and if ϕ is a first-floor-term-form of type-form $\mathfrak{s}$, then $\Pi(\Phi\,;\phi)$ is a term-form of type-form $\Pi(\Lambda Xt(X)\,;\phi)$. The associated variables and variable-forms are those for Φ and for ϕ.

(5) If Y is a variable-form (of first-floor or second-floor) with $[Y]=\mathfrak{s}$ and Φ is a term-form with $[\Phi]=t$, where Y is not bound in Φ or t, then $\lambda Y\Phi$ is

a term-form, whose type-form is either $s \to t$ (when Y does not occur in t) or $\Lambda Y t$ (when Y occurs in t). The variable-forms in $\lambda Y \Phi$ are the corresponding ones in Φ except that Y becomes bound. The associated variable-forms are those for Φ and the variable-forms in s.

(6) Let $(\mathcal{A}) \equiv A_1, \cdots, A_m$ be $\mathcal{L}_{tm}(\boldsymbol{I})$-recursive formulas, and let $(\Phi) \equiv \Phi_1, \cdots, \Phi_m, \Phi_{m+1}$ be term-forms of type-forms $(t) \equiv t_1, \cdots, t_m, t_{m+1}$ respectively. Then $\mathcal{C}[(\mathcal{A});(\Phi)]$ is a term-form whose variable-forms are those in $(\mathcal{A})$ and in (Φ). The type-form is $\mathcal{C}[(\mathcal{A});(t)]$, and the associated variable-forms are the variable-forms in $(\mathcal{A})$ and the associated variable-forms for $(\mathcal{A})$ and (Φ).

(7) Let t_0 be $(N_0 \to N_0) \to N_0$, let $t_1(z), \cdots, t_b(z)$ be type-forms with a free *at*-type variable z, let S be an *fn*-type variable and let m and l be *at*-type variables. Define from these p_d, $d=1, \cdots, b$, as in Definition 2.1*. p_d becomes a (second-floor-) type-form. For any such $p \equiv p_d$, $\mathcal{B}^p$ is an atomic term-form with p_d as its associated type-form, and whose associated variable-forms are the variable-forms occurring free in $r_1, \cdots, r_b$. (See (11) in Definition 2.1* for details.)

2) A term-form which does not have associated variable-forms is called a (second-floor-) hyper-term.

3) A hyper-term which does not have free variable-forms is called a (second-floor-) hyper-functional.

4) The constructional complexity of a term-form Φ will be denoted by $*(\Phi)$ $(<\omega)$.

5) For each hyper-functional Φ, we introduce a (functional) symbol Q_Φ $(=Q)$, which is to be interpreted to be the object represented by Φ, say J_Φ (which will also be written as J_Q). Q_Φ is not an official term in the language.

Note. The first-floor-variable-forms and the second-floor-variable-forms are to be distinguished even if their type-forms happen to be identical (which are of first-floor).

PROPOSITION 2.2. 1) *Second-floor-term-forms are closed under substitutions.*

2) *If Φ and Ψ are "essentially the same" and Q and R respectively correspond to Φ and Ψ, then $J_Q = J_R$.*

DEFINITION 2.5. We are to define assignments of functional symbols to variable-forms in a manner similar to Definition 1.6. 1)~7) there are valid here.

8) Third assignments. Let $z \equiv z_1, \cdots, z_n$ be a finite sequence of distinct second-floor-variable-forms of hyper-types (with possibly functional symbols), and let $\boldsymbol{c} \equiv c_1, \cdots, c_n$ be a finite sequence of functional symbols (of second-floor)

such that c_k is of the same hyper-type as z_k, $1 \leqq k \leqq n$. Then

$$\boldsymbol{cz} \equiv (z_1/c_1, \cdots, z_n/c_n)$$

will denote the (third) assignment of $\boldsymbol{c}$ to $\boldsymbol{z}$.

9) Let Φ be any formal object, and let $\boldsymbol{ba}$ be complete for Φ. Then $\boldsymbol{cba}$ will denote the result of replacement of c_z by c_k in $\boldsymbol{ba}\Phi$. If $\boldsymbol{z}$ covers all the free second-floor-variable-forms in $\boldsymbol{ba}\Phi$, then $\boldsymbol{c}$ will be said to be complete for $\boldsymbol{ba}\Phi$. In this case, if Φ is a second-floor-term-form, $\boldsymbol{cba}\Phi$ will become a second-floor-hyper-functional (in the extended language).

Note. As was noted above, the first-floor-variable-forms and the second-floor-ones are to be distinguished, so that the former are relevant to second assignments, while the latter are relevant to third assignments.

DEFINITION 2.6. 1) CNPR(L, S) (the continuity principle) will be assumed, where L is a second-floor-term-form and S is a first-floor-term-foɹm. (Compare this with the continuity principle in the proof of Proposition 1.4.)

2) The interpretation of a term-form Φ at a complete assignment $\boldsymbol{cba}$, $\boldsymbol{J}(\Phi, \boldsymbol{c}, \boldsymbol{b}, \boldsymbol{a})$, can be defined similarly to the interpretation in Definition 1.7.

(1) For any first-floor-term-form Φ, $\boldsymbol{J}(\Phi, \boldsymbol{b}, \boldsymbol{a})$ has been defined in Definition 1.7.

(2) If Φ is a second-floor-variable-form Y^s, then $\boldsymbol{cba}$ determines a functional symbol c of hyper-type $\boldsymbol{ba}_s$. Let $\boldsymbol{J}(\Phi, \boldsymbol{c}, \boldsymbol{b}, \boldsymbol{a})$ be J_c.

(3) The cases where Φ is one of the forms $\mathcal{B}$, $\Pi(\Psi; \chi)$ and $\mathcal{C}[(\mathcal{A}); (\Psi)]$ can be dealt with as in Definition 3.2*, where the conditions $(\mathcal{A})$ can be interpreted in the semantics of first-floor-term-forms. The case where Φ is $\lambda Y\Psi$ and $[\Psi]$ is free of Y can be dealt with similarly to the first-floor case. Y ranges over the functional symbols of hyper-type $[\boldsymbol{ba}Y]$.

(4) Consider $\lambda X\Psi$ where $[X]=_s$ and $[\Psi]=\iota(X)$. $\boldsymbol{J}(\lambda X\Psi; \boldsymbol{c}, \boldsymbol{b}, \boldsymbol{a})$ is the mechanism M which associates with each J_Q, where Q is the functional symbol of a first-floor-hyper-functional χ of hyper-type $\boldsymbol{a}_s$, the object $\boldsymbol{J}(\Psi, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{a})$, where $\boldsymbol{d}=(\boldsymbol{a}X/Q, \boldsymbol{b})$, the type-form of which is $\iota(\chi)$. M is of hyper-type $\Lambda\boldsymbol{a}X\boldsymbol{ba}\iota(X)$.

PROPOSITION 2.4. *The* $\boldsymbol{J}$ *above is well-defined.* (See Proposition 1.4.)

DIFINITION 2.7. The set $\boldsymbol{U}$ of the mechanisms in which the hyper-functionals are interpreted consistently with respect to $\boldsymbol{J}$ (see Definitions 1.7 and 2.6) will be called the two-storied universe of transfinite mechanisms. $\boldsymbol{U}$ contains the realizations of closed $\mathcal{L}_0(\boldsymbol{I})$-terms, first-floor-hyper-functionals and second-floor-hyper-functionals.

§ 3. The hyper-principle for the two-storied universe of transfinite mechanisms.

DEFINITION 3.1. 1) The language $\mathcal{L}_2$ we are to consider contains the language 2-$\mathcal{L}_{tm}(\boldsymbol{I})$ as well as the predicate constants Δ_1 and Δ_2. The logical connectives accepted are $\wedge$, $\vdash$ and $\forall$.

2) The formula-forms of $\mathcal{L}_2$ are defined as follows.

2.1) For any pair of term-forms Φ and Ψ of type N_0, $\Phi=\Psi$ is an atomic formula-form.

2.2) $\Delta_l(i, \boldsymbol{f}_l, X_l)$, $l=1, 2$, is an atomic formula-form, where i is of at-type, $\boldsymbol{f}_l$ stands for a finite sequence of term-forms of at-type or fn-type and X_l stands for a term-form, whose type-form will be specified in 3) below.

2.3) The class of formula-forms are closed with respect to $\wedge$, $\vdash$ and $\forall X$, X any variable or variable-form.

3) For $[X_l]$ (the type-form of X_l) in 2.2), we shall explain how to determine it with examples. We assume $\boldsymbol{f}_1 \equiv \boldsymbol{f}_2 \equiv f$, which is of fn-type. First let l be 1. Let us temporarily suppose that Ξ be a parameter which yields the type-form of the V_0 in $\Delta_1(j, g, V_0)$ (at $j<^1 i$ and any g); that is,

(1) $$\Pi(\Xi\,; j, g)=[V_0] \quad (=v_0(\Xi\,; j, g)) \quad \text{if} \quad j<^1 i.$$

Let us write $[V_0\,; \Xi]$ for this. Now define

(2) $$\alpha(\Xi\,; i)=\Lambda j \Lambda g C[j<^1 i\,; v_0(\Xi\,; j, g), ept],$$

$$\beta(\Xi\,; i, f)=\Lambda j \Lambda x(v_0(\Xi\,; j, s(i, f, j, x))\rightarrow N_0)$$

for a term s. $\alpha(\Xi\,; i)$ and $\beta(\Xi\,; i, f)$ are expressions in the language $\mathcal{L}_{tp}(\boldsymbol{I})$ with parameter Ξ. Assume that these α and β are re-assigned to Δ_1, and put

(3) $$\iota \equiv \iota(\Xi)=\Lambda m C[m=0, m=1\,; \alpha(\Xi\,; i), \beta(\Xi\,; i, f), ept].$$

Define $\boldsymbol{M}$ and $\boldsymbol{N}$ by

(4) $$\boldsymbol{M}(ept, \iota, o)=ept,$$

$$\boldsymbol{N}(ept, \iota\,; i)=\Lambda k C[k<^1 i\,; \boldsymbol{M}(ept, \iota\,; k), ept],$$

$$\boldsymbol{M}(ept, \iota\,; i)=\iota(\boldsymbol{N}(ept, \iota\,; i)).$$

We can show that, for each i in I_1, $\boldsymbol{M}(ept, \iota\,; i)$ is a first-floor-type-form, and hence $\mathfrak{R}^1[ept, \iota\,; i]$ can be admitted as a first-floor-type-form. Put

(5) $$[X_1]=\gamma(i, f)=\mathfrak{R}^1[ept, \iota\,; i]$$

for the X_1 in $\Delta_1(i, f, X_1)$. $\gamma(i, f)$ consistently determines the type-form of X_1 in $\Delta_1(i, f, X_1)$ for all i and f. (All the objects here are of first-floor.)

Next let l be 2. Suppose that Θ be a parameter which yields the type-form of the U_0 in $\varDelta_2(j, f, U_0)$ (at $j<^2 i$); that is,

(6) $$\Pi(\Theta; j, f)=[U_0] \ (=u_0(\Theta; j, f)) \quad \text{if} \quad j<^2 i.$$

Write $[U_0; \Theta]$ for this. Define

(7) $$\zeta(\Theta; i, f)=\Lambda j(\gamma(h_3(j), h_4(f))\to \mathcal{C}[j<^2 i; u_0(\Theta; j, f), ept]),$$

where h_3 is supposed to be a function which yields an element $h(j)$ of I_1 when j is an element of I_2, and $h_4(f)$ is an $\mathcal{L}_0(\boldsymbol{I})$-term with f. Assume that this ζ is pre-assigned to $\varDelta_2$, and put

(8) $$\mathfrak{s}\equiv\mathfrak{s}(\Theta)=\zeta(\Theta; i, f).$$

Define $\boldsymbol{K}$ and $\boldsymbol{L}$ by

(9) $$\boldsymbol{K}(ept, \mathfrak{s}; o)=ept,$$

$$\boldsymbol{L}(ept, \mathfrak{s}; i)=\Lambda l\mathcal{C}[l<^2 i; \boldsymbol{K}(ept, \mathfrak{s}; l), ept],$$

$$\boldsymbol{K}(ept, \mathfrak{s}; i)=\mathfrak{s}(\boldsymbol{L}(ept, \mathfrak{s}; i)).$$

We can show that, for each i in I_2, $\boldsymbol{K}(ept, \mathfrak{s}; i)$ is a first-floor-type-form, and hence $\mathfrak{R}^2[ept, \mathfrak{s}; i]$ can be admitted as a type-form. Put

(10) $$[X_2]=\delta(i, f)=\mathfrak{R}^2[ept, \mathfrak{s}; i]$$

for the X_2 in $\varDelta_2(i, f, X)$. $\delta(i, f)$ consistently determines the first-floor-type-form of such X_2 for all i and f.

DEFINITION 3.2. The axiom set $\mathcal{A}(\boldsymbol{I})$ of the $\mathcal{L}_2$-formula-forms consists of $(\mathcal{A}(\boldsymbol{I})-1)\sim(\mathcal{A}(\boldsymbol{I})-4)$ below.

$(\mathcal{A}(\boldsymbol{I})-1)$ The reduction rules of type-forms (see Definitions 1.2 and 2.1).

$(\mathcal{A}(\boldsymbol{I})-2)$ The axiom on $\varDelta_1$ consists of two implications $(\varDelta_{1,1})$ and $(\varDelta_{1,2})$. Abiding with the spirit of 3) in Definition 3.1, we work here also on an example. An expression of $\mathcal{L}_2$ with parameters, say G_1, is pre-assigned to $\varDelta_1$. As an example, suppose G_1 is of the form below.

$$G_1\equiv G_1(i, f, V, N, \Sigma_1)\equiv\forall j<^1 i\forall g(A(i, g)\vdash\Sigma_1(j, g, \Pi(V; j, g)))$$

$$\wedge\forall j<^1 i\forall x\forall V_2(\Sigma_1(j, s, V_2)\vdash B(\Pi(N; j, x, V_2), i, f, j, x)),$$

where $s\equiv s(i, f, j, x)$ is an $\mathcal{L}_0(\boldsymbol{I})$-term, A and B are $\mathcal{L}_0(\boldsymbol{I})$-recursive, and the type-forms of the variable-forms are listed below. Put

$$\xi(i, f)=\Lambda k\mathcal{C}[k<^1 i; \mathfrak{R}^1[ept, \mathfrak{t}; k], ept].$$

$$[V]=\alpha(\xi(i, f); i),$$

$$[N]=\beta(\xi(i, f); i, f),$$
$$[V_2]=v_0(\xi(i, f); j, s).$$

All the expressions are of first-floor. Also are pre-assigned first-floor-term-forms as below.

$$V^*=\lambda X\lambda j\lambda g\mathcal{C}[j<^1 i; \Pi(X; 0, j, g), ept],$$
$$N^*=\lambda X\lambda j\lambda x\lambda V_2\mathcal{C}[j<^1 i; \Pi(X; 1, j, x, V_2), ept],$$
$$X^*=\lambda V\lambda N(V, N),$$

where $[X]=\gamma(i, f)$ and (V, N) denotes the pair of V and N. Now, wit $[X_1]=\gamma(i, f)$, $(\Delta_{1,1})$ and $(\Delta_{1,2})$ are presented below.

$$(\Delta_{1,1}) \qquad \Delta_1(i, f, X_1)\vdash G_1(i, f, \Pi(V^*; X_1), \Pi(N^*; X_1), \Delta_1)$$
$$(\Delta_{1,2}) \qquad G_1(i, f, V, N, \Delta_1)\vdash \Delta_1(i, f, \Pi(X^*, V, N))$$

($\mathcal{A}(\boldsymbol{I})-3$) The axiom on Δ_2 consists of two implications $(\Delta_{2,1})$ and $(\Delta_{2,2})$. An expression of $\mathcal{L}_2$ with parameters, say G_2, is preassigned to Δ_2. As an example, suppose G_2 is of the form below.

$$G_2\equiv G_2(i, f, W, \Sigma_1, \Sigma_2)$$
$$\equiv \forall J\forall T[\forall V_1(\Sigma_1(h_1(i), h_2(f), V_1)\vdash \Pi(J; V_1)<^2 i$$
$$\wedge \Sigma_2(\Pi(J; V_1), f, \Pi(T; V_1)))\vdash C]$$
$$\wedge \forall j<^2 i\forall V_2(\Sigma_1(h_3(j), h_4(f), V_2)\vdash \Sigma_2(j, f, \Pi(W; j, V_3)),$$

where $h_1(i)$ and $h_2(f)$ are $\mathcal{L}_0(\boldsymbol{I})$-terms and $\mathcal{C}$ is $\mathcal{L}_0(\boldsymbol{I})$-recursive. Put

$$\eta(i, f, k)=\mathcal{C}[k<^2 i; \mathcal{R}^2[ept, {}_s; k], ept].$$
$$[J]=\gamma(h_1(i), h_2(f))\rightarrow N_0,$$
$$[T]=\Lambda V_1\mathcal{C}[\Pi(J; V_1)<^2 i; \eta(i, f, \Pi(J; V_1))]$$
$$[V_1]=\gamma(h_1(i), h_2(f)),$$
$$[V_2]=\gamma(h_3(j), h_4(f)),$$
$$[W]=\Lambda j([V_2]\rightarrow\eta(i, f, j))=\Lambda j(\gamma(h_3(j), h_4(f))\rightarrow\eta(i, f, j))$$
$$=\mathcal{R}^2(ept, {}_s; i).$$

Notice that T is of second-floor. Now, also are pre-assigned the following first-floor-term-forms.

$$W^*=\lambda X\lambda j\lambda V_2\mathcal{C}[j<^2 i; \Pi(X; j, V_2), ept],$$
$$X^*=\lambda W\cdot W,$$

where $[X]=\delta(i, f)$. Now, with $[X_2]=\delta(i, f)$, $(\Delta_{2,1})$ and $(\Delta_{2,2})$ are presented below.

$$(\Delta_{2,1}) \qquad \Delta_2(i, f, X_2) \vdash G_2(i, f, \Pi(W^* ; X_2), \Delta_1, \Delta_2)$$

$$(\Delta_{2,2}) \qquad G_2(i, f, W, \Delta_1, \Delta_2) \vdash \Delta_2(i, f, \Pi(X^* ; W))$$

Notice that

$$\delta(i, f) \Longrightarrow \Lambda j(\gamma(h_3(j), h_4(f)) \longrightarrow \eta(i, f, j))=[W].$$

$(\mathcal{A}(I)-4)$ A formal presentation of the continuity principle, CNPR$(L ; S)$, where L is of second-floor, while S is of first-floor.

DEFINITION 3.3. 1) The semantics of $\mathcal{L}_2$-formula-forms is defined as follows.

1.1) Assignments of functional symbols to variables and variable-forms defined in Definitions 1.6 and 2.5 are assumed.

1.2) The interpretations of term-forms at complete assignments defined in Definitions 1.7 and 2.6 are assumed.

1.3) Assignments to the free occurrences of variables and variable-forms in a formula-form can be defined naturally.

1.4) A formula-form $\Phi=\Psi$, where Φ and Ψ are term-forms of type N_0, is true under a complete assignment $\boldsymbol{d}$ if $\boldsymbol{J}(\Phi ; \boldsymbol{d})$ and $\boldsymbol{J}(\Psi ; d)$ are the same objects.

1.5) The logical connectives $\wedge$, $\vdash$ and $\forall$ are interpreted classically. The X in a quantifier $\forall X$ ranges over the set of mechanisms in $\boldsymbol{U}$ of hyper-type $[\boldsymbol{d}X]$ ($\boldsymbol{d}$ a complete assignment for the type-form of X). Recall that assignments must be discriminated according to the floor X belongs to; if X is a first-floor-variable-form, then the second assignment is eligible, while if it is of second-floor, then the third assignment is eligible.

1.6) As for Δ_l, consider Δ_1 first. Suppose for every $j<^1 i$ and every assignment to g and U, the truth value of $\Delta_1(j, g, U)$ has been determined. Then the truth value of $G_1(i, f, V, N, \Delta_1)$ is determined with respect to every complete assignment, since it suffices to check $\Delta_1(j, g, U)$ for $j<^1 i$. Now define the truth value of $\Delta_1(i, f, X)$ to be that of $G_1(i, f, \Pi(V^*, X_1), \Pi(N^* ; X_1), \Delta_1)$; that is, by equating the premise and the consequence of $\vdash$ in $(\Delta_{1,1})$. Δ_2 can be dealt with similarly. That is, assuming the truth values of Δ_1 and $\Delta_2(j, g, V)$ for all $j<^2 i$, define the truth value of $\Delta_2(i, f, X)$ to be that of

$$G_2(i, f, \Pi(W^* ; X_2), \Delta_1, \Delta_2).$$

2) The theories of second-floor-type-forms, of second-floor-term-forms and of $\mathcal{L}_2$-formula-forms, including the axiom set, the assumption CNPR, the interpretations and the two-storied universe $\boldsymbol{U}$, will be all put into one principle,

the *hyper-principle for the two-storied universe of transfinite mechanisms,* and will be symbolized by TM[2].

3) A formula-form of $\mathcal{L}_2$ is said to be TM[2]-valid if it becomes true under every complete assignment.

LEMMA. *Suppose Φ and Ψ are term-forms which become the same objects under every complete assignment. Then a formula $A(\Phi)$ is equivalent to $A(\Psi)$ with respect to every complete assignment.*

THEOREM. *The axioms $(\mathcal{A}(\boldsymbol{I})-2)\sim(\mathcal{A}(\boldsymbol{I})-4)$ are* TM[2]*-valid.*

PROOF. $(\mathcal{A}(\boldsymbol{I})-4)$ is valid by the assumption in Definition 2.6.

$(\mathcal{A}(\boldsymbol{I})-2)$ $(\varDelta_{1,1})$ is valid by definition. As for $(\varDelta_{1,2})$, suppose $G_1(i, f, V, N, \varDelta_1)$ is true. Since

$$\Pi(X^*; V, N)=(V, N),$$

the consequence of $(\varDelta_{1,2})$ is $\varDelta_1(i, f, (V, N))$.

$$\Pi(V^*; (V, N))=\lambda j\lambda g\mathcal{C}[j<^1 i; \Pi(V; j, g), ept],$$

$$\Pi(N^*; (V, N))=\lambda j\lambda x\lambda V_2\mathcal{C}[j<^1 i; \Pi(N; j, x, V_2), ept].$$

So,

$$G_1(i, f, \Pi(V^*, (V, N)), \Pi(N^*; (V, N)), \varDelta_1)\equiv G_1(i, f, V, N, \varDelta_1),$$

which is true by assumption. So, the equivalence in $(\varDelta_{1,1})$ implies $\varDelta_1(i, f, (V, N))$.

$(\mathcal{A}(\boldsymbol{I})-3)$ $(\varDelta_{2,1})$ is valid by definition. As for $(\varDelta_{2,2})$, suppose $G_2(i, f, W, \varDelta_1, \varDelta_2)$ is true. Since

$$\Pi(X^*; W)=W,$$

the consequence of $(\varDelta_{2,2})$ is $\varDelta_2(i, f, W)$.

$$\Pi(W^*; W)=\lambda j\lambda V_2\mathcal{C}[j<^2 i; \Pi(W; j, V_2), ept],$$

and so,

$$G_2(i, f, \Pi(W^*; W), \varDelta_1, \varDelta_2)\equiv G_2(i, f, W, \varDelta_1, \varDelta_2),$$

which is true by assumption. So, the equivalence in $(\varDelta_{2,1})$ implies $\varDelta_2(i, f, W)$.

Note. The axioms $(\mathcal{A}(\boldsymbol{I})-2)$ and $(\mathcal{A}(\boldsymbol{I})-3)$ are the central theme of TM[2], since, they describe the *mechanism* of transfinite inductive definitions in their concrete contexts.

References

[1] Takeuti, G. and Yasugi, M., The ordinals of the systems of second order arithmetic with the provably $\varDelta_2^1$-comprehension axiom and with the $\varDelta_2^1$-comprehension

axiom respectively, Japanese J. of Math., **41** (1973), 1-67.

[2] Yasugi, M., Construction principle and transfinite induction up to ε_0, J. Austral. Math. Soc., Ser. A, **32** (1982), 24-47.

[3] Yasugi, M., Groundedness property and accessibility of ordinal diagrams, J. Math. Soc. Japan, **37** (1985), 1-16.

[4] Yasugi, M., Hyper-principle and the functional structure of ordinal diagrams, Comm. Math. Univ. St. Pauli, **34** (1985), 227-263 (the opening part); **35** (1986), 1-37 (the concluding part).

Faculty of Science
Kyoto Sangyo University
Kita-ku
Kyoto, Japan 603