Vol.16 No.1 J. Comput. Sci. & Technol. Jan. 2001

Semantics of Constructions (I) — The Traditional Approach

FU Yuxi (傅育熙)

Department of Computer Science, Shanghai Jiaotong University, Shanghai 200030, P.R. China

E-mail:

Received January 11, 1999; revised April 14, 2000.

Abstract It is well known that impredicative type systems do not have set theoretical semantics. This paper takes a look at semantics of inductive types in impredicative type systems. A generalized inductive type is interpreted as an omega set generated by effectivizing a certain rule set. The result provides a semantic justification of inductive types in the calculus of constructions.

Keywords type theory, inductive type, ω-set

1 Introduction

Constructive type theory has been developed around three principles. The oft-quoted *Curry-Howard's principle*[1] may be stated as follows:

Constructive propositions are types.

This is what underlies Girard's work on system F and some recent interest in polymorphism and functional programming languages. Here the notion of sets is missing, which is the prime reason for the limitation of the languages designed with this principle. Perhaps this is not a serious drawback as far as functional programming is concerned. The *Martin-Löf's principle* is on the other end of the pole. It says that

Constructive sets are types.

What is lacking in these calculi is a logic. As a consequence the Leibniz equality is not expressible. We are therefore forced to internalize the definitional equality. Hence the somewhat odd identity sets[2]. Languages in this group have good mechanism for defining and manipulating data types. But applications to program specifications and verifications are limited due to the absence of a logic. From [3], and especially [4] (see also [5]), we now know that it is perfectly sound to unify these two kinds of languages. We will call *Russell's principle* the following:

The range of a constructive significance is a type.

The idea has its origin in *Principia Mathematica*[6]. To make sensible use of this principle, one has first of all to answer the following question: what is the relationship between propositions and sets? Obviously things should be structured. By Russell's principle, there should be a type *Prop* of all propositions. There are three obvious choices. (1) The sets sit on top of *Prop*, that is $Prop : Type$; this is the *Calculus of Constructions*, or *CC*[3,5,7,8]. (2) The sets sit within *Prop*, that is $Type \subset Prop$. Languages incorporating this design decision have elimination rules for closed universes, but the usefulness of it is questionable. (3) The sets and propositions are separated[9]. *ECC* (Enriched Calculus of Constructions) is the best representative in the *CC*-like calculi. Part of the expressive power of *ECC* comes

This work is funded by the National Natural Science Foundation of China (No.69973030). It is also supported by BASICS, Center of Basic Studies in Computing Science, sponsored by Shanghai Education Committee. BASICS is affiliated to the Department of Computer Science at Shanghai Jiaotong University.

from the fact that $Type_i$ ($i \in \omega$) are open universes, which is a consequence of $Prop : Type_0$. The studies carried out in [10] and [11] clarify, both negatively and positively, the role of propositions in the *Calculus of Constructions*. In [10], a counter example is given showing that if we identify sets with propositions, the *Calculus of Constructions* is not conservative over *HOL*, the constructive version of Church's higher order logic[12]. In [11], it is proved that, when properly separated, it is conservative over *HOL*. The significance of their work is in showing that the first two principles are different.

There remain many questions about the *Calculus of Constructions*. To start with, there is no civilized way of carrying out metatheoretical investigations into various *Enriched Calculi of Constructions*. The model theoretic methods such as the "candidates de reductibilité" are doomed to be complicated. The main idea in those methods is to reduce the issue to a proof of the soundness of some term model, which inherits all the combinatorial complexity from a case-study approach. Second we don't know the expressive power of such languages. What we now can do is to interpret various traditional constructive languages in *ECC* to have a good understanding of the relationships. In doing so, one should take into account of the fact that *ECC* contains a logic. The usual interpretation in Martin-Löf's set theory, see for example [13, 14], might have a more faithful counterpart in *ECC* — that is logical, formulas are interpreted as impredicative propositions while the notion of sets is kept at the *Type*-level as it as. See [15, 16] for an initial attempt.

Russell's principle subsumes the other two. As a result, we have this useful slogan:

> `One can extend the` *Calculus of Constructions* `to include whatever`
> `one has in Martin-Löf's set theory.`

Two points need to be clarified. First, by Martin-Löf's set theory we mean the traditional one. What has prevented Martin-Löf from calling something type theory should be a good enough reason for us not to call it Martin-Löf's set theory. Second, one might argue against the slogan by saying that universes in Martin-Löf's set theory are closed while $Type_i$ ($i \in \omega$) are not. This is really a misconception. The closed universes should sit inside each $Type_i$. The *Unifying Theory of Dependent Types*[17] is a calculus proposed along this line, although the author's emphasis is on the decidability of the language.

The main purpose of this paper is to reinforce our confidence in the above slogan by taking a look at what happens in model theory. In [18], a set-theoretic model is given to the inductive types defined in Martin-Löf's set theory, using Aczel's notion of rule sets[19]. A similar model using κ-continuous functors is described in [20]. Neither of them can treat inductive types in the *Calculus of Constructions*, which is impredicative. In [18] it was said that "This interpretation does of course not extend to the full system of the calculus of constructions extended with inductive types. Having a set-theoretic model is one of the properties which distinguish predicative from the impredicative type theory.". Of course nothing prevents us from giving a constructive set theoretical model. In fact, the model is one version of the recursive set theory. This reflects the computational aspect of the inductive types. Inductive types are by essence computational objects. In this paper we are going to transplant the classical set-theoretical construction to the category ω-SET. The definition of the interpretation follows [18] closely. Our contribution is to effectivize the sets obtained from certain rule sets.

2 The Enriched Calculus of Constructions

We present the language in the style of Tarski. This formulation[8,11] has some theoretical advantages. For instance, the standard ω-SET model is perfect for this explicit calculus. The following rules are the skeleton for the *Calculus of Constructions*.

$$\frac{}{\vdash Prop : Type} \qquad \frac{\Gamma \vdash A : Type}{\Gamma, x : A \vdash Prop : Type} \qquad \frac{\Gamma, x : A, \Gamma' \vdash Prop : Type}{\Gamma, x : A, \Gamma' \vdash x : A}$$

$$\frac{\Gamma \vdash P : Prop}{\Gamma \vdash Prf(P) : Type} \qquad \frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \lambda x{:}A.b : \Pi x{:}A.B} \qquad \frac{\Gamma \vdash f : \Pi x{:}A.B \quad \Gamma \vdash a : A}{\Gamma \vdash fa : B[a]}$$

$$\frac{\Gamma, x : A \vdash B : Type \quad \Gamma \vdash A : Type}{\Gamma \vdash \Pi x{:}A.B : Type} \qquad \frac{\Gamma, x : A \vdash P : Prop}{\Gamma \vdash \forall x{:}A.P : Prop}$$

$$\frac{\Gamma, x : A \vdash M : Prf(P)}{\Gamma \vdash \Lambda x{:}A.M : Prf(\forall x : A.P)} \qquad \frac{\Gamma \vdash M : Prf(\forall x : A.P) \quad \Gamma \vdash N : A}{\Gamma \vdash M \bullet N : Prf(P[N/x])}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash B : Type \quad A = B}{\Gamma \vdash a : B}$$

We have omitted all the nonextensional equational rules defining $=$.

There are many ways to extend the language just described. Here we enrich it with the generalized inductive types as formulated in [18].

Context

$$\frac{\Gamma \vdash A_1 : Type, \quad \Gamma, x_1 : A_1 \vdash A_2 : Type, \ldots, \quad \Gamma, x_1 : A_1, \ldots, x_{n-1} : A_{n-1} \vdash A_n : Type}{\Gamma \vdash [x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n] \; Ctxt}$$

Context realization

$$\frac{\Gamma \vdash a_1 : A_1, \quad \Gamma \vdash a_2 : A_2[a_1], \ldots, \Gamma \vdash a_n : A_n[a_1, \ldots, a_{n-1}]}{\Gamma \vdash (a_1, \ldots, a_n) : [x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n]}$$

For a context $\Delta \stackrel{\text{def}}{=} [x_1 : A_1, \ldots, x_n : A_n]$, $\Pi z : \Delta.C$ is intuitively 'isomorphic' to $\Pi x_1 : A_1. \cdots \Pi x_n : A_n.C$. Substitution into contexts is computed componentwise: $\Delta[a]$ means $[x_1 : A_1[a], \ldots, x_n : A_n[a]]$.

Formation

$$\frac{\begin{array}{c}\Gamma \vdash \Delta_1 \; Ctxt, \quad \Gamma\Delta_1 \vdash \Phi_{11} \; Ctxt, \ldots, \quad \Gamma\Delta_1 \vdash \Phi_{1n_1} \; Ctxt, \ldots, \\ \Gamma \vdash \Delta_n \; Ctxt, \quad \Gamma\Delta_n \vdash \Phi_{n1} \; Ctxt, \ldots, \Gamma\Delta_n \vdash \Phi_{nn_n} \; Ctxt\end{array}}{\Gamma \vdash \mu(\Delta_1, \ldots, \Delta_n).[\Phi_{11}, \ldots, \Phi_{1n_1}; \ldots; \Phi_{n1}, \ldots, \Phi_{nn_n}] : Type}$$

We will abbreviate $\mu(\Delta_1, \ldots, \Delta_n).[\Phi_{11}, \ldots, \Phi_{1n_1}; \ldots; \Phi_{n1}, \ldots, \Phi_{nn_n}]$ to $\mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}]$ or even to μ.

Introduction

$$\frac{\Gamma \vdash \boldsymbol{a} : \Delta_i, \quad \Gamma \vdash b_k : \Phi_{ik}[\boldsymbol{a}] \to \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}], \; k \in [1 \cdot\cdot n_i]}{\Gamma \vdash intro_i^{\mu}(\boldsymbol{a}, b_1, \ldots, b_{n_i}) : \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}]} \qquad (i = 1 \cdot\cdot n)$$

Elimination

$$\frac{\begin{array}{l}\Gamma, z : \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}] \vdash C : Type \\ \Gamma \vdash d_i \in \Pi \boldsymbol{x} : \Delta_i. \left(\begin{array}{l}\Pi y_1 : \Phi_{i1}[\boldsymbol{x}] \to \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}]. \\ \Pi h_{i1} : (\Pi \boldsymbol{z} : \Phi_{i1}[\boldsymbol{x}].C[y_1(\boldsymbol{z})]). \\ \cdots \\ \Pi y_{n_i} : \Phi_{in_i}[\boldsymbol{x}] \to \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}]. \\ \Pi h_{in_i} : (\Pi \boldsymbol{z} : \Phi_{in_i}[\boldsymbol{x}].C[y_{n_i}(\boldsymbol{z})]).\end{array}\right) C[intro_i^{\mu}(\boldsymbol{x}, y_1, \ldots, y_{n_i})] \\ for \; i = 1 \cdot\cdot n\end{array}}{\Gamma \vdash rec_{\mu}(d_1, \ldots, d_n) \in \Pi z : \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}].C}$$

Computation

$$\frac{\begin{array}{l}\Gamma, z : \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}] \vdash C : Type \\ \Gamma \vdash d_i \in \Pi \boldsymbol{x} : \Delta_i. \left(\begin{array}{l}\Pi y_1 : \Phi_{i1}[\boldsymbol{x}] \to \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}]. \\ \Pi h_{i1} : (\Pi \boldsymbol{z} : \Phi_{i1}[\boldsymbol{x}].C[y_1(\boldsymbol{z})]). \\ \cdots \\ \Pi y_{n_i} : \Phi_{in_i}[\boldsymbol{x}] \to \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}]. \\ \Pi h_{in_i} : (\Pi \boldsymbol{z} : \Phi_{in_i}[\boldsymbol{x}].C[y_{n_i}(\boldsymbol{z})]).\end{array}\right) C[intro_i^{\mu}(\boldsymbol{x}, y_1, \ldots, y_{n_i})] \\ for \; i = 1 \cdot\cdot n \\ \Gamma \vdash \boldsymbol{a} : \Delta_i, \; \Gamma \vdash b_1 : \Phi_{i1}[\boldsymbol{a}] \to \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}], \ldots, \Gamma \vdash b_{n_i} : \Phi_{in_i}[\boldsymbol{a}] \to \mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}]\end{array}}{\Gamma \vdash rec_{\mu}(\boldsymbol{d})(intro_i^{\mu}(\boldsymbol{a}, \boldsymbol{b})) = \left(\begin{array}{c} d_i(\boldsymbol{a}, b_1, \lambda \boldsymbol{z} : \Phi_{i1}[\boldsymbol{a}].rec_{\mu}(\boldsymbol{d})(b_1(\boldsymbol{z})), \ldots, \\ b_{n_i}, \lambda \boldsymbol{z} : \Phi_{in_i}[\boldsymbol{a}].rec_{\mu}(\boldsymbol{d})(b_{n_i}(\boldsymbol{z})))\end{array}\right) : C[intro_i^{\mu}(\boldsymbol{a}, \boldsymbol{b})]}$$

The formation and introduction rules tell us that the generalized inductive types are two-dimensional generalizations of the well-known W-types. The elimination and computation rules describe how functions on these inductive types are defined and how they are computed. In the latter two rules, we have linearized what are completely independent of each other.

3 A Quick Review of the ω-Set Model

This section reviews some of the basic facts about the ω-Set model[4,5,21]. In the sequel we will overload the syntax for the type theory and its interpretation in ω-Set. For recursion theory, see [22].

Definition 3.1. *An ω-set $(X, \vdash_X)$ consists of a set X, the underlying set, and a relation $\vdash_X \subset \omega \times X$ such that for any $a \in X$, there exists some $n \in \omega$ satisfying $n \vdash_X a$ (every element of X is witnessed by some recursive function). A morphism from $(X, \vdash_X)$ to $(Y, \vdash_Y)$ is a function $f : X \longrightarrow Y$ that is tracked (realized) by some recursive function n. By "n tracks f" we mean that if $p \vdash_X a$ then $n \cdot p \vdash_Y fa$, where $n \cdot p$ is the result of applying the n-th recursive function to the number p.*

Assume Γ is an ω-set and $A, A' : |\Gamma| \longrightarrow \omega$-Set are two maps. We can define the ω-sets $\sigma(\Gamma, A)$ and $\pi(\Gamma, A)$ and two maps $A+A', A\times A' : |\Gamma| \longrightarrow \omega$-Set. If $B : |\sigma(\Gamma, A)| \longrightarrow \omega$-Set is another map, we can define two more maps $\sigma_\Gamma(A, B), \pi_\Gamma(A, B) : |\Gamma| \longrightarrow \omega$-Set. Notice that a map $f : |\Gamma| \rightarrow \omega$-Set is defined by prescribing $f(\alpha)$ for an arbitrary $\alpha \in |\Gamma|$.

ω-set	the underlying set	the realizability relation
$\sigma(\Gamma, A)$	$\{(\gamma, a) \mid \gamma \in \lvert\Gamma\rvert \wedge a \in \lvert A(\gamma)\rvert\}$	$\langle m, n\rangle \vdash_{\sigma(\Gamma,A)} (\gamma, a)$ if and only if $m \vdash_\Gamma \gamma \wedge n \vdash_{A(\gamma)} a$
$\sigma_\Gamma(A, B)(\alpha)$	$\{(a, b) \mid a \in \lvert A(\alpha)\rvert \wedge b \in \lvert B(\alpha, a)\rvert\}$	$\langle m, n\rangle \vdash_{\sigma_\Gamma(A,B)(\alpha)} (a, b)$ if and only if $m \vdash_{A(\alpha)} a \wedge n \vdash_{B(\alpha,a)} b$
$\pi(\Gamma, A)$	$\{f : \lvert\Gamma\rvert \rightarrow \bigcup_{\gamma\in\lvert\Gamma\rvert} \lvert A(\gamma)\rvert \mid$ $\forall\gamma \in \lvert\Gamma\rvert . f(\gamma) \in \lvert A(\gamma)\rvert$ $\wedge \exists n \in \omega . n \vdash_{\pi(\Gamma,A)} f\}$	$n \vdash_{\pi(\Gamma,A)} f$ if and only if $\forall\gamma \in \lvert\Gamma\rvert . \forall p \in \omega .$ $p \vdash_\Gamma \gamma \Rightarrow n \cdot p \vdash_{A(\gamma)} f(\gamma)$
$\pi_\Gamma(A, B)(\alpha)$	$\{f : \lvert A(\alpha)\rvert \rightarrow \bigcup_{a\in\lvert A(\alpha)\rvert} \lvert B(\alpha, a)\rvert \mid$ $\forall a \in \lvert A(\alpha)\rvert . f(a) \in \lvert B(\alpha, a)\rvert$ $\wedge \exists n \in \omega . n \vdash_{\pi_\Gamma(A,B)(\alpha)} f\}$	$n \vdash_{\pi_\Gamma(A,B)(\alpha)} f$ if and only if $\forall a \in \lvert A(\alpha)\rvert . \forall p \in \omega .$ $p \vdash_{A(\alpha)} a \Rightarrow n \cdot p \vdash_{B(\alpha,a)} f(a)$
$(A + A')(\alpha)$	$\{(0, a) \mid a \in \lvert A(\alpha)\rvert\} \cup \{(1, b) \mid b \in \lvert A'(\alpha)\rvert\}$	$\langle i, n\rangle \vdash_{(A+A')(\alpha)} (j, c)$ if and only if $\begin{cases} i = j = 0 \wedge n \vdash_{\lvert A(\alpha)\rvert} c \\ i = j = 1 \wedge n \vdash_{\lvert A'(\alpha)\rvert} c \end{cases}$
$(A \times A')(\alpha)$	$\{(a, a') \mid a \in \lvert A(\alpha)\rvert \wedge a' \in \lvert A'(\alpha)\rvert\}$	$\langle m, n\rangle \vdash_{(A\times A')(\alpha)} (a, a')$ if and only if $m \vdash_{A(\alpha)} a \wedge n \vdash_{A'(\alpha)} a'$

Notice that $\sigma_\Gamma(A, B)(\alpha) = \sigma(A(\alpha), B(\alpha, \_))$ and $\pi_\Gamma(A, B)(\alpha) = \pi(A(\alpha), B(\alpha, \_))$.

Fig.1. Some constructions in ω-Set.

If A is an ω-set, we write $|A|$ for its underlying set. The category ω-Set is locally cartesian closed with finite colimits[23]. Fig.1 summarizes some standard constructions. Perhaps more importantly, ω-Set contains a remarkable internal category[24,25].

Definition 3.2. *A* per*, partial equivalence relation, is a transitive symmetric relation A on the set ω of natural numbers. We write mAn for $(m, n) \in A$. $Q(A)$ is the set of equivalence classes of A. $[n]_A$ is the equivalence class represented by n. $Dom(A) \stackrel{\text{def}}{=} \{n|nAn\}$. $n \in A$ will mean $n \in Dom(A)$. A map from a per A to another per B is the set $[l]_{A\rightarrow B}$ of all natural numbers such that $\forall m, n \in [l]_{A\rightarrow B} . \forall p, q \in Dom(A) . pAq \Rightarrow (m \cdot p)B(n \cdot q)$.*

Given pers A and B, we can define the usual constructions as follows. The product $A \times B$ is the per such that $m(A \times B)n$ iff $\pi_0 m A \pi_0 n$ and $\pi_1 m B \pi_1 n$. The exponential $A \rightarrow B$ contains all the pairs $\langle m, n\rangle$ such that for any pair $\langle a, b\rangle \in A$, $\langle m \cdot a, n \cdot b\rangle \in B$. The coproduct $A + B$ contains all the pairs $\langle m, n\rangle$ such that either $\pi_0 m = \pi_0 n = 0$ and $\pi_1 m A \pi_1 n$ or $\pi_0 m = \pi_0 n = 1$ and $\pi_1 m B \pi_1 n$. The initial object is the empty relation while the terminal object is the total relation. In fact, the category PER is a locally cartesian closed category with finite colimits. But what makes PER particularly useful is the fact that it is a small complete full subcategory of ω-SET. This property enables us to interpret the *Calculus of Constructions* in ω-SET: types are interpreted as ω-sets and propositions are interpreted as pers.

Therefore to show that the *Enriched Calculus of Constructions* can be modeled in ω-SET, we only have to explain how the generalized inductive types are interpreted in this category. In the ω-SET model, a context is interpreted as an ω-set, a judgement $\Gamma \vdash A : Type$ is modeled by a map from the underlying set of Γ to ω-SET. The empty context is interpreted by the terminal ω-set $(\{\star\}, \omega \times \{\star\})$. The context $\Gamma, x : A$ is interpreted by $\sigma(\Gamma, A)$, where $A : |\Gamma| \longrightarrow \omega\text{-SET}$ is the denotation of $\Gamma \vdash A : Type$. $\Gamma \vdash \Pi x : A.B : Type$ is modeled by $\pi_\Gamma(A, B) : |\Gamma| \longrightarrow \omega\text{-SET}$. And the denotation of $\Gamma \vdash [x_1 : A_1, \ldots, x_n : A_n]$ *Ctxt* is given by $\sigma_\Gamma(A_1, \sigma_{\sigma(\Gamma,A_1)}(A_2, \ldots, \sigma_{\sigma(\cdots\sigma(\Gamma,A_1),\ldots,A_{n-2})}(A_{n-1}, A_n)\ldots))$. A term $\Gamma \vdash a : A$ is interpreted by a morphism $a : \Gamma \longrightarrow \sigma(\Gamma, A)$ that satisfies the *first projection property*: $a; \pi_1 = Id_\Gamma$ where $\pi_1 : \sigma(\Gamma, A) \longrightarrow \Gamma$ is the first projection. For details, see *loc.cit.*

4 The Model

Definition 4.1. *A rule on the set U is a pair $\frac{u}{v}$ such that $u \subset U$ and $v \in U$. A rule set on U is a set of rules on U. Given a rule set R on U, a set A is R-closed if for any rule $\frac{u}{v} \in R$, $u \subset A$ implies $v \in A$. The set inductively defined by R is the set $\mathcal{I}(R) \stackrel{\text{def}}{=} \bigcap\{A | A \text{ is } R\text{-closed}\}$, the smallest R-closed set. A rule set R is deterministic if $\frac{u_1}{v} \in R \wedge \frac{u_2}{v} \in R \Rightarrow u_1 = u_2$.*

Associated with each rule set R is the R-induction: suppose P is a property; if for every rule $\frac{u}{v} \in R$, $\forall x \in u.P(x)$ implies $P(v)$, then P holds for every member of $\mathcal{I}(R)$. This is a natural generalization of the transfinite induction in set theory[26]. The corresponding generalization of definition by recursion is also available — to define a function on $\mathcal{I}(R)$, we do it in the way the elements of $\mathcal{I}(R)$ are generated.

Usually there is a more familiar way of obtaining what we want. We start from the empty set and keep throwing new elements into it. The transfinite sequence is bounded upwards often for simple cardinality reason. Therefore the sequence closes at some ordinal and the set obtained at this stage is $\mathcal{I}(R)$. It is obviously R-closed; on the other hand, each element of the set must be in all R-closed sets.

Let's look at the formation rule first. By assumption, $\Delta_i : |\Gamma| \longrightarrow \omega\text{-SET}$ for $i \in [1 \cdot\cdot n]$. For each $\alpha \in |\Gamma|$, $\Phi_{ik}(\alpha, \_) : |\Delta_i(\alpha)| \longrightarrow \omega\text{-SET}$ for $k \in [1 \cdot\cdot n_i]$. Let V_κ be a sufficiently large universe. Define R_α to be the following rule set on V_κ:

$$\bigcup_{i \in [1\cdot\cdot n]} \left\{ \frac{\bigcup_{k \in [1\cdot\cdot n_i]} range(\phi_{ik})}{\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle} \;\middle|\; \begin{array}{l} \delta_i \in |\Delta_i(\alpha)| \wedge \\ \phi_{i1} \in |\Phi_{i1}(\alpha, \delta_i)| \to V_\kappa \\ \wedge \cdots \wedge \\ \phi_{in_i} \in |\Phi_{in_i}(\alpha, \delta_i)| \to V_\kappa \end{array} \right\}$$

where $n_{intro_i} \in \omega$ is a code for $intro_i$ and $range(\phi_{ik})$ is the range of the function ϕ_{ik}. What are the requirements on κ? The following analysis is again taken from [18]:

- For $\kappa \in \omega$, V_κ is finite. So κ must be no less than ω.
- V_κ should be closed under pairing. Because V_β is the greatest element in $V_{\beta+1}$, $\langle V_\beta, V_\beta\rangle \notin V_{\beta+1}$. We conclude that κ must be a limit ordinal.
- $|\Phi_{ij}(\alpha, \delta_i)| \to V_\kappa \subset V_\kappa$ for all possible i and j. This suggests that κ is regular.

If $\kappa' > \kappa$, we obtain another rule set R' on $V_{\kappa'}$ by replacing V_κ by $V_{\kappa'}$ in the above definition.

Proposition 4.2. $\mathcal{I}(R_\alpha) = \mathcal{I}(R')$.

Proof. Clearly $R_\alpha \subset R'$. Suppose $\frac{u}{v} \in R'$ and $u \subset \mathcal{I}(R_\alpha)$. Then $u = \cup_{k\in[1\cdot\cdot n_i]} range(\phi_{ik})$ by definition. So $u = \cup_{k\in[1\cdot\cdot n_i]} range(\phi_{ik}) \subset \mathcal{I}(R_\alpha) \subset V_\kappa$. Therefore $\phi_{ik} \in |\varPhi_{ik}(\alpha,\delta_i)| \to V_\kappa$ for $k \in [1\cdot\cdot n_i]$, $\alpha \in |\Gamma|$ and $\delta_i \in |\varDelta_i(\alpha)|$. It follows that $\frac{u}{v} \in R_\alpha$. It is concluded that $v \in \mathcal{I}(R_\alpha)$. That is $\mathcal{I}(R_\alpha)$ is R'-closed. On the other hand, if B is R'-closed, then by R_α-induction, it is easy to show that $\mathcal{I}(R_\alpha) \subset B$. Therefore $\mathcal{I}(R_\alpha)$ is the least R'-closed set. □

The next proposition collects some properties of R_α and $\mathcal{I}(R_\alpha)$, which illustrates part of the reason why $\mathcal{I}(R_\alpha)$ models the inductive type.

Proposition 4.3. (1) *R_α is deterministic.* (2) *If $x \in \mathcal{I}(R_\alpha)$, then $x = \langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle$ for some $i \in [1\cdot\cdot n_i]$, $\delta_i \in |\varDelta_i(\alpha)|$ and $\phi_{ik} \in |\varPhi_{ik}(\alpha,\delta_i)| \to V_\kappa$ where $k \in [1\cdot\cdot n_i]$.* (3) *Suppose $\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle \in \mathcal{I}(R_\alpha)$. Then there exists a unique rule $\frac{u}{\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle} \in R_\alpha$. Besides, $u = \cup_{k\in[1\cdot\cdot n_i]} range(\phi_{ik})$.* (4) *Suppose $\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle \in \mathcal{I}(R_\alpha)$. Then ϕ_{ik} ($k \in [1\cdot\cdot n_i]$) is a function from $|\varPhi_{ik}(\alpha,\delta_i)|$ to $\mathcal{I}(R_\alpha)$.*

Proof. (1) This is just the extensionality of functions. (2) The set of all conclusions of the rules in R_α is R_α-closed. (3) Immediate from (1), (2) and the definition of R_α. (4) Suppose $\cup_{k\in[1\cdot\cdot n_i]} range(\phi_{ik}) \not\subseteq \mathcal{I}(R_\alpha)$. There must exist some $s \in \cup_{k\in[1\cdot\cdot n_i]} range(\phi_{ik})$ that is not in $\mathcal{I}(R_\alpha)$. Consider $\mathcal{I}(R_\alpha)\setminus\{\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle\}$. For any $\frac{u}{v} \in R_\alpha$ such that $u \subset \mathcal{I}(R_\alpha)\setminus\{\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle\}$, $u \subset \mathcal{I}(R_\alpha)$. So $v \in \mathcal{I}(R_\alpha)$. But v cannot be $\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle$ for otherwise $u = \cup_{k\in[1\cdot\cdot n_i]} range(\phi_{ik})$ by (3), which is impossible because s is in $\cup_{k\in[1\cdot\cdot n_i]} range(\phi_{ik})$ but certainly not in u. It follows that $v \in \mathcal{I}(R_\alpha)\setminus\{\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle\}$ and the latter is R_α-closed. It is a contradiction. Hence $\cup_{k\in[1\cdot\cdot n_i]} range(\phi_{ik}) \subset \mathcal{I}(R_\alpha)$. □

We are now going to construct an ω-set $\omega(\mathcal{I}(R_\alpha))$ from $\mathcal{I}(R_\alpha)$. The construction proceeds by first constructing a series of functions $r^\beta : \mathcal{I}(R_\alpha) \longrightarrow P(\omega)$, where β is an ordinal and $P(\omega)$ the power set of ω, and a series of ω-sets $\omega^0 \subset \omega^1 \subset \cdots$ derived from the functions.

1) The function r^0 associates an empty set with each element of $\mathcal{I}(R_\alpha)$. $\omega^0 \stackrel{\text{def}}{=} (\emptyset,\emptyset)$.

2) $r^{\beta+1}(\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle) \stackrel{\text{def}}{=} S_i$ where

$$S_i \stackrel{\text{def}}{=} \left\{ \langle n_{intro_i}, n_{\delta_i}, n_{\phi_{i1}}, \ldots, n_{\phi_{in_i}}\rangle \;\middle|\; \begin{array}{l} \delta_i \in |\varDelta_i(\alpha)| \wedge \\ \phi_{i1} \in |\varPhi_{i1}(\alpha,\delta_i) \to \omega^\beta| \\ \wedge\cdots\wedge \\ \phi_{in_i} \in |\varPhi_{in_i}(\alpha,\delta_i) \to \omega^\beta| \\ \wedge n_{\delta_i} \vdash_{\varDelta_i(\alpha)} \delta_i \\ \wedge n_{\phi_{i1}} \vdash_{\varPhi_{i1}(\alpha,\delta_i)\to\omega^\beta} \phi_{i1} \\ \wedge\cdots\wedge \\ n_{\phi_{in_i}} \vdash_{\varPhi_{in_i}(\alpha,\delta_i)\to\omega^\beta} \phi_{in_i} \end{array} \right\}.$$

$\omega^{\beta+1} \stackrel{\text{def}}{=} (|\omega^{\beta+1}|, \vdash_{\omega^{\beta+1}})$ where $x \in |\omega^{\beta+1}|$ iff $r^{\beta+1}(x) \neq \emptyset$, $n \vdash_{\omega^{\beta+1}} x$ iff $n \in r^{\beta+1}(x)$.

3) For a limit ordinal θ, let

$$r^\theta(x) \stackrel{\text{def}}{=} \cup_{\beta<\theta} r^\beta(x) \quad \text{for all } x \in \mathcal{I}(R_\alpha),$$

$$\omega^\theta \stackrel{\text{def}}{=} (\cup_{\beta<\theta}|\omega^\beta|, \cup_{\beta<\theta} \vdash_{\omega^\beta}).$$

Given $s,t : \mathcal{I}(R_\alpha) \longrightarrow P(\omega)$, define $s \le t$ iff $\forall x \in \mathcal{I}(R_\alpha).s(x) \subset t(x)$. Obviously, for ordinals $\beta' \in \beta''$, $r^{\beta'} \le r^{\beta''}$. By simple cardinality argument, we know that there exists an ordinal β such that for all $\theta \ge \beta$, $r^\theta = r^\beta$. By the least ordinal theorem[26], we can choose the least such β. Define $\omega(\mathcal{I}(R_\alpha))$ to be ω^β. We can now interpret $\mu(\varDelta_1,\ldots,\varDelta_n).[\varPhi_{11},\ldots,\varPhi_{1n_1};\ldots;\varPhi_{n1},\ldots,\varPhi_{nn_n}]$ by $\lambda\alpha : |\Gamma|.\omega(\mathcal{I}(R_\alpha))$. The interpretation of the elimination rule will be accommodated by Proposition 4.5.

The eliminators on $\mu(\boldsymbol{\varDelta}).[\boldsymbol{\varPhi}]$ should usually be interpreted in the way the elements of $|\omega(\mathcal{I}(R_\alpha))|$ are generated. This can be done because of the next proposition.

Proposition 4.4. *Let $\dot{R}_\alpha$ be obtained from R_α by removing all the rules whose conclusions are in $\mathcal{I}(R_\alpha) \setminus |\omega(\mathcal{I}(R_\alpha))|$. Then $|\omega(\mathcal{I}(R_\alpha))| = \mathcal{I}(\dot{R}_\alpha)$.*

Proof. (1) $|\omega(\mathcal{I}(R_\alpha))|$ is $\dot{R}_\alpha$-closed. Suppose $\frac{u}{v} \in \dot{R}_\alpha$ and $u \subset |\omega(\mathcal{I}(R_\alpha))|$. Clearly $v \in \mathcal{I}(R_\alpha)$. If $v \notin |\omega(\mathcal{I}(R_\alpha))|$, then $v \in \mathcal{I}(R_\alpha) \setminus |\omega(\mathcal{I}(R_\alpha))|$. But this would imply that $\frac{u}{v} \notin \dot{R}_\alpha$. (2) Suppose A is $\dot{R}_\alpha$-closed. By induction in the way the elements of $|\omega(\mathcal{I}(R_\alpha))|$ are constructed, one can easily show that $\omega^\beta \subset A$ for all ordinal β. □

We have defined the ω-set that is the interpretation of the inductive type. We need also of course to interpret the elimination and computation rules. For that purpose, we define another rule set whose least closed set is the graph of the underlying function modeling the eliminator. Suppose $\alpha \in |\Gamma|$ and $\delta_i \in |\Delta_i(\alpha)|$. By assumption, $C : |\sigma(\Gamma, \lambda\alpha : |\Gamma|.\omega(\mathcal{I}(R_\alpha)))| \longrightarrow \omega$-SET and $\phi_{i1} : |\Phi_{i1}(\alpha, \delta_i)| \longrightarrow |\omega(\mathcal{I}(R_\alpha))|$. Therefore $C[(\alpha, \_)] : |\omega(\mathcal{I}(R_\alpha))| \longrightarrow \omega$-SET. Write $C[(\alpha, \phi_{i1}(\_))]$ for the composition of ϕ_{i1} and $C[(\alpha, \_)]$. F_α is defined as follows:

$$\bigcup_{i\in[1\cdot\cdot n]} \left\{ \frac{\bigcup_{k\in[1\cdot\cdot n_i]} \{\langle \phi_{ik}(x), h_{ik}(x)\rangle \mid x \in |\Phi_{ik}(\alpha,\delta_i)|\}}{\langle\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle, d_i(\alpha)(\delta_i, \phi_{i1}, h_{i1}, \ldots, \phi_{in_i}, h_{in_i})\rangle} \;\middle|\; \begin{array}{l} \delta_i \in |\Delta_i(\alpha)| \wedge \\ \phi_{i1} \in |\Phi_{i1}(\alpha,\delta_i) \to \omega(\mathcal{I}(R_\alpha))| \\ h_{i1} \in |\pi(\Phi_{i1}(\alpha,\delta_i), C[(\alpha, \phi_{i1}(\_))])| \\ \wedge \cdots \wedge \\ \phi_{in_i} \in |\Phi_{in_i}(\alpha,\delta_i) \to \omega(\mathcal{I}(R_\alpha))| \\ h_{in_i} \in |\pi(\Phi_{in_i}(\alpha,\delta_i), C[(\alpha, \phi_{in_i}(\_))])| \end{array} \right\}.$$

Proposition 4.5. *$\mathcal{I}(F_\alpha)$ is the graph of a function in $|\pi(\omega(\mathcal{I}(R_\alpha)), C[(\alpha, \_)])|$.*

Proof. By induction in the way the elements of $\omega(\mathcal{I}(R_\alpha))$ are generated, we will construct a collection of sets $G^0 \subset G^1 \subset \cdots \subset \mathcal{I}(F_\alpha)$. Suppose we have constructed the sets up to β such that they satisfy two conditions:

(i) G^θ $(\theta \le \beta)$ is the graph of some function whose domain is a subset ψ^θ of $|\omega(\mathcal{I}(R_\alpha))|$ and for each element x of ψ^θ, $\{G^\theta\}(x) \in |C[(\alpha, x)]|$.

(ii) If $\langle\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle, d_i(\alpha)(\delta_i, \phi_{i1}, h_{i1}, \ldots, \phi_{in_i}, h_{in_i})\rangle \in G^{\theta+1}$, then for $k \in [1\cdot\cdot n_i]$, the equation $h_{ik} = \lambda x : |\Phi_{ik}(\alpha,\delta_i)|.\{G^\theta\}(\phi_{ik}(x))$ holds.

Now suppose the following rule is in F_α, and $\bigcup_{k\in[1\cdot\cdot n_i]} \{\langle \phi_{ik}(x), h_{ik}(x)\rangle \mid x \in |\Phi_{ik}(\alpha,\delta_i)|\} \subset G^\beta$.

$$\frac{\bigcup_{k\in[1\cdot\cdot n_i]} \{\langle \phi_{ik}(x), h_{ik}(x)\rangle \mid x \in |\Phi_{ik}(\alpha,\delta_i)|\}}{\langle\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle, d_i(\alpha)(\delta_i, \phi_{i1}, h_{i1}, \ldots, \phi_{in_i}, h_{in_i})\rangle}$$

(1) By definition, $\langle\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle, d_i(\alpha)(\delta_i, \phi_{i1}, h_{i1}, \ldots, \phi_{in_i}, h_{in_i})\rangle \in \mathcal{I}(F_\alpha)$. By the assumption (i), we know that for every $x \in |\Phi_{ik}(\alpha,\delta_i)|$, $h_{ik}(x) = \{G^\beta\}(\phi_{ik}(x))$ for $k \in [1\cdot\cdot n_i]$. The extensionality enables us to conclude that $h_{ik} = \lambda x : |\Phi_{ik}(\alpha,\delta_i)|.\{G^\theta\}(\phi_{ik}(x))$ for $k \in [1\cdot\cdot n_i]$.

(2) By the assumption on G^β, (1) and the definition of d_i $(i \in [1\cdot\cdot n])$, it is obvious that

$$d_i(\alpha)(\delta_i, \phi_{i1}, h_{i1}, \ldots, \phi_{in_i}, h_{in_i}) \in |C[(\alpha, \langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle)]|.$$

(3) Suppose $\langle\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle, d_i(\alpha)(\delta_i, \phi_{i1}, h'_{i1}, \ldots, \phi_{in_i}, h'_{in_i})\rangle \in G^\theta$, $\theta \le \beta$, then by induction hypothesis,

$$h'_{ik} = \lambda x : |\Phi_{ik}(\alpha,\delta_i)|.\{G^\theta\}(\phi_{ik}(x)) = \lambda x : |\Phi_{ik}(\alpha,\delta_i)|.\{G^\beta\}(\phi_{ik}(x)) = h_{ik}$$

for every $k \in [1\cdot\cdot n_i]$.

From (1), (2) and (3), we conclude that

$$G^{\beta+1} \stackrel{\text{def}}{=} \{\langle\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle, d_i(\alpha)(\delta_i, \phi_{i1}, h_{i1}, \ldots, \phi_{in_i}, h_{in_i})\rangle\} \cup G^\beta$$

is a subset of $\mathcal{I}(F_\alpha)$ and satisfies the conditions (i) and (ii). The case for a limit ordinal is handled as usual. We therefore have a sequence

$$G^0 \subset G^1 \subset \cdots \subset G^\beta \subset G^{\beta+1} \subset \cdots.$$

The cardinality restriction implies that the sequence must close at some stage, say, β. But G^β is clearly F_α-closed. So $G^\beta = \mathcal{I}(F_\alpha)$. That is $\mathcal{I}(F_\alpha)$ is the graph of some function.

To show that $\mathcal{I}(F_\alpha)$ is in $|\pi(\omega(\mathcal{I}(R_\alpha)), C[(\alpha, \_)])|$, we must show that it is tracked by some recursive function. Suppose $r \vdash_\Gamma \alpha$. Define a function as follows:

case $(m)_0$ **of**
$n_{intro_1} : (n_{d_1} \cdot r) \cdot ((m)_1, (m)_2, \Lambda l.f \cdot ((m)_2 \cdot l), \ldots, (m)_{n_1+1}, \Lambda l.f \cdot ((m)_{n_1+1} \cdot l));$
$\vdots$
$n_{intro_n} : (n_{d_n} \cdot r) \cdot ((m)_1, (m)_2, \Lambda l.f \cdot ((m)_2 \cdot l), \ldots, (m)_{n_n+1}, \Lambda l.f \cdot ((m)_{n_n+1} \cdot l))$
end

$\Lambda l.f \cdot ((m)_2 \cdot l) \cdots$ are obtained by s-m-n theorem. Here are some observations. (i) By Church thesis, the function is effective on the variables f and m. (ii) It is effective to construct the above function from Γ, the reason being that given $\alpha \in |\Gamma|$ and $r \vdash_\Gamma \alpha$, one can obtain $n_{d_1} \cdot r, \ldots, n_{d_n} \cdot r$ effectively from $n_{d_1}, \ldots, n_{d_n}$. (iii) As $(\omega, \cdot)$ is a partial combinatory algebra, there exists an effective procedure that computes an index e such that $e \cdot (f, m) \simeq$ the above function. Here $\simeq$ is defined as follows: $x \simeq y$ iff $\forall n.(x \cdot n \downarrow \Leftrightarrow y \cdot n \downarrow) \wedge (x \cdot n \downarrow \Rightarrow x \cdot n = y \cdot n)$. Using s-m-n theorem again, we can compute a Gödel index $\Lambda f \Lambda m.e \cdot (f, m)$ for $\lambda f \lambda m.e \cdot (f, m)$ from e. The recursion theorem then says that there is an (index of an) effective function $R(\alpha)$ such that the following holds:

$$R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \simeq \Lambda m.e \cdot (R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)), m) \tag{1}$$

Notice that by (i), (ii), (iii) and the effectiveness of the recursion theorem, $R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m))$ is effectively computed. This point is important for the interpretation of the eliminators.

There remains one important thing to be proved — for any $c \in |\omega(\mathcal{I}(R_\alpha))|$ and any $n \vdash_{\omega(\mathcal{I}(R_\alpha))} c$, $R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \cdot n$ is defined and

$$R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \cdot n \vdash_{C[\alpha,c]} \mathcal{I}(F_\alpha)(c). \tag{2}$$

This is proved by induction in the way the elements of $\omega(\mathcal{I}(R_\alpha))$ are generated. An element of $|\omega(\mathcal{I}(R_\alpha))|$ must be of the form $\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i} \rangle$ and any of its realizers must be, by definition, of the form $\langle n_{intro_i}, n_{\delta_i}, n_{\phi_{i1}}, \ldots, n_{\phi_{in_i}} \rangle$. So we have to show that

$$\begin{aligned} & R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \cdot \langle n_{intro_i}, n_{\delta_i}, n_{\phi_{i1}}, \ldots, n_{\phi_{in_i}} \rangle \\ & \vdash_{C[(\alpha, \langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i} \rangle)]} \mathcal{I}(F_\alpha)(\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i} \rangle). \end{aligned} \tag{3}$$

Suppose it is added to the definition of $\omega(\mathcal{I}(R_\alpha))$ at the stage $\beta + 1$. Then by (1)

$$\begin{aligned} & R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \cdot \langle n_{intro_i}, n_{\delta_i}, n_{\phi_{i1}}, \ldots, n_{\phi_{in_i}} \rangle \\ \simeq & (\Lambda m.e \cdot (R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)), m)) \cdot \langle n_{intro_i}, n_{\delta_i}, n_{\phi_{i1}}, \ldots, n_{\phi_{in_i}} \rangle \\ \simeq & (n_{d_i} \cdot r) \cdot (n_{\delta_i}, n_{\phi_{i1}}, \Lambda l.R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \cdot (n_{\phi_{i1}} \cdot l), \ldots, \\ & \quad n_{\phi_{in_i}}, \Lambda l.R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \cdot (n_{\phi_{in_i}} \cdot l)) \end{aligned}$$

We want to show that the expression at the third and forth lines of the above partial equation is defined. We know that $\phi_{i1} \in |\Phi_{i1}(\alpha, \delta_i) \to \omega^\beta|$ implies $\phi_{i1} \in |\Phi_{i1}(\alpha, \delta_i) \to \omega(\mathcal{I}(R_\alpha))|$. So by the structural definition of the interpretation on the elimination (and computation) rule(s), $n_{d_i} \cdot r$ is defined on $n_{\delta_i}, n_{\phi_{i1}}, \ldots, n_{\phi_{in_i}}$. Now if $l \vdash_{\Phi_{i1}(\alpha,\delta_i)} \phi$, then $n_{\phi_{i1}} \cdot l \vdash_{\omega^\beta} \phi_{i1}(\phi)$, which implies $n_{\phi_{i1}} \cdot l \vdash_{\omega(\mathcal{I}(R_\alpha))} \phi_{i1}(\phi)$. This is added to the definition of $\omega(\mathcal{I}(R_\alpha))$ in a stage prior to $\beta + 1$. So by induction hypothesis $R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \cdot (n_{\phi_{i1}} \cdot l)$ is defined and $R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \cdot (n_{\phi_{i1}} \cdot l) \vdash_{C[\alpha, \phi_{i1}(\phi)]} \mathcal{I}(F_\alpha)(\phi_{i1}(\phi))$. It is concluded that $\Lambda l.R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m))(n_{\phi_{i1}} \cdot l) \vdash_{\pi(\Phi_{i1}(\alpha,\delta_i), C[(\alpha, \phi_{i1}(\_))])} \lambda \phi.\mathcal{I}(F_\alpha)(\phi_{i1}(\phi))$. Therefore $n_{d_i} \cdot r$ is defined at $\Lambda l.R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m)) \cdot (n_{\phi_{i1}} \cdot l)$. It follows that $R(\alpha)(\Lambda f \Lambda m.e \cdot (f, m))$

is defined at $\langle n_{intro_i}, n_{\delta_i}, n_{\phi_{i1}}, \ldots, n_{\phi_{in_i}}\rangle$. By the assumption, the definitions of $\omega^{\beta+1}$ and $\mathcal{I}(F_\alpha)$ and the property (ii) that $\mathcal{I}(F_\alpha)$ satisfies, we have

$$\begin{aligned}
&(n_{d_i}\cdot r)\cdot(n_{\delta_i}, n_{\phi_{i1}}, \Lambda l.R(\alpha)(\Lambda f\Lambda m.e\cdot(f,m))\cdot(n_{\phi_{i1}}\cdot l), \ldots, n_{\phi_{in_i}},\\
&\Lambda l.R(\alpha)(\Lambda f\Lambda m.e\cdot(f,m))\cdot(n_{\phi_{in_i}}\cdot l))\\
&\vdash_{C[\alpha,\langle n_{intro_i},\delta_i,\phi_{i1},\ldots,\phi_{in_i}\rangle]}\\
&d_i(\alpha)(\delta_i, \phi_{i1}, \lambda\phi.\mathcal{I}(F_\alpha)(\phi_{i1}(\phi)), \ldots, \phi_{in_i}, \lambda\phi.\mathcal{I}(F_\alpha)(\phi_{in_i}(\phi)))\\
&= \mathcal{I}(F_\alpha)(\langle n_{intro_i}, \delta_i, \phi_{i1}, \ldots, \phi_{in_i}\rangle).
\end{aligned}$$

We conclude that (3) holds. That is $\mathcal{I}(F_\alpha)$ is tracked by $R(\alpha)(\Lambda f\Lambda m.e\cdot(f,m))$. □

Theorem 4.6. *The above construction gives a sound interpretation to inductive types in the Calculus of Constructions.*

Proof. The generalized inductive type $\mu(\boldsymbol{\Delta}).[\boldsymbol{\Phi}]$ is interpreted as $\lambda\alpha : |\Gamma|.\omega(\mathcal{I}(R_\alpha)) : |\Gamma| \longrightarrow \omega\text{-}\mathrm{SET}$. The eliminator is taken care of by the effective function $\lambda\alpha.(\alpha, \mathcal{I}(F_\alpha)) : \Gamma \longrightarrow \sigma(\Gamma, \pi_\Gamma(\lambda\alpha{:}|\Gamma|.\omega(\mathcal{I}(R_\alpha)), C))$ and the equation rule is sound by the very definition of $R(\alpha)(\Lambda f\Lambda m.e\cdot(f,m))$ and (1). □

Example 4.7. Some of the basic sets in Martin-Löf's set theory can be defined as follows:

$$\begin{aligned}
Unit &\stackrel{\text{def}}{=} \mu([]).[]\\
A+B &\stackrel{\text{def}}{=} \mu([x:A],[y:B]).[]\\
N &\stackrel{\text{def}}{=} \mu([],[]).[;[]]\\
List(A) &\stackrel{\text{def}}{=} \mu([],[x:A]).[[];[]]\\
Ord &\stackrel{\text{def}}{=} \mu([],[],[O:Type]).[;[];[O:Type]]\\
\Sigma x:A.B &\stackrel{\text{def}}{=} \mu([x:A,y:B]).[]\\
Wx:A.B &\stackrel{\text{def}}{=} \mu([x:A]).[[y:B]].
\end{aligned}$$

Notice the difference between an empty context, which is inhabited by (), and no context at all. If we apply the previous constructions to $\mu([],[]).[;[]]$, what we get is as follows: [] is interpreted as $(\{\star\}, \omega\times\{\star\})$. Suppose we code the $intro_0$ and $intro_1$ by 0 and 1 respectively and simplify a function $\{\star\} \longrightarrow V$ to an element of V. Then the rule set is

$$R_N \stackrel{\text{def}}{=} \left\{\frac{}{\langle 0,\star\rangle}\right\} \cup \left\{ \frac{\{\langle\overbrace{1,\ldots,1}^{i}, \langle 0,\star\rangle\cdots\rangle\}}{\langle\underbrace{1,\ldots,1}_{i+1}, \langle 0,\star\rangle\cdots\rangle} \;\middle|\; i\in\omega \right\}.$$

The smallest R_N-closed set is $N \stackrel{\text{def}}{=} \{\langle\overbrace{1,\ldots,1}^{i}, \langle 0,\star\rangle\cdots\rangle | i\in\omega\}$. We have to effectivize the set obtained. N_0 is the initial ω-set. N_1 is the ω-set $(\{\langle 0,\star\rangle\}, \omega\times\{\langle 0,\star\rangle\})$. N_2 is the ω-set $(\{\langle 0,\star\rangle, \langle 1,\langle 0,\star\rangle\rangle\}, \omega\times\{\langle 0,\star\rangle, \langle 1,\langle 0,\star\rangle\rangle\})$ and so on. It is concluded that $\omega(N)$ is $(N, \omega\times N)$. So according to our interpretation, the denotation of $\mu([],[]).[;[]]$ is (isomorphic to) the standard natural number object in ω-SET.

Similarly the interpretations of the first, second and sixth of the above defined types are isomorphic to those given by the well-established interpretation.

5 Models of Some Other Variants

We now briefly discuss models of inductive types in related type systems.

5.1 Inductive Families

In [18], a class of types, that of inductive families, is introduced, which bear some resemblance to the combination of the two variants of tree types proposed in [27]. The ω-set semantics of the inductive families can be similarly given.

5.2 The Recursive Model

The method described in Section 4 can also be applied to PER. What we get is a model of Martin-Löf's set theory extended with the generalized inductive types. To model universes, we must associate a name with each per that is the interpretation of some type in ML^{∞}, the Martin-Löf's set theory with an infinite hierarchy of universes. Then a type appearing on the left hand side of ':' is interpreted as a name; and the same type is modeled by the corresponding per when it appears on the right side of ':'. This is basically the extension of the recursive model given in [28, 29].

In the per-setting, two points are worth mentioning: (i) because of the double role a type plays in Martin-Löf's set theory with small universes, there is a tremendous amount of encoding involved; if we want to model the eliminators for the universes, the problem is even more severe; (ii) the universe V_κ in the definition of the rule sets can always be taken to be the terminal per $\omega\times\omega$, which is the largest element in the set $(Obj(\text{PER}),\subseteq)$. So typically a rule set R_α is defined as follows:

$$\bigcup_{i\in[1\cdots n]}\left\{\frac{\bigcup_{k\in[1\cdots n_i]} range(\phi_{ik})}{\langle n_{intro_i},\delta_i,\phi_{i1},\ldots,\phi_{in_i}\rangle}\;\middle|\;\begin{array}{l}\delta_i\in\Delta_i(\alpha)\wedge\\ \phi_{i1}\in\Phi_{i1}(\alpha,\delta_i)\to\omega\times\omega\\ \wedge\cdots\wedge\\ \phi_{in_i}\in\Phi_{in_i}(\alpha,\delta_i)\to\omega\times\omega\end{array}\right\}.$$

Notice that in the above definition all the elements and the tuples are natural numbers. A transfinite induction should be used to define a partial equivalence relation on $\mathcal{I}(R_\alpha)$. This process produces a per $\mathcal{P}(\mathcal{I}(R_\alpha))$. For details, see [30–32].

5.3 Inductive Types Defined Using a Logical Framework

One of the advantages of using a logical framework is the possibility of separating the computational content of a language to be defined from the extensional properties of the meta calculus. The language proposed in [33] is designed to maximize this gain. To be able to define the generalized inductive types, the Martin-Löf's logical framework is extended with kind schemata. These schemata are 'small' in the sense that they only talk about the small universe *Set*. The generalized inductive types, which reside in *Set*, are then defined in terms of the kind schemata.

From a model theoretic point of view, *Set* is an internal category of the ambient category within which the logical framework is interpreted. So as usual we can interpret *Set* as the ω-SET and kinds as objects of the category Ω-SET of ω-sets defined on a large universe. The rule sets as defined in Section 4 live in Ω-SET. The 'smallness' of the kind schemata implies that the least closed sets of these rule sets are in ω-SET. For more about categorical semantics of logical frameworks, see [34–39].

5.4 Inductive Types via W-Types

It is well-known that inductive types are definable in an extensional dependent calculus with W-types[17,40]. The interpretation given in Section 4 is extensional. It follows that it also interprets the W-types with the extensional rules. So the extensional *Calculus of Constructions* enriched with W-types is accounted for by the method illustrated in Section 4.

5.5 Other Formulations of Generalized Inductive Types

Once the essence of the generalized inductive types is understood, different formulations should present no problems. [20] and [41] contain two other formulations. The advantage of the formulation in [20] is that it is easier to see how to generalize(!) the generalized inductive types. We call *general inductive types* those inductive types defined in [20] with strict positivity being relaxed to just positivity.

With little modification, our ω-SET model is a sound interpretation of the generalized inductive types formulated in [20]. Our method, however, does *not* give a model of the *general* inductive types in ω-SET. But it *does* give a model of the *general* inductive types in PER. This is due to the existence of a largest universe, as it were, in PER. For more details, see [30–32].

5.6 Internal Inductive Types

A totally different approach is to code up inductive types internally. This is adopted in [41–45]. This method does not pose any model theoretical problems in the traditional sense. The semantic interest here is about how to relate two levels of models. See [46, 47] for details.

5.7 Categorical Inductive Types

In [47], a categorical formulation of recursion is given. It is interesting to see what this general definition means in concrete categories in which dependent typed calculus can be modeled. See [30] for some examples.

References

[1] Howard W. The formulae-as-type notion of construction. In To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus and Formalism, Seldin J, Hindley J (eds.). Academic Press, 1980.
[2] Nordström B, Petersson K, Smith J. Programming in Martin-Löf's type theory — An introduction. International Series of Monographs on Computer Science 7, Oxford University Press, 1990.
[3] Coquand T. An Analysis of Girard's Paradox. In *Proceedings of the First Symposium on Logic in Computer Science,* IEEE Computer Science Press, 1986, pp.227–236.
[4] Luo Z. A higher-order calculus and theory abstraction. *Information and Computation,* Academic Press, 1991, 90: 107–137.
[5] Luo Z. An extended calculus of constructions [dissertation]. LFCS, University of Edinburgh, 1990.
[6] Russell B, Whitehead A. The Principia Mathematica. Cambridge University Press, 1912.
[7] Coquand T, Huet G. The calculus of constructions. *Information and Computation,* 1988, 76: 95–120.
[8] Coquand T. Mathematical investigations of a calculus of constructions. In *Logic and Computer Science,* Odifreddi P (eds.), Academic Press, 1990, pp.91–122.
[9] Fu Yuxi. Principles of type theory. *Journal of Software,* 1997, Supplement.
[10] Berardi S. Non-conservativity of Coquand's Calculus with Respect to Higher-Order Intuitionistic Logics, 1989.
[11] Luo Z. A problem of adequacy: Conservativity of calculus of constructions over higher-order logic. LFCS, University of Edinburgh, 1990.
[12] Church A. A formulation of the simple theory of types. *Journal of Symbolic Logic,* 1940, 5: 56–68.
[13] Aczel P. The Type Theoretic Interpretation of Constructive Set Theory. In *Logic Colloquium'77,* Macintyre A, Pacholski L, Paris J (eds.), North-Holland, 1978, pp.55–66.
[14] Aczel P. The type theoretic interpretation of constructive set theory: Inductive definitions. In *Logic, Methodology and Philosophy of Science, VII,* R Barcan Marcus, G J W Dorn, P Weingartner (eds.), North-Holland, 1986.
[15] Fu Yuxi. A Decidable Interpretation of Constructive Sets. In *The Proceedings of the Sixth Asian Logic Conference,* May 20–24, Beijing, 1996.
[16] Fu Yuxi. Constructive sets in computable sets. *Journal of Computer Science and Technology,* 1997, 12(5): 425–440.
[17] Goguen H, Luo Z. Inductive data types: Well-ordering types revisited. LFCS, University of Edinburgh, 1992.

[18] Dybjer P. Inductive Sets and Families in Martin-Löf Type Theory and Their Set-Theoretic Semantics. In *Proceedings of the First Workshop on Logical Frameworks,* Huet G, Plotkin G (eds.), Cambridge University Press, 1991, pp.280–306.
[19] Aczel P. An introduction to inductive definitions. In *Handbook of Mathematical Logic,* Barwise J (ed.), North-Holland, 1977, pp.739–782.
[20] Coquand T, Paulin-Mohring C. Inductively defined types. In *COLOG'88, Lecture Notes in Computer Science 417,* Springer, 1990, pp.50–66.
[21] Ore C. The extended calculus of constructions (ECC) with inductive types. *Information and Computation,* 1992, 99: 231–264.
[22] Rogers H. Theory of Recursive Functions and Effective Computability. McGraw Hill, 1967.
[23] Longo G, Moggi E. Constructive natural deduction and its 'modest' interpretation. In *Semantics of Natural and Computer Languages,* M.I.T. Press, 1990.
[24] Hyland M. The effective topose. In *L.E.J. Brouwer Centenary Symposium,* Troelstra A, van Dalen D (eds.), North-Holland, 1982, pp.165–216.
[25] Hyland M. A small complete category. *Annals of Pure and Applied Logic,* 1988, 40: 135–165.
[26] Levy A. Basic Set Theory. Springer, 1979.
[27] Petersson K, Synek D. A set constructor for inductive sets in Martin-Löf's type theory. In *Category Theory and Computer Science, Lecture Notes in Computer Science 389,* Pitt D, Rydeheard D *et al.* (eds.), Springer, 1989, pp.128–140.
[28] Beeson M. Recursive models of constructive set theories. *Annals of Mathematical Logic,* 1982, 23: 127–178.
[29] Beeson M. Foundations of Constructive Mathematics. Springer, 1985.
[30] Fu Yuxi. Some semantic issues in type theory [dissertation]. Department of Computer Science, Manchester University, England, 1992.
[31] Fu Yuxi. Topics in type theory. Technical Report, UMCS-92-11-1, Department of Computer Science, University Manchester, 1992.
[32] Fu Yuxi. Recursive models of general inductive types. *Fundamenta Informaticae,* 1996, 26: 115–131.
[33] Luo Z. A unifying theory of dependent types: The schematic approach. In *Logical Foundations of Computer Science–Tver'92,* Lecture Notes in Computer Science 620, Springer, 1992.
[34] Fu Yuxi. Internal typed calculi and frame languages. In *Proceedings of the International Workshop on Advanced Software Technology,* Shanghai, 1994, pp.153–166.
[35] Fu Yuxi. Equalities Defined in Logical Frameworks. In *The Proceedings of the Sixth Asian Logic Conference*, May 20-24, Beijing, 1996.
[36] Fu Yuxi, Song Zhexian, Sun Yongqiang. The syntax, pragmatics and semantics of logical framework — The syntax and pragmatics of a new logical framework. *Journal of Software*, 1996, Supplement: 224–231.
[37] Fu Yuxi. Categorical properties of logical frameworks. *Mathematical Structures in Computer Science*, Cambridge University Press, 1997, 7: 1–47.
[38] Fu Yuxi. The semantic model of logical framework. *Journal of Software*, 1997, Supplement.
[39] Fu Yuxi. Relative properties of frame language. *Journal of Computer Science and Technology*, 1999, 14(4): 320–327.
[40] Dybjer P. An inversion principle for Martin-Löf's type theory. In *Proceedings of the Workshop on Programming Logic*, PMG-report 54, Chalmer University of Technology, 1989.
[41] Pfenning F, Paulin-Mohring C. Inductively defined types in the calculus of constructions. In *Mathematical Foundations of Programming Semantics*, Lecture Notes in Computer Science 442, Springer-Verlag, 1991, pp.209–228.
[42] Wraith G. A note on categorical datatypes. In *Category Theory and Computer Science*, Lecture Notes in Computer Science 389, Springer, 1989, pp.213–223.
[43] Fu Yuxi. Encodings in polymorphism Revisited. Technical Report, UMCS-93-6-4, Department of Computer Science, University of Manchester, 1993.
[44] Fu Yuxi. Structures defined in polymorphism. *Journal of Computer Science and Technology*, 1998, 13(6): 579–587.
[45] Fu Yuxi. Internal Structures of Inductive Types, submitted for publication.
[46] Pitts A. Polymorphism is Set-theoretic Constructively. In *Category Theory and Computer Science*, Lecture Notes in Computer Science 283, Springer, 1987, pp.12–39.
[47] Mendler N. Predicative type universes and primitive recursion. In *Proceedings of the Sixth Symposium on Logic in Computer Science*, IEEE Computer Science Press, 1991, pp.173–184.

FU Yuxi is a professor at the Department of Computer Science of Shanghai Jiaotong University. He received his Ph.D. degree in 1992 from Manchester University, England. His current research interest covers concurrency theory, type theory and semantics.