ГОДИШНИК НА СОФИЙСКИЯ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
Книга 1 — Математика и механика
Том 90, 1996

ANNUAIRE DE L'UNIVERSITE DE SOFIA „ST. KLIMENT OHRIDSKI“

FACULTE DE MATHEMATIQUES ET INFORMATIQUE
Livre 1 — Mathématiques et Mecanique
Tome 90, 1996

A SCHEMATIC PROOF OF STRONG NORMALIZATION FOR THE SYSTEMS OF THE λ-CUBE*

MILENA STEFANOVA

This paper describes a set-theoretical argument for proving *Strong Normalization* (SN) for the systems of the so-called λ-cube. The argument is relatively simple and, moreover, flexible. It can be adapted to extensions of the systems considered, such as additional sorts, inductive types or sub-types.

Keywords: typed lambda calculus, normalization, inductive types
1991/95 Math. Subject Classification: 03B15, 03B40

1. INTRODUCTION

In the recent years a lot of attention has been paid to the property of Strong Normalization for second- and higher-order dependent type systems. The number of the existing SN-proofs can be informally divided into two groups:

- *'syntactically-oriented' proofs*—proofs which are based on mixed syntactical and semantical methods ([6, 5, 3, 14, 15]), and
- *'semantically-oriented' proofs*—pure semantical proofs ([1, 7, 8, 9, 16]).

Most of these proofs make use of the idea of interpreting all typable terms as elements of sets of strongly-normalizing terms. Further, one can prove that a typable term belongs to the interpretation of its type and thus it is strongly-normalizing. However, semantically-oriented proofs make use of fully-compositional

* Lecture presented at the Fourth Logical Biennial, Gjuletchitza, September 12–14, 1996.

models, while syntactically-oriented ones are based on models which disregard some of the dependencies in a typable term. This has several consequences for both the kinds SN-proofs.

In general, the syntactically-oriented proofs are relatively simple but lacking flexibility and modularity. It is not easy and sometimes impossible to adapt them to various extensions such as additional type-constructions, more universes or sorts or sub-types. Any extension of the systems requires reconsideration and significant changes in such a SN-proof (see, for example, [14, 15, 13]). Furthermore, the relative simplicity of such a syntactically-oriented proof is usually lost after adapting it to a richer system.

Semantically-oriented proofs are based on operational or denotational semantics of the system under consideration (see [7, 1, 8]). These proofs seem to be more flexible than the syntactically-oriented ones in the sense that they can be easily adapted to various extensions of the system in question. Furthermore, they suggest generic methods for normalization proofs of PTSs (see [8, 9, 1, 16]).

However, in order to obtain compositional interpretations, most of them introduce very complicated structures, which are difficult to be mapped intuitively to the corresponding type system. Most of them use a realizability or categorical semantics (see [1, 8]) instead of a naive set-theoretic semantics as in the syntactically-oriented proofs.

The SN-proof considered in this paper combines advantages of syntactically- and semantically-oriented proofs: simplicity, flexibility and genericity. It can be classified as semantically-oriented. It is based on a naive set-theoretical semantics and as so is similar to the syntactically-oriented proofs. The principal difference with them is that type-dependencies are not disregarded in the interpretations, i.e., the interpretations are fully-compositional. This is achieved by defining simultaneously the interpretations of types and their elements.

The benefits one gets from this proof are in general the same as those in [1, 8, 9] — extendibility to more powerful systems. However, it is still simpler to interpret new type-constructors and reductions in the present set-theoretical setting. The flexibility of the proof presented is shown by extending it to systems with inductive types. We treat the case of *Natural Numbers* in the last section.

2. BARENDREGT'S CUBE

In this section precise definitions of the pure type systems in Barendregt's cube are given (see also [2]).

Definition 2.1 (PTS-definition). A *system of Barendregt's cube* λS is a triple $\lambda S = \langle \mathcal{T}, \mathcal{R}, \mathbf{R} \rangle$ such that:

- $\mathcal{T}$ is a set of *pseudoterms* defined by the abstract syntax

$$\mathcal{T} := Var \mid \{*, \Box\} \mid \mathcal{T}\mathcal{T} \mid \lambda Var{:}\mathcal{T}.\mathcal{T} \mid \Pi Var{:}\mathcal{T}.\mathcal{T},$$

where $Var = Var^* \bigcup Var^\Box$ and Var^*, and $Var^\Box$ are infinite enumerable disjoint sets of *object* and *constructor variables*, respectively. The object variables will be denoted by the small Latin letters x, y, z (with or without subscripts) and the constructor variables — by the small Greek letters α, β, γ. When we do not want to make a distinction between object and constructor variables, we will use the small Latin letters u, v, w.

The notions of β-reduction and β-conversion are defined on $\mathcal{T}$ by the contraction rule

$$(\lambda v{:}T_1.T_2)T_3 \to_\beta T_2[T_3/v];$$

• $\mathcal{R}$ is the set of *rules* of the system λS and consists of ordered pairs (s_1, s_2), such that $s_1, s_2 \in \{*, \Box\}$ and $(*, *) \in \mathcal{R}$;

• **R** is the set of *derivation rules* of λS specified bellow.

$(axiom)$ $\qquad \vdash * : \Box,$

$$(var) \qquad \frac{T \vdash s :}{v{:}T \vdash v : T}, \qquad s \in \{*, \Box\}\ ,\ v \in Var^* \setminus FV(\Gamma),$$

$$(weak) \qquad \frac{T \vdash s : \quad M \vdash U :}{v{:}T \vdash M : U}, \qquad s \in \{*, \Box\}\ ,\ v \in Var^* \setminus FV(\Gamma),$$

$$(\prod) \qquad \frac{T \vdash s_1 : \quad v{:}T \vdash U : s_2}{\Pi v{:}T.U \vdash s_2 :}, \qquad (s_1, s_2) \in \mathcal{R}\ ,\ v \in Var^{s_1},$$

$$(\lambda) \qquad \frac{v{:}T \vdash M : U \quad \Pi v{:}T.U \vdash s :}{\lambda v{:}T.M \vdash \Pi v{:}T.U :}, \qquad s \in \{*, \Box\},$$

$$(app) \qquad \frac{M \vdash \Pi v{:}T.U : \quad N \vdash T :}{MN \vdash U[N/v] :},$$

$$(conv) \qquad \frac{M \vdash T : \quad U \vdash s :}{M \vdash U :} \quad T =_\beta U, \qquad s \in \{*, \Box\}.$$

The eight systems of the λ-cube are listed below (see Table 1) according to the sets of their rules. The set of (*typable*) *terms* of the system λS is defined by

$$\texttt{Terms} := \{T \in \mathcal{T} \mid \exists \Gamma, U\ (T \vdash U : \text{ or } U \vdash T :)\ \}.$$

It is convenient to divide the typable terms into subsets (see [2, 4]) in the following way:

$$\begin{aligned}
\texttt{Kind}(\lambda S) &:= \{A \in \mathcal{T} \mid \exists \Gamma\ (A \vdash \Box :)\ \}, \\
\texttt{Constr}(\lambda S) &:= \{C \in \mathcal{T} \mid \exists \Gamma, A\ (C \vdash A :) : \Box\ \}, \\
\texttt{Type}(\lambda S) &:= \{\sigma \in \mathcal{T} \mid \exists \Gamma\ (\sigma \vdash * :)\ \}, \\
\texttt{Obj}(\lambda S) &:= \{t \in \mathcal{T} \mid \exists \Gamma, \sigma\ (t \vdash \sigma :) : *\ \}.
\end{aligned}$$

Table 1. The systems of the λ-cube

System	$(*,*)$	$(*,\Box)$	$(\Box,*)$	$(\Box,\Box)$
$\lambda\rightarrow$	×	-	-	-
λP	×	×	-	-
$\lambda 2$	×	-	×	-
$\lambda P2$	×	×	×	-
$\lambda\underline{\omega}$	×	-	-	×
$\lambda P\underline{\omega}$	×	×	-	×
$\lambda\omega$	×	-	×	×
λC	×	×	×	×

We will skip the subscript S in the above notations when it is clear which is the system under consideration.

3. INFORMAL OUTLINE OF THE PROOF

Let λS be a system of the λ-cube. Classification of typable terms of λS into objects, types, constructors and kinds determines a hierarchical structure which will be called *type hierarchy* in the sequel (see Fig. 1(a)). The type hierarchy has a fine structure — it contains two sub-hierarchies: the one of types and the other of kinds (see Fig. 1(b)).

Intuitively, every type is the set of objects of this type, and every kind is the set of constructors typable with it. All of these four levels are connected by **Type** $\subseteq$ **Constr**.

The typable terms of the system λS are interpreted in levels according to their level in the type hierarchy. In fact, the type hierarchy is mapped into a set-theoretical hierarchy, which will be called *λS-hierarchy*. The carrier, or the bottom level of the λS hierarchy is simply the set $\mathcal{T}$ of pseudo-terms.

Each system λS of the λ-cube is determined by its PTS-specification and its derivation rules. There are two sorts in each of the systems of the λ-cube: one of types $(*)$ and another of kinds $(\Box)$. Suppose that these sorts are interpreted by the set-universes $\mathbf{U}_S^*$ and $\mathbf{U}_S^\Box$. The conditions which $\mathbf{U}_S^*$ and $\mathbf{U}_S^\Box$ should satisfy are determined by the rest of the specification of λS, i.e., by its axioms and PTS-rules.

There is one axiom for each λS of the λ-cube, namely, $* : \Box$. This corresponds to the requirement $\mathbf{U}_S^* \in \mathbf{U}_S^\Box$. Further, suppose that (s_1, s_2) is a rule of λS. That means that one of the derivation rules of λS is

$$\frac{\Gamma \vdash T : s_1 \quad \Gamma, v{:}T \vdash U : s_2}{\Gamma \vdash \Pi v{:}T.U : s_2}\ .$$

This rule says informally that the sort s_2 is closed under dependent-product terms. The corresponding "meaning" in the model of this derivation rule would be that

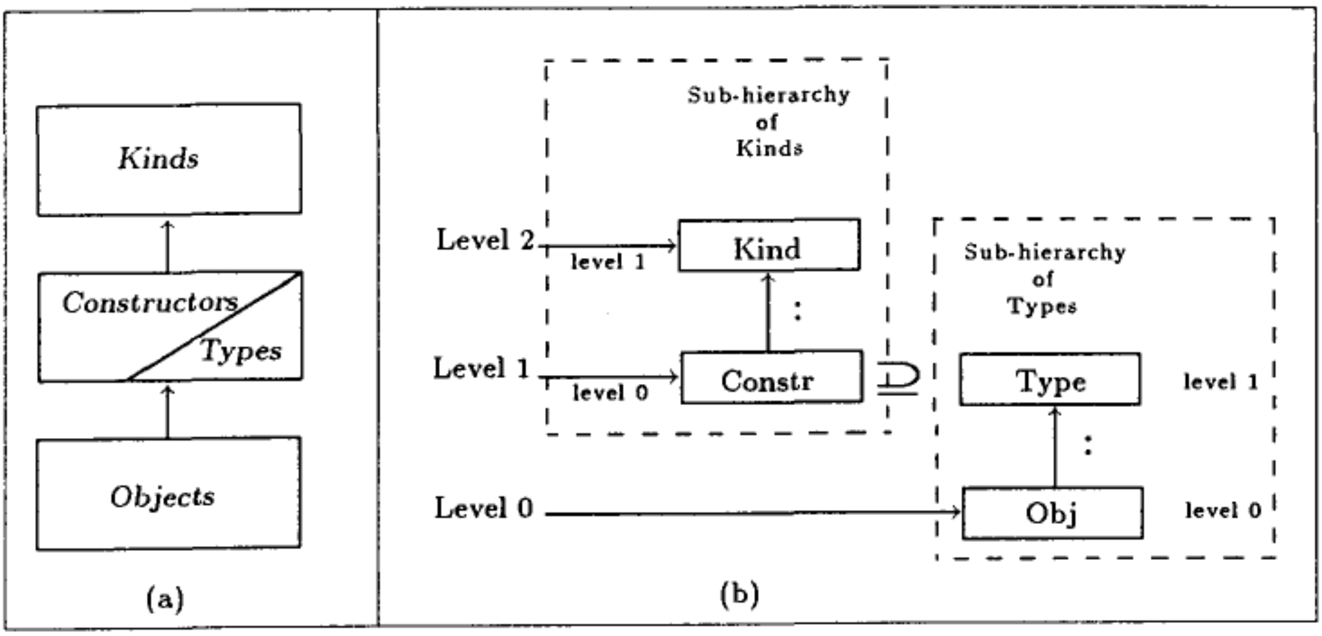

Fig. 1. The type-hierarchy

the universe $\mathbf{U}_S^{s_2}$ is closed under some suitable operation $\Pi_{s_2}^{s_1}$. More precisely, $\Pi_{s_2}^{s_1}$ takes as arguments a set $X \in \mathbf{U}_S^{s_1}$ and a family $\{Y_x\}_{x \in X}$ of sets of $\mathbf{U}_S^{s_2}$ and returns a result again in $\mathbf{U}_S^{s_2}$. In order to improve readability, we will denote the application $\Pi_{s_2}^{s_1}(X, \{Y_x\}_{x \in X})$ by

$$\Pi_{s_2}^{s_1} x \in X.Y_x.$$

Now, suppose we have found the collections $\mathbf{U}_S^*$ and $\mathbf{U}_S^\Box$ and the operations $\Pi_{s_2}^{s_1}$. The type-hierarchy of λS is mapped into a set-theoretical hierarchy (λS-hierarchy) through the interpretation functions $\llbracket\ \rrbracket^2$, $\llbracket\ \rrbracket^1$ and $\llbracket\ \rrbracket^0$ (see Fig. 2). Note that it is not allowed to construct elements of a lower level of the λS-hierarchy by means of elements on higher levels (the crossed arrows in Fig. 2). The typing relation : between legal terms is mapped into the relation $\in$ on sets, so that if X is an element on level i of the λS-hierarchy, then there is an element Y on level $i+1$ such that $X \in Y$.

The typable terms of λS are interpreted as follows:

• Every kind A, $(\Gamma \vdash A : \Box)$ is mapped by the interpretation function $\llbracket\ \rrbracket^2$ to an element of $\mathbf{U}_S^\Box$. Intuitively, dependent kinds are interpreted with the help of the operations $\Pi_\Box^*$ and $\Pi_\Box^\Box$, if respectively the rules $(*, \Box)$ and $(\Box, \Box)$ are present in the specification of λS;

• Every constructor C, $(\Gamma \vdash C : A : \Box)$ is mapped by the function $\llbracket\ \rrbracket^1$ to an element of the collection

$$\bigcup \mathbf{U}_S^\Box = \{X \mid \exists Z(X \in Z \ \wedge\ Z \in \mathbf{U}_S^\Box)\}$$

in such a way that $\llbracket C \rrbracket^1 \in \llbracket A \rrbracket^2$. In particular, every type σ, $(\Gamma \vdash \sigma : *)$ is interpreted as an element of the universe $\mathbf{U}_S^*$. Impredicative types (i.e., types formed by the rule $(\Box, *)$) are interpreted with the help of the operation $\Pi_*^\Box$. Pure product types (rule $(*, *)$) are interpreted by using Π_*^*. Constructors formed by λ-abstrac-

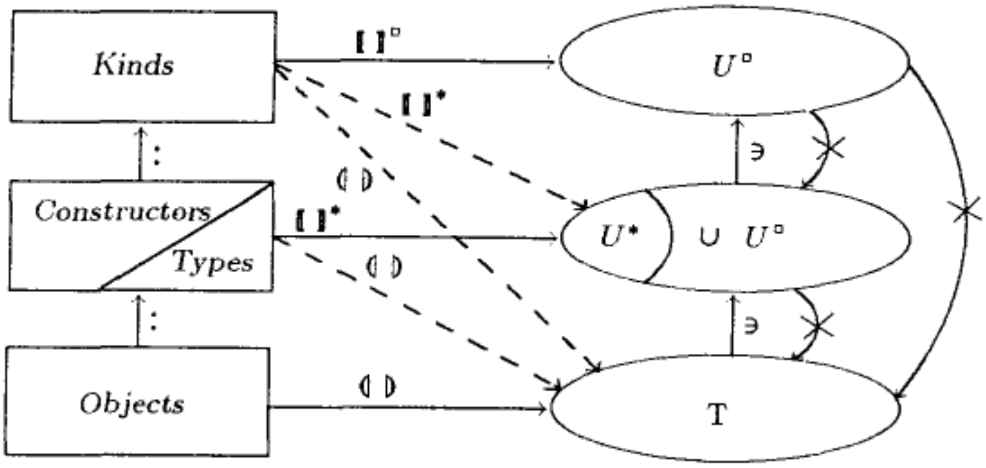

Fig. 2. λS-hierarchy

tion are interpreted as set-theoretical functions and applications of constructors to terms — as set-theoretical applications of functions to their arguments;

• Finally, every objects t, $(\Gamma \vdash t : \sigma : *)$ is mapped to a pseudo-term in a trivial way by using abstraction and application operations on pseudo-terms. For the interpretations of objects we have $\llbracket t \rrbracket^0 \in \llbracket \sigma \rrbracket^1$.

The interpretations $\llbracket\ \rrbracket^0$, $\llbracket\ \rrbracket^1$ and $\llbracket\ \rrbracket^2$ are compositional, i.e., the interpretation of a term is built up from the interpretations of its sub-terms by means of proper operations. For that reason all constructors are mapped into the set of pseudo-terms, in order to be able to interpret objects of the form $\lambda x{:}\sigma.t$ as $\lambda x{:}\llbracket \sigma \rrbracket^0.\llbracket t \rrbracket^0$. This implies that all kinds should be mapped into $\mathbf{U}^*_S$ in order to prove proper inclusion properties for the new interpretations of constructors. To summarize (see also Fig. 2):

— Every kind A is mapped into $\mathbf{U}^*_S$ by the function $\llbracket\ \rrbracket^1$ and into $\mathcal{T}$ by the function $\llbracket\ \rrbracket^0$.

— Every constructor C $(\Gamma \vdash C : A : \Box)$ is mapped into $\mathcal{T}$ by the function $\llbracket\ \rrbracket^0$ in a way that $\llbracket C \rrbracket^0 \in \llbracket A \rrbracket^1$.

Note that for the systems $\lambda\rightarrow$ and λP it is not necessary to interpret kinds as pseudo-terms, but we will do it in order to obtain more uniform treatment for all the systems of the cube. However, these two cases can be treated separately.

A final remark is that the interpretation functions $\llbracket\ \rrbracket^2$ and $\llbracket\ \rrbracket^1$ are constructed simultaneously on the structure of the typable terms. Due to this, it is possible to keep type dependencies in the interpretations.

4. THE FORMALIZATION

Any system λS of the λ-cube is interpreted into ZF-set theory. The typable terms are interpreted as sets and the typing relation ":" as the inclusion relation

$\in$ between sets. In particular, every object is mapped into a pseudo-term[1], every constructor — into a set-theoretical function, every type — into a set of pseudo-terms, and every kind — into a set of set-theoretical functions.

Note that the existence of this model does not contradict the result in [11], which simply says that in polymorphic λ-calculus one cannot interpret all abstraction-terms (i.e., terms of the form $\lambda v{:}T_1.T_2$) as set-theoretical functions and all application-terms (i.e., terms of the form T_1T_2) as function-applications. We interpret only the abstraction and application terms which are at the predicative level of λS as set-theoretical functions and function applications. The terms which are at the impredicative level are interpreted as λ-abstractions and applications of pseudo-terms.

4.1. PRELIMINARIES

As it has been mentioned before, the set $\mathcal{T}$ of pseudo-terms will be identified with the set ω. Thus an additional equality to the usual set-theoretical equality on ω will be used in order to represent β-equality. It will be denoted ambiguously by $=_\beta$.

Definition 4.1. Let a and b be sets. We say that a is *v-equal* to b (notation $a =_v b$) iff a and b are both pseudo-terms or are both sets and:

(i) $a =_\beta b$ in the case $a, b \in \mathcal{T}$;

(ii) $a = b$, otherwise.

Note that if $t_1 =_\beta t_2$ and $t_1 \in a$, it is not necessarily $t_2 \in a$. We extend the equality $=_v$ on sequences of elements of U in the following way.

Definition 4.2. Let α, γ be sequences of elements of U. Then $\alpha =_v \gamma$ iff $|\alpha| = |\gamma|$ and $\alpha(i) =_v \gamma(i)$ for all $i = 1, \ldots, |\alpha|$.

The set-theoretical functions which will be used in the model of λS form a restricted class of the functions in set theory. They are defined below.

Definition 4.3. Let a and b be sets such that $a, b \notin \mathcal{T}$. The set F is a *v-function* from a to b (notation $F : a \xrightarrow{v} b$) iff F consists of ordered pairs $\langle x, y \rangle$ such that

$$\forall x \in a \; \exists y_x \in b \; (\langle x, y_x \rangle \in F)$$

and

$$\forall x_1, x_2, y_1, y_2 \; (x_1 =_v x_2 \wedge \langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle \in F \Rightarrow y_1 =_v y_2).$$

[1] Note that the set of pseudo-terms can be identified with the set ω of standard sets representing natural numbers, so every pseudo-term t can be thought as a numeral $\underline{n}_t$ which is uniquely assigned to it. For convenience we will use in the sequel the set of pseudo-terms instead of the set ω of their images into ZF set theory.

Remarks 4.4.

• Let $F : a \xrightarrow{v} b$. The *v-function application* is defined in the usual way. That is, if $\langle x, y\rangle \in F$, then $\mathsf{App}_v(F, x) =_v y$. For simplicity the v-function application will be written as $F(x)$, since the usual function-application will not be used in the sequel. Thus, if $F(x)$ is defined, that means that F is a v-function, i.e., it respects β-equality.

• Let $F : a \xrightarrow{v} b$ and $F(x) =_v y_x$ for $x \in a$. This F will be denoted by $\lambda_v x{\in}a.y_x$.

Lemma 4.5. *The v-functions $F_i : a_i \xrightarrow{v} b_i$, $i = 1, 2$, are v-equal, e.g. $F_1 =_v F_2$ iff $a_1 =_v a_2$ and for all $x \in a_1$, $F_1(x) =_v F_2(x)$.*

Definition 4.6. Let a be a set such that $a \notin \mathcal{T}$, and let $\{b_x\}_{x\in a}$ be a family of sets such that $b_x \notin \mathcal{T}$ for all $x \in a$:

(i) The set-theoretical v-dependent product is defined as

$$\Pi_v x{\in}a.b_x := \{F : a \xrightarrow{v} \bigcup_{x\in a} b_x \mid \forall x \in a\ (F(x) \in b_x)\}.$$

(ii) The dependent sum is defined as

$$\Sigma_v x{\in}a.b_x := \{\langle m, n\rangle \mid m \in a, n \in b_m\}.$$

Note that if $F \in \Pi_v x{\in}a.b_x$ and $x_1 =_v x_2$, then $F(x_1) =_v F(x_2)$.

Lemma 4.7. *Let $a, a' \notin \mathcal{T}$ and let $\{b_x\}_{x\in a}$, $\{b'_x\}_{x\in a'}$ be families of sets such that $b_x, b'_x \notin \mathcal{T}$. Then*

$$\Pi_v x{\in}a.b_x =_\beta \Pi_v x{\in}a'.b'_x \iff a =_\beta a' \& \forall x \in a\ (b_x =_\beta b'_x).$$

The hierarchy of sets into which the typable terms (kinds, constructors, types and objects) will be mapped is specified as follows:

Definition 4.8. For every ordinal number $\alpha \in \mathtt{Or}$ the sets $\mathtt{V}_\alpha(\mathcal{T})$ are defined in the following way:

(i) $\mathtt{V}_0(\mathcal{T}) = \mathcal{T}$;

(ii) $\mathtt{V}_{\alpha+1}(\mathcal{T}) = \mathtt{V}_\alpha(\mathcal{T}) \bigcup \mathcal{P}(\mathtt{V}_\alpha(\mathcal{T}))$;

(iii) $\mathtt{V}_\alpha(\mathcal{T}) = \bigcup_{\beta<\alpha} \mathtt{V}_\beta(\mathcal{T})$ if α is a limit ordinal.

Definition 4.9. Let $\circ$ be an operation which takes as arguments a set and a family of sets, indexed by this set, and gives as a result a set:

(i) The set $\mathcal{A} \subseteq \mathtt{V}_\alpha(\mathcal{T})$ is *α-closed* under the operation $\circ$ if for any set $a \in \mathcal{A}$ and any family $\{b_x\}_{x\in a}$ of sets from $\mathcal{A}$, the set $\circ(a, \{b_x\}_{x\in a})$ belongs to $\mathcal{A}$;

(ii) The set $\mathcal{A} \subseteq \mathtt{V}_\alpha(\mathcal{T})$ is *weakly-α-closed* under the operation $\circ$ if there exists an ordinal β such that $\beta < (\alpha - 1)$ and moreover, for all ordinals γ such that $\beta \leq \gamma < \alpha$ and for any family of sets $\{b_x\}_{x \in a}$, for which $b_x \in \mathcal{A} \cap \mathtt{V}_\gamma(\mathcal{T})$, the set $\circ(a, \{b_x\}_{x\in a})$ belongs to $\mathcal{A}$.

4.2. THE UNIVERSE $\mathbf{U}_S^*$ AND $\mathbf{U}_S^\Box$

The interpretation $\mathbf{U}_S^\Box$ of the predicative universe $\Box$ is chosen to be the set

$$\mathtt{V}_\omega(\mathcal{T}) \setminus \mathtt{V}_1(\mathcal{T}).$$

The next lemma shows that it is weakly-ω-closed under the set-theoretical dependent product Π_υ (defined in Definition 4.6).

Lemma 4.10. *The universe $\mathbf{U}_S^\Box$ is weakly-ω-closed under the operation Π_υ.*

Proof. We have to find an ordinal $\beta < \omega$ such that for all $n \geq \beta$ it holds that

$$\Pi_\upsilon x{\in}a.b_x \in \mathbf{U}_S^\Box$$

if $a \in \mathbf{U}_S^\Box$ and $\{b_x\}_{x\in a}$ is a family of sets such that $b_x \in \mathbf{U}_S^\Box \cap \mathtt{V}_n(\mathcal{T})$ for any $x \in a$. Note that

$$\mathbf{U}_S^\Box \bigcup \mathtt{V}_n(\mathcal{T}) =_\upsilon \mathtt{V}_n(\mathcal{T}) \setminus \mathtt{V}_1(\mathcal{T}).$$

Now, let us choose $\beta = 2$. Let $a \in \mathbf{U}_S^\Box$ and let $n \geq 2$. From the definition of $\mathbf{U}_S^\Box$ it follows that there exists a natural number $m \geq 2$ such that $a \in \mathtt{V}_m(\mathcal{T})$.

The elements of the set $\Pi_\upsilon x{\in}a.b_x$ are υ-functions and thus sets of pairs of the form (x, y), where $x \in a$ and $y \in b_x$. By definition, a pair (x, y) is a set $\{x, \{x, y\}\}$. Thus, if $x \in a \in \mathtt{V}_m(\mathcal{T})$ and $y \in b_x \in \mathtt{V}_n(\mathcal{T})$, then $(x, y) \in \mathtt{V}_{\max(m-1,n-1)+2}(\mathcal{T})$. Consequently,

$$\Pi_\upsilon x{\in}a.b_x \in \mathtt{V}_{\max(m-1,n-1)+4}(\mathcal{T}),$$

and hence $\Pi_\upsilon x{\in}a.b_x \in \mathbf{U}_S^\Box$ since obviously

$$\Pi_\upsilon x{\in}a.b_x \notin \mathtt{V}_1(\mathcal{T}).$$ ∎

It is convenient to specify the interpretation $\mathbf{U}_S^*$ of the impredicative universe $*$ to be the collection SAT_β of β-saturated sets. SAT_β is closed under arbitrary non-empty intersections and under an operation of dependent product defined on the set $\mathcal{T}$ of pseudo-terms.

Let $\mathsf{SN}_\beta \subset \mathcal{T}$ be the set of pseudo-terms which are strongly normalizing under β-reduction.

Definition 4.11. The set $\mathcal{B}_\beta$ of *β-base terms* is defined as the smallest set satisfying the following conditions:

(i.) $Var^* \bigcup Var^\Box \subset \mathcal{B}_\beta$;

(ii.) If $M \in \mathcal{B}_\beta$ and $N \in \mathsf{SN}_\beta$, then $MN \in \mathcal{B}_\beta$.

Definition 4.12. The *β-key-reduction* is the relation $\xrightarrow{k}_\beta$ defined by the contraction schemes for β-reduction and the following compatibility condition:

$$M_1 \xrightarrow{k}_\beta M_2 \Longrightarrow M_1 N \xrightarrow{k}_\beta M_2 N.$$

Lemma 4.13. *If the proper sub-terms of a term* M *are β-strongly normalizing,* $M \xrightarrow{k}_\beta N$ *and* $N \in \mathsf{SN}_\beta$*, then* $M \in \mathsf{SN}_\beta$.

Definition 4.14. The set $X \subset \mathcal{T}$ of pseudo-terms is called *β-saturated* if the following conditions hold:

(i.) $X \subseteq \mathsf{SN}_\beta$;

(ii.) $\mathcal{B}_\beta \subset X$;

(iii.) If $M \xrightarrow{k}_\beta N$, $N \in X$ and the proper subterms of M are β-strongly normalizing, then $M \in X$.

The collection of all β-saturated sets will be denoted by SAT_β. Thus one chooses $\mathbf{U}^* \equiv \mathsf{SAT}_\beta$.

Definition 4.15. The operation Π^*_* of dependent product on $\mathcal{T}$ takes as arguments a set $X \subseteq \mathcal{T}$ and a function $F : X \to \mathcal{P}(\mathcal{T})$ and is defined as follows:

$$\Pi^*_* m \in X.F(m) := \{t \in \mathcal{T} \mid \forall q \in X \ (tq \in F(q))\ \}.$$

The operation $\Pi^\square_*$ is defined as intersection of sets. Namely,

$$\Pi^*_* x \in X.Y_x = \bigcap_{x \in X} Y_x.$$

Note that $X \neq \emptyset$ for any $X \in \mathbf{U}^\square_S$. The next lemma shows that the universe $\mathbf{U}^*_S$, e.g. SAT_β, satisfies the necessary closureness properties.

Lemma 4.16. *The set* SAT_β *is closed under* Π^*_* *and under arbitrary non-empty intersections.*

4.3. THE INTERPRETATIONS

In this subsection the interpretations $[\![\ ]\!]^2$, $[\![\ ]\!]^1$ and $[\![\ ]\!]^0$ are defined (see Fig. 2). For that purpose we need two valuations

$$\xi : Var^\square \to \bigcup \mathbf{U}^\square_S \quad \text{and} \quad \rho : Var^* \bigcup Var^\square \to \mathcal{T}$$

to interpret all constructor variables at the middle level of the λS-hierarchy (see Fig. 2) and all constructor and object variables at the level of atoms. The interpretation $[\![\ ]\!]^0_{\xi,\rho}$ is obtained simply by applying the substitution ρ on its argument. Thus it does not depend on the assignment ξ and for this reason it will be written as $[\![\ ]\!]^0_\rho$.

The other two interpretations are constructed simultaneously by induction on the structure of typable terms of the system λS.

Definition 4.17. Let $\rho : Var^* \bigcup Var^\Box \to \mathcal{T}$ be a valuation of constructor and object variables. The *atom-interpretation*,

$$[\![\ ]\!]^0_\rho : \{\Box\} \bigcup \mathtt{Kind} \bigcup \mathtt{Constr} \bigcup \mathtt{Obj} \to \mathcal{T},$$

is defined as $[\![T]\!]^0_\rho = \rho(T)$, where $\rho(T)$ is the term obtained from T by applying the substitution ρ to T.

Definition 4.18. Let $\mathcal{R}_{\lambda S}$ be the set of PTS-rules of the system λS. Let $\rho : Var^* \bigcup Var^\Box \to \mathcal{T}$ and $\xi : Var^\Box \to \bigcup \mathbf{U}^\Box_S$ be valuations. The *constructor-interpretation* of constructor and kinds,

$$[\![\ ]\!]^1_{\xi,\rho} : \{\Box\} \bigcup \mathtt{Kind} \bigcup \mathtt{Constr} \to \bigcup \mathbf{U}^\Box_S,$$

and the *kind-interpretation* of kinds,

$$[\![\ ]\!]^2_{\xi,\rho} : \{\Box\} \bigcup \mathtt{Kind} \to \mathbf{U}^\Box_S,$$

are defined simultaneously by induction on the structure of the typable terms as follows:

Sorts of λS:

$$[\![*]\!]^2_{\xi,\rho} = [\![\Box]\!]^2_{\xi,\rho} = \mathsf{SAT}_\beta,$$

$$[\![*]\!]^1_{\xi,\rho} = [\![\Box]\!]^1_{\xi,\rho} = \mathsf{SN}_\beta.$$

Kinds of λS:

- $(\Box, \Box) \in \mathcal{R}_{\lambda S}$ ($A, B \in \mathtt{Kind}(\lambda S)$, $\alpha \in Var^\Box$).

$$[\![\Pi\alpha{:}A.B]\!]^2_{\xi,\rho} \simeq \Pi_v a{\in}[\![A]\!]^2_{\xi,\rho}.\Pi_v m{\in}[\![A]\!]^1_{\xi,\rho}.[\![B]\!]^2_{\xi[\alpha:=a],\rho[\alpha:=m]},$$

$$[\![\Pi\alpha{:}A.B]\!]^1_{\xi,\rho} \simeq \bigcap_{a\in[\![A]\!]^2_{\xi,\rho}} \Pi^*_* m \in [\![A]\!]^1_{\xi,\rho}.[\![B]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m]}.$$

- $(*, \Box) \in \mathcal{R}_{\lambda S}$ ($A \in \mathtt{Kind}(\lambda S)$, $\sigma \in \mathtt{Type}(\lambda S)$, $x \in Var^*$).

$$[\![\Pi x{:}\sigma.A]\!]^2_{\xi,\rho} \simeq \Pi_v m{\in}[\![\sigma]\!]^1_{\xi,\rho}.[\![A]\!]^2_{\xi,\rho[x:=m]},$$

$$[\![\Pi x{:}\sigma.A]\!]^1_{\xi,\rho} \simeq \Pi^*_* m \in [\![\sigma]\!]^1_{\xi,\rho}.[\![A]\!]^1_{\xi,\rho[x:=m]}.$$

Constructors of λS:

- Constructor-variables ($\alpha \in Var^\Box$).

$$[\![\alpha]\!]^1_{\xi,\rho} = \xi(\alpha).$$

- $(\square,\square) \in \mathcal{R}_{\lambda S}$ ($P, Q \in \texttt{Constr}(\lambda S)$, $A \in \texttt{Kind}(\lambda S)$, $\alpha \in Var^{\square}$).

$$\begin{aligned} [\![PQ]\!]^1_{\xi,\rho} &\simeq [\![P]\!]^1_{\xi,\rho}([\![Q]\!]^1_{\xi,\rho}, [\![Q]\!]^0_{\rho}), \\ [\![\lambda\alpha{:}A.P]\!]^1_{\xi,\rho} &\simeq \lambda_v a{\in}[\![A]\!]^2_{\xi,\rho}.\lambda_v m{\in}[\![A]\!]^1_{\xi,\rho}.[\![P]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m]}. \end{aligned}$$

- $(*,\square) \in \mathcal{R}_{\lambda S}$ ($P \in \texttt{Constr}(\lambda S)$, $t \in \texttt{Obj}(\lambda S)$, $\sigma \in \texttt{Type}(\lambda S)$, $x \in Var^*$).

$$\begin{aligned} [\![Pt]\!]^1_{\xi,\rho} &\simeq [\![P]\!]^1_{\xi,\rho}([\![t]\!]^0_{\rho}), \\ [\![\lambda x{:}\sigma.P]\!]^1_{\xi,\rho} &\simeq \lambda_v m{\in}[\![\sigma]\!]^1_{\xi,\rho}.[\![P]\!]^1_{\xi,\rho[x:=m]}. \end{aligned}$$

- $(\square,*) \in \mathcal{R}_{\lambda S}$ ($A \in \texttt{Kind}(\lambda S)$, $\sigma \in \texttt{Type}(\lambda S)$, $\alpha \in Var^{\square}$).

$$[\![\Pi\alpha{:}A.\sigma]\!]^1_{\xi,\rho} \simeq \bigcap_{a\in[\![A]\!]^2_{\xi,\rho}} \Pi^*_* m \in [\![A]\!]^1_{\xi,\rho}.[\![\sigma]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m]}.$$

- $(*,*) \in \mathcal{R}_{\lambda S}$ ($\sigma, \tau \in \texttt{Type}(\lambda S)$, $x \in Var^*$).

$$[\![\Pi x{:}\sigma.\tau]\!]^1_{\xi,\rho} \simeq \Pi^*_* m \in [\![\sigma]\!]^1_{\xi,\rho}.[\![\tau]\!]^1_{\xi,\rho[x:=m]}.$$

Remark 4.19. The equality $\simeq$ is the usual Kleene equality as the interpretations $[\![\ ]\!]^2_{\xi,\rho}$ and $[\![\ ]\!]^1_{\xi,\rho}$ may not be always defined.

The next lemma says that the atom-interpretations of β-equal terms of λS are also β-equal.

Lemma 4.20. *If $t_1, t_2 \in \texttt{Term}(S)$ and $t_1 =_\beta t_2$, then $[\![t_1]\!]^0_\rho =_\beta [\![t_2]\!]^0_\rho$.*

For the interpretations $[\![\ ]\!]^k_{\xi,\rho}$, $k = 1, 2$, the substitution property, which is stated in the next lemma, holds.

Lemma 4.21 (Substitution). *If $C \in \texttt{Constr}(\lambda S)$, $t \in \texttt{Obj}(\lambda S)$, $M, M[C/\alpha]$, $M[t/x] \in \texttt{Kind}(\lambda S) \cap \texttt{Constr}(\lambda S)$, then*

$$\begin{aligned} [\![M[C/\alpha]]\!]^k_{\xi,\rho} &\simeq [\![M]\!]^k_{\xi[\alpha:=[\![C]\!]^1_{\xi,\rho}],\rho[\alpha:=[\![C]\!]^0_{\rho}]}, \\ [\![M[t/x]]\!]^k_{\xi,\rho} &\simeq [\![M]\!]^k_{\xi,\rho[x:=[\![t]\!]^0_{\rho}]} \end{aligned}$$

for $k = 1, 2$.

Lemma 4.22. *Let $M_1, M_2 \in \texttt{Constr}(\lambda S) \cup \texttt{Kind}(\lambda S)$. If $M_1 \to_\beta M_2$ and $[\![M_1]\!]^k_{\xi,\rho}$ is defined, then $[\![M_1]\!]^k_{\xi,\rho} =_v [\![M_1]\!]^k_{\xi,\rho}$ for $k = 1, 2$.*

Proof. Let $M_1 \to_\beta M_2$. The following cases are treated:

• Let $(\lambda\alpha{:}A.C)Q \to_\beta C[\alpha := Q]$ for $A \in \texttt{Kind}(\lambda S)$ and $C, Q, (\lambda\alpha{:}A.C)Q \in \texttt{Constr}(\lambda S)$. Assume that the interpretation $[\![(\lambda\alpha{:}A.C)Q]\!]^1_{\xi,\rho}$ is defined, i.e., it is equal to $[\![C]\!]^1_{\xi[\alpha:=[\![Q]\!]^1_{\xi,\rho}],\rho[\alpha:=[\![Q]\!]^0_{\rho}]}$ (see Definition 4.18). Thus, from Lemma 4.21, it follows that the constructor-interpretation of $C[\alpha := Q]$ is defined and moreover, it is equal to the constructor-interpretation of $(\lambda\alpha{:}A.C)Q$.

• Let $Ct_1 \to_\beta Ct_2$, where $C \in \texttt{Constr}(\lambda S)$ and $t_1, t_2 \in \texttt{Obj}(\lambda S)$, and let $\llbracket Ct_1 \rrbracket^1_{\xi,\rho}$ be well-defined. That means that $\llbracket C \rrbracket^1_{\xi,\rho}$ is a v-function (see Remarks 4.4) and since $\llbracket t_1 \rrbracket^0_\rho =_\beta \llbracket t_2 \rrbracket^0_\rho$, it follows that $\llbracket Ct_1 \rrbracket^1_{\xi,\rho} =_\beta \llbracket Ct_2 \rrbracket^1_{\xi,\rho}$.

The rest of the cases are trivial and their proof is similar. We have proved that if $M_1 \to_\beta M_2$ and $\llbracket M_1 \rrbracket^k_{\xi,\rho}$ is defined for $k = 1, 2$, then the interpretation $\llbracket M_2 \rrbracket^k_{\xi,\rho}$ is also defined and v-equal to $\llbracket M_1 \rrbracket^k_{\xi,\rho}$. ∎

Definition 4.23. The object interpretations ρ_1 and ρ_2 are *compatible* under the β-equality if for all $v \in Var$, $\rho_1(v) =_\beta \rho_2(v)$.

The proofs of the next two lemmas are trivial by induction on the structure of typable terms.

Lemma 4.24. *If $M \in \texttt{Term}(\lambda S)$ and ρ_1 and ρ_2 are compatible object-valuations, then*

$$\llbracket M \rrbracket^0_{\rho_1} =_\beta \llbracket M \rrbracket^0_{\rho_2}.$$

Lemma 4.25. *Let $M \in \texttt{Constr}(\lambda S) \cup \texttt{Kind}(\lambda S)$ and let ρ_1 and ρ_2 be compatible object-interpretations. If the interpretations $\llbracket M \rrbracket^k_{\xi,\rho_1}$ and $\llbracket M \rrbracket^k_{\xi,\rho_2}$ are defined ($k = 1, 2$), then*

$$\llbracket M \rrbracket^k_{\xi,\rho_1} =_v \llbracket M \rrbracket^k_{\xi,\rho_1}.$$

We have mentioned earlier that the interpretations of the typable terms should satisfy some inclusion properties (Section 3). For that purpose, we introduce the notion of satisfaction of a context Γ. In such a way we restrict the possible valuations, so that the interpretations $\llbracket\ \rrbracket^1_{\xi,\rho}$ and $\llbracket\ \rrbracket^2_{\xi,\rho}$ are defined.

Definition 4.26. The valuations

$$\xi : Var^\square \to \bigcup \mathbf{U}^\square_S \quad \text{and} \quad \rho : Var^* \bigcup Var^\square \to \mathcal{T}$$

satisfy the context Γ (notation $\xi, \rho \models \Gamma$) iff:

(i) for every constructor variable α and kind A, such that $(\alpha : A) \in \Gamma$,

$$\xi(\alpha) \in \llbracket A \rrbracket^2_{\xi,\rho} \quad \text{and} \quad \rho(\alpha) \in \llbracket A \rrbracket^1_{\xi,\rho}, \quad \text{and}$$

(ii) for every object variable x and type σ, such that $(x : \sigma) \in \Gamma$,

$$\rho(x) \in \llbracket \sigma \rrbracket^1_{\xi,\rho}.$$

Definition 4.27. The (legal) context Γ *models that the (typable) term M has a type T* (notation $\Gamma \models M : T$) iff:

(i) if $M \in \texttt{Obj}(\lambda S)$, then for all $\xi, \rho \models \Gamma$,

$$[\![M]\!]^0_\rho \in [\![T]\!]^1_{\xi,\rho};$$

(ii) if $M \in \texttt{Constr}(\lambda S)$, then for all $\xi, \rho \models \Gamma$,

$$[\![M]\!]^0_\rho \in [\![T]\!]^1_{\xi,\rho} \text{ and } [\![M]\!]^1_{\xi,\rho} \in [\![T]\!]^2_{\xi,\rho};$$

(iii) if $M \in \texttt{Kind}(\lambda S)$, then there exists a natural number $n \geq 2$, such that for all $\xi, \rho \models \Gamma$,

$$[\![M]\!]^2_{\xi,\rho} \in \mathtt{V}_n(T),\ [\![M]\!]^1_{\xi,\rho} \in \mathsf{SAT}_\beta \text{ and } [\![M]\!]^0_\rho \in \mathsf{SN}_\beta.$$

Theorem 4.28 (Soundness). *If $\Gamma \vdash M : T$, then $\Gamma \models M : T$.*

Proof. The proof of 1–3 is done by induction on derivations. The following cases are treated:

• The (λ) rule. The case when the bound variable is a constructor variable and the term formed by the λ-rule is a constructor (i.e., $(\Box, \Box) \in \mathcal{R}_{\lambda S}$) is considered. The proof for all other cases of the λ-rule is done in a similar way.

$$\frac{\Gamma, \alpha{:}A \vdash P : B \quad \Gamma \vdash \Pi\alpha{:}A.B : \Box}{\Gamma \vdash \lambda\alpha{:}A.P : \Pi\alpha{:}A.B}$$

Let $\xi, \rho \models \Gamma$. We have to prove that

$$[\![\lambda\alpha{:}A.P]\!]^1_{\xi,\rho} \in [\![\Pi\alpha{:}A.B]\!]^2_{\xi,\rho} \text{ and } [\![\lambda\alpha{:}A.P]\!]^0_\rho \in [\![\Pi\alpha{:}A.B]\!]^1_{\xi,\rho}.$$

From the induction hypothesis $[\![A]\!]^2_{\xi,\rho}$ is an element of $\mathbf{U}^\Box_S$, $[\![A]\!]^1_{\xi,\rho}$ is an element of SAT_β, and for all $a \in [\![A]\!]^2_{\xi,\rho}$ and $m \in [\![A]\!]^1_{\xi,\rho}$

$$[\![B]\!]^2_{\xi[\alpha:=a],\rho[\alpha:=m]} \in \mathbf{U}^\Box_S.$$

Furthermore, for $a \in [\![A]\!]^2_{\xi,\rho}$ and $m \in [\![A]\!]^1_{\xi,\rho}$

$$[\![P]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m]} \in [\![B]\!]^2_{\xi[\alpha:=a],\rho[\alpha:=m]}. \tag{1}$$

From Lemma 4.25 it follows

$$m_1 =_\beta m_2 \Longrightarrow [\![P]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m_1]} =_v [\![P]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m_2]}$$

for all $a \in [\![A]\!]^2_{\xi,\rho}$ and $m_1, m_2 \in [\![A]\!]^1_{\xi,\rho}$, and hence the function

$$\lambda_v a{\in}[\![A]\!]^2_{\xi,\rho}.\lambda_v m{\in}[\![A]\!]^1_{\xi,\rho}.[\![P]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m]}$$

is indeed a v-function. Thus, from (1) and by the definition of $[\![\lambda\alpha{:}A.P]\!]^1_{\xi,\rho}$ it follows

$$[\![\lambda\alpha{:}A.P]\!]^1_{\xi,\rho} \in [\![\Pi\alpha{:}A.B]\!]^2_{\xi,\rho}.$$

To prove $[\![\lambda\alpha{:}A.P]\!]^0_\rho \in [\![\Pi\alpha{:}A.B]\!]^1_{\xi,\rho}$, we have to prove that for any $a \in [\![A]\!]^2_{\xi,\rho}$ and $m \in [\![A]\!]^1_{\xi,\rho}$

$$[\![\lambda\alpha{:}A.P]\!]^0_\rho m \in [\![B]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m]}.$$

The term $[\![\lambda\alpha{:}A.P]\!]^0_\rho m$ key-reduces to the term $[\![P]\!]^0_{\rho[\alpha:=m]}$, which by the induction hypothesis belongs to the saturated set $[\![B]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m]}$ and hence the term $[\![\lambda\alpha{:}A.P]\!]^0_\rho m$ itself belongs to $[\![B]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=m]}$.

• The (*app*) rule. Again only the case when the applied terms are constructors is considered (i.e., $(\square,\square)\in\mathcal{R}_{\lambda S}$). The proof of the other cases is done in the same or even simpler way.

$$\frac{\Gamma\vdash P:\Pi\alpha{:}A.B:\square \quad \Gamma\vdash Q:A:\square}{\Gamma\vdash PQ:B[Q/\alpha]}$$

Let $\xi,\rho\models\Gamma$. From the induction hypothesis it follows

$$[\![P]\!]^1_{\xi,\rho}\in[\![\Pi\alpha{:}A.B]\!]^2_{\xi,\rho},\quad [\![Q]\!]^1_{\xi,\rho}\in[\![A]\!]^2_{\xi,\rho}\ \text{ and }\ [\![Q]\!]^0_\rho\in[\![A]\!]^1_{\xi,\rho}.$$

Thus,

$$[\![PQ]\!]^1_{\xi,\rho}\in[\![B]\!]^2_{\xi[\alpha:=[\![Q]\!]^1_{\xi,\rho}],\rho[\alpha:=[\![Q]\!]^0_\rho]}.$$

From the Substitution Lemma (see Lemma 4.21) it follows

$$[\![PQ]\!]^1_{\xi,\rho}\in[\![B[\alpha:=Q]]\!]^2_{\xi,\rho}.$$

• The $(\prod)$ rule. The case of $(\square,\square)\in\mathcal{R}_{\lambda S}$ is considered again. The proof in the case $(*,\square)$ is similar and the proofs in the cases $(\square,*)$ or $(*,*)$ follow directly form the closure properties of SAT_β (see Lemma 4.16). Let now the last rule in the derivation of $\Gamma\vdash M:T$ be

$$\frac{\Gamma\vdash A:\square \quad \Gamma,\alpha{:}A\vdash B:\square}{\Gamma\vdash\Pi\alpha{:}A.B:\square}$$

Let $\xi,\rho\models\Gamma$. From the induction hypothesis it follows that there exist $m,n\geq 2$ such that

$$[\![A]\!]^2_{\xi,\rho}\in\mathtt{V}_m(\mathcal{T}),\quad [\![A]\!]^1_{\xi,\rho}\in\ \mathsf{SAT}_\beta,\quad\text{and}\quad [\![B]\!]^2_{\xi[\alpha:=a],\rho[\alpha:=m]}\in\mathtt{V}_n(\mathcal{T})$$

for any $a\in[\![A]\!]^2_{\xi,\rho}$ and $b\in[\![A]\!]^1_{\xi,\rho}$. Thus $[\![\Pi\alpha{:}A.B]\!]^2_{\xi,\rho}$ is defined and equal to the set

$$\Pi_v a{\in}[\![A]\!]^2_{\xi,\rho}.\Pi_v b{\in}[\![A]\!]^1_{\xi,\rho}.[\![B]\!]^2_{\xi[\alpha:=a],\rho[\alpha:=b]}. \tag{2}$$

The elements of this set are v-functions which consist of triples of the form (a,b,c) with $a\in[\![A]\!]^2_{\xi,\rho}$, $b\in[\![A]\!]^1_{\xi,\rho}$ and $c\in[\![B]\!]^2_{\xi[\alpha:=a],\rho[\alpha:=b]}$. Thus for any such triple (a,b,c) it follows $a\in\mathtt{V}_{m-1}(\mathcal{T})$, $b\in\mathtt{V}_0(\mathcal{T})$ and $c\in\mathtt{V}_{n-1}(\mathcal{T})$. Note that by definition

$$(a,b,c)\equiv\{a,\{a,\{b,\{b,c\}\}\}\}.$$

Thus $(a,b,c)\in\mathtt{V}_{\max(n+1,m-1)+2}(\mathcal{T})$ and hence

$$\Pi_v a{\in}[\![A]\!]^2_{\xi,\rho}.\Pi_v b{\in}[\![A]\!]^1_{\xi,\rho}.[\![B]\!]^2_{\xi[\alpha:=a],\rho[\alpha:=b]}\in\mathtt{V}_{\max(n-1,m-1)+4}(\mathcal{T}).$$

Consequently,

$$[\![\Pi\alpha{:}A.B]\!]^2_{\xi,\rho}\in\mathtt{V}_{\max(n-1,m-1)+3}(\mathcal{T}).$$

• The (*conv*) rule.

$$\frac{\Gamma \vdash M : T \quad \Gamma \vdash U : s}{\Gamma \vdash M : U} \quad T =_\beta U.$$

Let, for example, U be a kind, i.e., $s \equiv \Box$. From the induction hypothesis it follows

$$[\![M]\!]^k_{\xi,\rho} \in [\![T]\!]^{k+1}_{\xi,\rho}$$

for $k = 0, 1$. Moreover, the interpretations $[\![T]\!]^{k+1}_{\xi,\rho}$ and $[\![U]\!]^{k+1}_{\xi,\rho}$ are defined for $k = 0, 1$. Since the property of confluence holds for β-conversion on the set of pseudo-terms, it follows that there exists a pseudo-term V such that $T \twoheadrightarrow_\beta V$ and $U \twoheadrightarrow_\beta V$. Thus, from Lemma 4.22 it follows

$$[\![T]\!]^{k+1}_{\xi,\rho} =_v [\![V]\!]^{k+1}_{\xi,\rho} \text{ and } [\![U]\!]^{k+1}_{\xi,\rho} =_v [\![V]\!]^{k+1}_{\xi,\rho}$$

for $k = 1, 2$, and hence $[\![T]\!]^{k+1}_{\xi,\rho} =_v [\![U]\!]^{k+1}_{\xi,\rho}$. Consequently,

$$[\![M]\!]^k_{\xi,\rho} \in [\![T]\!]^{k+1}_{\xi,\rho}.$$

The theorem is proved. ■

5. STRONG NORMALIZATION

The property of Strong Normalization for the system λS of the λ-cube is obtained as a corollary of the Soundness Theorem.

Theorem 5.1 (Strong Normalization for λS). *For every context Γ of λS and for every terms M and T, such that $\Gamma \vdash M : T$, it follows that $M \in \mathsf{SN}_\beta$.*

Proof. We define a maximum element for every kind-interpretation of kinds in the following way:

$$\begin{aligned}
max(\,\mathsf{SAT}_\beta) &= \mathsf{SN}_\beta,\\
max([\![\Pi\alpha{:}A.B]\!]^2_{\xi,\rho}) &= \lambda_v a{\in}[\![A]\!]^2_{\xi,\rho}.\lambda_v m{\in}[\![A]\!]^1_{\xi,\rho}.max([\![B]\!]^2_{\xi[\alpha:=a],\rho[\alpha:=m]}),\\
max([\![\Pi x{:}\sigma.B]\!]^2_{\xi,\rho}) &= \lambda_v m{\in}[\![\sigma]\!]^1_{\xi,\rho}.max([\![B]\!]^2_{\xi,\rho[x:=m]}).
\end{aligned}$$

Let $\rho(v) = v$ for every variable v, and $\xi(\alpha) = max([\![A]\!]^2_{\xi,\rho})$ for every $(\alpha{:}A) \in \Gamma$. (This is possible due to the linearity of the legal contexts.) Obviously, the so-chosen valuations satisfy Γ. From the Soundness Theorem it follows that $[\![M]\!]^0_\rho \in [\![T]\!]^1_{\xi,\rho} \subseteq \mathsf{SN}_\beta$, and hence $M \in \mathsf{SN}_\beta$. ■

6. INDUCTIVE TYPES

In the following a method for extending the present SN-proof to systems with *inductive types* is presented. For simplicity we consider only the system λC. It is the most general system of the λ-cube and all non-trivial cases are captured.

In order to use a typing system for practical applications, there should be a certain mechanism for defining data types (e.g., inductive types) in it. The study of inductive types, however, happens to be a rather difficult task as well for defining a general inductive scheme as for studying the metatheory of type systems with inductive types. A general scheme for defining inductive types is presented, for example, in [10]. Here we will use a particular example, e.g., the type Nat of *Natural Numbers*, to show the flexibility of the present SN-proof. The system obtained from λC by adding the derivation and reduction rules for Nat will be denoted by $\lambda C + \mathsf{Nat}$.

The main problems for studying the metatheory of λC with inductive types arise when there are inductive types at the impredicative level of λC (i.e., which are of type $*$) and it is possible to define a type (respectively, predicate) over the elements of some inductive type. That means that for different elements of this inductive type the elimination scheme yields different types. Thus, the elements of an inductive type are distinguishable and one can prove inequalities like $0 \neq 1$. Therefore, the well-known syntactically-oriented proofs of Strong Normalization for λC and similar systems ([5, 6]), which exploit the idea of unifying all inhabitants of a type, are not directly adaptable for systems with inductive types. In such proofs many new technical complications must be added in order to adapt them to a system with inductive types (see [14, 15, 13]). Further, it is very likely that it is not possible to extend such syntactically-oriented proofs to normalization proofs of systems with mixed inductive types and kinds. Since the dependencies between constructors and objects are not disregarded in the interpretations presented here, we do not face the above problems. The present SN-proof is extended in a straightforward way to systems with other type-constructors. In the present section such an extension is shown for the system $\lambda C + \mathsf{Nat}$.

The additional rules of $\lambda C + \mathsf{Nat}$ are listed in Table 2.

The rule (elim, $*$) is called *small elimination* and the rule (elim, $\square$) — *large elimination*.

There are two additional reduction rules for computing the values of recursive functions over Nat. This sort of reduction is called *ι-reduction*. The contraction rules for the ι-reduction are defined as follows:

$$\begin{array}{rcl} \mathtt{Rec}(P[x]; f_0, f_s[x,v])(0) & \to_\iota & f_0, \\ \mathtt{Rec}(P[x]; f_0, f_s[x,v])(sn) & \to_\iota & f_s[x := n, v := \mathtt{Rec}(P[x]; f_0, f_s[x,v])(n)]. \end{array}$$

The proof is extended as follows. First, the notion of v-equality is modified in order to comprise ι-reduction as well. This modification is obvious (see Definition 4.1). Further, the notion of saturated set should be adapted to $\beta\iota$-reduction. This is done below. Let $\mathsf{SN}_{\beta\iota} \subseteq \mathcal{T}$ be the set of pseudo-terms which are strongly normalizing under $\beta\iota$-reduction.

Table 2. Rules for the type Nat

(form)	$\vdash \mathsf{Nat} : *$	
(intro1)	$\vdash 0 : \mathsf{Nat}$	
(intro2)	$\dfrac{\Gamma \vdash n : \mathsf{Nat}}{\Gamma \vdash \mathbf{s}(n) : \mathsf{Nat}}$	
(elim, s)	$\dfrac{\Gamma \vdash n : \mathsf{Nat} \quad \Gamma, x{:}\mathsf{Nat} \vdash P : s \quad \Gamma \vdash f_0 : P[x := 0] \quad \Gamma, x{:}\mathsf{Nat}, v{:}P \vdash f_s : P[x := \mathbf{s}(x)]}{\Gamma \vdash \mathtt{Rec}(P[x]; f_0, f_s[x,v])(n) : P[x := n]}$	$s \in \{*, \square\}$

Definition 6.1. The set of *$\beta\iota$-base terms* $\mathcal{B}_{\beta\iota}$ is defined as the smallest set satisfying the following conditions:

(i) $Var \subseteq \mathcal{B}_{\beta\iota}$;

(ii) if $M \in \mathcal{B}_{\beta\iota}$ and $N \in \mathsf{SN}_{\beta\iota}$, then $MN \in \mathcal{B}_{\beta\iota}$;

(iii) if $M \in \mathcal{B}_{\beta\iota}$ and $P, f_0 f_s, t \in \mathsf{SN}_{\beta\iota}$, then $\mathtt{Rec}(P[x]; f_0, f_s[x,v])(M) \in \mathcal{B}_{\beta\iota}$.

Definition 6.2. One step *$\beta\iota$-key-reduction* is defined by the contraction scheme

$$\begin{array}{rcl} (\lambda v{:}T_1.T_2)T_3 & \xrightarrow{k}_{\beta\iota} & T_2[v := T_3], \\ \mathtt{Rec}(P[x]; f_0, f_s[x,v])(0) & \xrightarrow{k}_{\beta\iota} & f_0, \\ \mathtt{Rec}(P[x]; f_0, f_s[x,v])(sn) & \xrightarrow{k}_{\beta\iota} & f_s[x := n, v := \mathtt{Rec}(P[x]; f_0, f_s[x,v)(n)] \end{array}$$

and by the compatibility extensions

$$\begin{array}{lcl} T_1 \xrightarrow{k}_{\beta\iota} T_2 & \Longrightarrow & T_1 M \xrightarrow{k}_{\beta\iota} T_2 M, \\ T_1 \xrightarrow{k}_{\beta\iota} T_2 & \Longrightarrow & \mathtt{Rec}(P[x]; f_0, f_s[x,v])(T_1) \xrightarrow{k}_{\beta\iota} \mathtt{Rec}(P[x]; f_0, f_s[x,v])(T_2). \end{array}$$

Fact 6.3. If the proper sub-terms of a term T_1 are $\beta\iota$-strongly-normalizing, $T_1 \xrightarrow{k}_{\beta\iota} T_2$ and $T_2 \in \mathsf{SN}_{\beta\iota}$, then $T_1 \in \mathsf{SN}_{\beta\iota}$.

The next definition describes the collection $\mathsf{SAT}_{\beta\iota}$ of $\beta\iota$-saturated sets. Note that only the subscripts differ from those in Definition 4.14.

Definition 6.4. The set X of pseudo-terms is *$\beta\iota$-saturated* if:

(i) $X \subseteq \mathsf{SN}_{\beta\iota}$;

(ii) $\mathcal{B}_{\beta\iota} \subseteq X$;

(iii) if $T_1 \xrightarrow{k}_{\beta\iota} T_2$, $T_2 \in X$ and the proper sub-terms of T_1 are $\beta\iota$-strongly-normalizing, then $T_1 \in X$.

It is easy to check that the closureness properties listed in Lemma 4.16 are also valid for the collection $\mathsf{SAT}_{\beta\iota}$.

The interpretations which are needed to be added to those in Definitions 4.18 and 4.17 are specified as follows:

Atom-interpretations:

$$\llbracket \mathsf{Nat} \rrbracket^0_\rho := \mathsf{Nat},$$
$$\llbracket \mathtt{s}(n) \rrbracket^0_\rho := \mathtt{s}(\llbracket n \rrbracket^0_\rho),$$
$$\llbracket \mathtt{Rec}(P[x]; f_0, f_s[x,v])(n) \rrbracket^0_\rho := \mathtt{Rec}(\llbracket P \rrbracket^0_{\rho[x:=x]}[x]; \llbracket f_0 \rrbracket^0_\rho, \llbracket f_s \rrbracket^0_{\rho[x:=x,v:=v]}[x,v])(\llbracket n \rrbracket^0_\rho).$$

Constructor-interpretations:

• The constructor-interpretation of the type Nat is defined to be the smallest $\beta\iota$-saturated set which contains 0 and is closed under s, that is

$$\llbracket \mathsf{Nat} \rrbracket^1_{\xi,\rho} := \mu X \in \mathsf{SAT}_{\beta\iota}(0 \in X \& (x \in X \Longrightarrow \mathtt{s}(x) \in X)). \tag{3}$$

• The constructor-interpretation of terms of the form

$$\mathtt{Rec}(A[x]; f_0, f_s[x,\alpha])(n),$$

obtained by applying the rule for large elimination over Nat, is defined below. Its definition uses recursion over the set $\llbracket \mathsf{Nat} \rrbracket^1_{\xi,\rho}$. First some auxiliary functions are defined. Let in the following

$$G_0 \equiv \llbracket f_0 \rrbracket^1_{\xi,\rho}. \tag{4}$$

Let also $g(n)$ and $G_s(n,a)$ be (dependent) set-theoretical functions defined by the equations ($n \in \llbracket \mathsf{Nat} \rrbracket^1_{\xi,\rho}$, $a \in \llbracket A \rrbracket^2_{\xi,\rho[x:=n]}$ and $z \notin FV(A, f_0, f_s, x, \alpha)$)

$$g(n) := \llbracket \mathtt{Rec}(A[x]; f_0, f_s[x,\alpha])(z) \rrbracket^0_{\rho[z:=n]}, \tag{5}$$

$$G_2(n,a) = \llbracket f_s \rrbracket^1_{\xi[\alpha:=a],\rho[\alpha:=g(n),x:=n]}. \tag{6}$$

Finally, let $G(n)$ be a set-theoretical function with domain $\llbracket \mathsf{Nat} \rrbracket^1_{\xi,\rho}$, defined by

$$G(n) = max(\llbracket A \rrbracket^2_{\xi,\rho[x:=n]}). \tag{7}$$

Now we define a function $F(n)$ by recursion on $n \in \llbracket \mathsf{Nat} \rrbracket^1_{\xi,\rho}$:

$$\begin{array}{lcll} F(0) & = & G_0, & \\ F(\mathtt{s}(n)) & = & G_s(n, F(n)), & \\ F(b) & = & G(b) & \text{if } b \in \mathcal{B}_{\beta\iota}, \\ F(M) & = & F(M') & \text{if } M \xrightarrow{k}_{\beta\iota} M'. \end{array} \tag{8}$$

Then we can define

$$[\![\mathbf{Rec}(P[x]; f_0, f_s[x,v])(n)]\!]^1_{\xi,\rho} \simeq F([\![n]\!]^0_\rho). \tag{9}$$

The Lemmas 4.20–4.22, 4.24, 4.25 and their proof are extended in an obvious way. The proof of the Soundness Theorem 4.28 is also extended in a trivial way for the new rules. The case when the last rule in the derivation of a judgment $\Gamma \vdash M : T$ is the rule for large elimination over Nat, will be treated here:

$$\frac{\begin{array}{c}\Gamma \vdash t : \mathsf{Nat} \quad \Gamma, x{:}\mathsf{Nat} \vdash A : \square \\ \Gamma \vdash f_0 : A[x := 0] \quad \Gamma, x{:}\mathsf{Nat}, v{:}A \vdash f_s : A[x := \mathbf{s}(x)]\end{array}}{\Gamma \vdash \mathbf{Rec}(A[x]; f_0, f_s[x,\alpha])(t) : A[x := t]}$$

We have to prove that if $\xi, \rho \models \Gamma$, then

$$[\![\mathbf{Rec}(A[x]; f_0, f_s[x,v])(t)]\!]^1_{\xi,\rho} \in [\![A[x := t]]\!]^2_{\xi,\rho}$$

and

$$[\![\mathbf{Rec}(A[x]; f_0, f_s[x,\alpha])(t)]\!]^0_\rho \in [\![A[x := t]]\!]^1_{\xi,\rho}.$$

From the induction hypothesis the following inclusions follow:

(i) $[\![t]\!]^0_\rho \in [\![\mathsf{Nat}]\!]^1_{\xi,\rho}$;

(ii) $\forall n \in [\![\mathsf{Nat}]\!]^1_{\xi,\rho}$:

$$\begin{array}{lcl} [\![A]\!]^2_{\xi,\rho[x:=n]} & \in & \mathbf{U}^\square, \\ [\![A]\!]^1_{\xi,\rho[x:=n]} & \in & \mathsf{SAT}_{\beta\iota}, \\ [\![A]\!]^0_{\rho[x:=n]} & \in & \mathsf{SN}_{\beta\iota}; \end{array}$$

(iii) $$\begin{array}{lcl} [\![f_0]\!]^1_{\xi,\rho} & \in & [\![A]\!]^2_{\xi,\rho[x:=0]}, \\ [\![f_0]\!]^0_\rho & \in & [\![A]\!]^1_{\xi,\rho[x:=0]}; \end{array}$$

(iv) $\forall n \in [\![\mathsf{Nat}]\!]^1_{\xi,\rho}$, $\forall a \in [\![A]\!]^2_{\xi,\rho[x:=n]}$, $\forall l \in [\![A]\!]^1_{\xi,\rho[x:=n]}$:

$$\begin{array}{lcl} [\![f_s]\!]^1_{\xi[\alpha:=a],\rho[\alpha:=l,x:=n]} & \in & [\![A]\!]^2_{\xi,\rho[x:=\mathbf{s}(n)]}, \\ [\![f_s]\!]^0_{\rho[\alpha:=l,x:=n]} & \in & [\![A]\!]^1_{\xi,\rho[x:=\mathbf{s}(n)]}. \end{array}$$

Note that G_0 and the function $G(n)$ are well-defined (see (4), (iii), (7) and (ii)). We will prove also that the function $G_s(n,a)$ is well-defined. First, we shall prove that $g(n) \in [\![A]\!]^1_{\xi,\rho[x:=n]}$ (see (5)) by induction on $n \in [\![\mathsf{Nat}]\!]^1_{\xi,\rho}$:

1. Let $n \equiv 0$.

$$g(0) = \mathbf{Rec}([\![A]\!]^0_{\rho[x:=x]}[x]; [\![f_0]\!]^0_\rho, [\![f_s]\!]^0_{\rho[x:=x,\alpha:=\alpha]})(0),$$

and hence $g(0) \xrightarrow{k}_{\beta\iota} [\![f_0]\!]^0_\rho$. Thus, from (iii), (ii) and Definition 6.4 it follows

$$g(0) \in [\![A]\!]^1_{\rho[x:=0]}.$$

2. Let $n \equiv \mathbf{s}(m)$ for some $m \in [\![\mathsf{Nat}]\!]^1_{\xi,\rho}$ and let us assume $g(m) \in [\![A]\!]^1_{\xi,\rho[x:=m]}$.

$$g(\mathbf{s}(m)) = \mathtt{Rec}([\![A]\!]^0_{\rho[x:=x]}[x]; [\![f_0]\!]^0_\rho, [\![f_s]\!]^0_{\rho[x:=x,\alpha:=\alpha]})(\mathbf{s}([\![m]\!]^0_\rho)).$$

Thus $g(\mathbf{s}(m)) \xrightarrow{k}_{\beta\iota} [\![f_s]\!]^0_{\rho[\alpha:=g(m),x:=m]}$ and hence from (iv), (ii) and Definition 6.4 we obtain

$$g(\mathbf{s}(m)) \in [\![A]\!]^1_{\xi,\rho[x:=\mathbf{s}(m)]}.$$

3. Let $n \in \mathcal{B}_{\beta\iota}$. Then also $g(n) \in \mathcal{B}_{\beta\iota}$ (see Definition 6.1 and (5)) and hence $g(n) \in [\![A]\!]^1_{\xi,\rho[x:=n]}$ since $[\![A]\!]^1_{\xi,\rho[x:=n]}$ is a $\beta\iota$-saturated set.

4. Finally, let $n \xrightarrow{k}_{\beta\iota} n'$ and let us assume $g(n') \in [\![A]\!]^1_{\xi,\rho[x:=n']}$. Note that $g(n) \xrightarrow{k}_{\beta\iota} g(n')$ (see Definition 6.2) and hence $g(n) \in [\![A]\!]^1_{\xi,\rho[x:=n']}$ since $[\![A]\!]^1_{\xi,\rho[x:=n']}$ is a saturated set. Further, from Lemma 4.25 it follows

$$[\![A]\!]^1_{\xi,\rho[x:=n']} = [\![A]\!]^1_{\xi,\rho[x:=n]},$$

and hence $g(n) \in [\![A]\!]^1_{\xi,\rho[x:=n]}$.

Thus we have proved that $g(n) \in [\![A]\!]^1_{\xi,\rho[x:=n]}$. This implies that, first, the function $G_s(n,a)$ (see (6)) is well-defined, and second,

$$[\![\mathtt{Rec}(A[x]; f_0, f_s[x,\alpha])(t)]\!]^0_\rho \in [\![A[x := t]]\!]^1_{\xi,\rho} \tag{10}$$

since $[\![\mathtt{Rec}(A[x]; f_0, f_s[x,\alpha])(t)]\!]^0_\rho = g([\![t]\!]^0_\rho)$.

It follows now that the function $F(n)$ (see (8)) is well-defined, because G_0 and the functions G and G_s are well-defined. Now we shall prove that $F(n) \in [\![A]\!]^2_{\xi,\rho[x:=n]}$ by induction on $n \in [\![\mathsf{Nat}]\!]^1_{\xi,\rho}$:

1. Let $n \equiv 0$. By definition

$$F(0) = [\![f_0]\!]^1_{\xi,\rho}.$$

Thus, from (iii) it follows $F(0) \in [\![A]\!]^2_{\xi,\rho[x:=0]}$.

2. Let $n \equiv \mathbf{s}(m)$ for some $m \in [\![\mathsf{Nat}]\!]^1_{\xi,\rho}$ and let us assume

$$F(m) \in [\![A]\!]^2_{\xi,\rho[x:=m]}.$$

By definition (see (8)) $F(\mathbf{s}(m)) = [\![f_s]\!]^1_{\xi[\alpha:=F(m)],\rho[\alpha:=g(m),x:=m]}$. Thus, from (iv) it follows $F(\mathbf{s}(m)) \in [\![A]\!]^2_{\xi,\rho[x:=\mathbf{s}(m)]}$.

3. Let $n \in \mathcal{B}_{\beta\iota}$. In this case $F(n) = max([\![A]\!]^2_{\xi,\rho[x:=n]})$ and hence $F(n) \in [\![A]\!]^2_{\xi,\rho[x:=n]}$.

4. Let now $n \xrightarrow{k}_{\beta\iota} n'$ and let $F(n') \in [\![A]\!]^2_{\xi,\rho[x:=n']}$. By definition $F(n) = F(n')$ and thus from Lemma 4.25 it follows

$$F(n) \in [\![A]\!]^2_{\xi,\rho[x:=n]}.$$

Thus we have proved that $F(n) \in \llbracket A \rrbracket^2_{\xi,\rho[x:=n]}$ and hence

$$\llbracket \mathsf{Rec}(A[x]; f_0, f_s[x,v])(t) \rrbracket^1_{\xi,\rho} \in \llbracket A[x := t] \rrbracket^2_{\xi,\rho}. \tag{11}$$

The proof of the Soundness Theorem in the case of the rule for large elimination over Nat follows directly from (10) and (11).

The proof of Strong Normalization for $\lambda C + \mathsf{Nat}$ follows in a trivial way from the Soundness Theorem (see Theorem 5.1).

7. DISCUSSION

We have presented a simple semantical proof of Strong Normalization for the systems of the λ-cube. We have shown that the property of Strong Normalization can be derived directly from a simple denotational semantics of the system considered. Further, the flexibility of this semantical proof has been illustrated by extending the system $\lambda C + \mathsf{Nat}$.

We have not addressed the following questions, which deserve some attention:

• *Generalized Inductive Definitions.* The proof presented here is extendible in a straightforward way to a proof of Strong Normalization of systems with generalized inductive definitions. Such definitions are a convenient tool for defining various inductive types, such as lists of a type σ, sigma types, finite sets, etc. A proof of Strong Normalization of λC enriched with generalized inductive definitions is presented in [12].

• *Inductive kinds.* In some systems there is a clear distinction between the level of formulas ($*$) and the level of domains ($\Box$). In such systems one prefers to define data-types rather as kinds than as types. An interesting issue, which seems to have not been considered yet in the literature, is the metatheory of a system in which inductive definitions are allowed at the both levels $*$ and $\Box$. In such systems one can define inductive predicates (for example $=_A : A \to A \to *$) at the level of formulas and inductive data types (for example $Nat : \Box$) at the level of domains. The proof described here is adapted to systems with inductive kinds in [12].

• *Generic strong normalization argument.* The proof presented above suggests a generic method for proving Strong Normalization for PTSs. The genericity lies in the fact that the properties of interpretations $\mathbf{U}^*$ and $\mathbf{U}^\Box$ of the universes $*$ and $\Box$ are derived directly from the PTS-presentation of the systems in the λ-cube. For example, the axiom $* : \Box$ is interpreted by $\mathbf{U}^* \in \mathbf{U}^\Box$, and the PTS-rules — by requiring adequate closure properties on $\mathbf{U}^*$ and $\mathbf{U}^\Box$.

We outline how one can generalize the method to a subclass PTSs.

A PTS $\mathcal{S} = \langle S, \mathcal{A}, \mathcal{R} \rangle$ is specified by three sets: S of sorts, $\mathcal{A}$ of axioms, and $\mathcal{R}$ of rules (see [2] or [4] for a detailed presentation of PTS-s). The set S of sorts is simply a set of fixed constants s_i. Every axiom has the form $s_i : s_j$, and every rule — (s_i, s_j, s_k). The PTS-rules say what kind of dependent products can be constructed inside the system $\mathcal{S}$. For example, if $(s_i, s_j, s_k) \in \mathcal{R}$, then the following ($\prod$)-rule is allowed in the system $\mathcal{S}$:

$$\frac{A \vdash s_i : \quad v{:}A \vdash B : s_j}{\Pi v{:}A.B \vdash s_k :}.$$

A relation $<$ is defined on the sets of sorts to be the smallest relation satisfying the following conditions:

(i) $s_i : s_j \Longrightarrow s_i < s_j$;

(ii) $s_i < s_j$ and $s_j < s_k \Longrightarrow s_i < s_k$.

Below we sketch out the properties each interpretation $\mathbf{U}^s$ of a sort s should possess:

— for any axiom $(s_i : s_j) \in \mathcal{A}$ it holds that $\mathbf{U}^{s_i} \in \mathbf{U}^{s_j}$;

— for any rule $(s_i, s_j, s_k) \in \mathcal{R}$, such that $s_i \leq s_k$, one can define an operation $\Pi^{s_i}_{s_j}$ for which holds

$$\forall X \in \mathbf{U}^{s_i}\ \forall \{Y_x\}_{x \in X} \in \mathbf{U}^{s_j}\ \ \Pi^{s_i}_{s_j} x \in X.Y_x\ \in \mathbf{U}^{s_k};$$

— for any rule $(s_i, s_j, s_k) \in \mathcal{R}$, such that $s_i > s_k$, it follows that $\mathbf{U}^{s_k}$ is closed under arbitrary non-empty intersections;

— for each sort s, $\emptyset \notin \mathbf{U}^s$.

It is interesting to see for which PTSs the universes U_s exist. For example, it is clear that for PTSs, for which the relation $<$ is not a strict order, i.e. $s < s$ for some sort s, such universes can not be found. Further, one needs to study more precisely the dependencies in the PTS considered, in order to specify as precise as possible the operations $\Pi^{s_i}_{s_j}$.

• *Models.* The Strong Normalization proof presented here is based on specific models of the systems of the λ-cube. In [12] an abstract notion of a model of λC will be presented. This abstract model construction generalizes the ideas presented here.

REFERENCES

1. Altenkirch, T. Constructions, Inductive Types and Strong Normalization. Ph.D. thesis, Laboratory for the Foundations of Computer Science, University of Edinburgh, 1993.
2. Barendregt, H. Typed lambda calculi. In: *Handbook of Logic in Computer Science,* Abramski (Ed.), Oxford University Press, 1992.

3. Berardi, S. Type dependence and constructive mathematics. Ph.D. thesis, Dipartamento Matematica, Universita di Torino, 1990.
4. Geuvers, H. Logics and Type Systems. Ph.D. thesis, Katholieke Universiteit Nijmegen, The Netherlands, 1993.
5. Geuvers, H., M. Nederhof. Modular proof of strong normalization for the calculus of construction. *J. of Functional Programming*, **1**(2), 1992, 1–155.
6. Geuvers, H. A short and flexible proof of strong normalization for the calculus of constructions. Tech. rep., Dept. of Computer Science, University of Eindhoven, 1995.
7. Goguen, H. A typed Operational Semantics for UTT. Ph.D. thesis, Laboratory for the Foundations of Computer Science, University of Edinburgh, 1994.
8. Hyland, J., C.-H. Ong. Modified realizability and strong normalization proofs. In: *Typed Lambda Calculi and Applications*, M. Bezem and J. Groote (Eds.), 1993.
9. Ong, C.-H., E. Ritter. A generic strong normalization argument: Application to the calculus of constructions. Tech. rep., University of Cambridge, Computer Laboratory, 1994.
10. Paulin-Mohring, C. Inductive definitions in the system coq, rules and properties. In: *TLCL 1993*, vol. 664 of Lecture Notes in Computer Science, 1993.
11. Reynolds, J. Polymorphism is not set-theoretic. In: *Lecture Notes in Computer Science*, **173**, G. Kahn, D. McQueen, G. Plotkin (Eds.), 1984.
12. Stefanova, M. Properties of Type Systems. Ph.D. thesis, University of Nijmegen, The Netherlands, 1998.
13. Stefanova, M. A proof of strong normalization for the calculus of constructions with natural numbers. Tech. rep., University of Nijmegen, The Netherlands, 1994.
14. Werner, B. A normalization proof for an impredicative type system with large elimination over integers. In: *Workshop on Types for Proofs and Programs*, B. Nordströrm, K. Petersson, G. Plotkin (Eds.), Bastad, 1992.
15. Werner, B. Une Thèorie des Constructions Inductives. Ph.D. thesis, A l'Universitè Paris 7, May 1994.
16. Werner, B. A generic normalization proof for pure type systems. Talk given at the workshop TYPES'96, 1996.

Received on September 12, 1996
Revised on July 15, 1998

Faculty of Mathematics and Informatics
University of Nijmegen, The Netherlands
E-mail: `milena@cs.kun.nl`