A Simple Model Construction for the Calculus of Constructions

M. Stefanova[1] and H. Geuvers[2⋆]

[1] Faculty of Mathematics and Informatics
University of Nijmegen, The Netherlands
e-mail: milena@cs.kun.nl
[2] Faculty of Mathematics and Informatics
University of Eindhoven, The Netherlands
e-mail: herman@win.tue.nl

Abstract. *We present a model construction for the Calculus of Constructions (CC) where all dependencies are carried out in a set-theoretical setting. The Soundness Theorem is proved and as a consequence of it Strong Normalization for CC is obtained. Some other applications of our model constructions are: showing that CC + Classical logic is consistent (by constructing a model for it) and showing that the Axiom of Choice is not derivable in CC (by constructing a model in which the type that represents the Axiom of Choice is empty).*

1 Introduction

In the literature there are many investigations on the semantics of polymorphic λ-calculus with dependent types (see for example [12, 11, 10, 1, 5, 13]). Most of the existing models present a semantics for systems in which the inhabitants of the impredicative universe (*types*) are "lifted" to inhabitants of the predicative universe (*kinds*) (see [16]). Such systems are convenient to be modeled by locally Cartesian-closed categories having small Cartesian-closed subcategories. A well-known instance of these categorical models is the category of ω-sets (or D-sets) and its subcategory of modest sets, which is isomorphic to the category of partial equivalence relations (PER). Then the types are interpreted as PERs and then "lifted" through an isomorphism to modest sets and hence to ω-sets.

In practical applications, however, one prefers to use a different simple syntactical presentation of type systems - the so-called *Pure Type Systems* (PTSs). A semantics of such a system is usually obtained by implicitly or explicitly encoding the system into the system with "lifted" types, so the types are interpreted in the same way. The resulting semantics, even the one presented by concrete models (see [12, 13]) is still complicated as it gives an indirect meaning of PTSs. Moreover, most concrete models of such type systems are extensional in the sense that the interpretation of a type is a set with an equivalence relation on it with the equivalence relation on the function space defined as the extensional equality

⋆ Part of this research was performed while the author was working at the University of Nijmegen, on the ESPRIT BRA project 'Types for Proofs and Programs'

of functions. As the syntax is not extensional, these models are less suitable for showing non-provability of various statements in PTSs.

This paper presents a new class of concrete models for the *Calculus of Constructions* (CC) presented as a PTS. The models are intensional - semantical objects are equal iff they are equal in the underlying weakly-extensional combinatory algebra. (So, two functions of the same type that have the same graph are not necessarily equal).

Furthermore, a new direct meaning is assigned to the typable expressions of CC, without "lifting" the interpretations of types to interpretations in the predicative universe. There are three disjoint collections of semantical objects in each model: *elements* (of the underlying combinatory algebra) to interpret *objects* (inhabitants of types), *poly-functionals* to interpret *constructors* (inhabitants of kinds) and *predicative sets* to interpret kinds. A special case of poly-functionals are specific sets, called *polysets*. Types are interpreted as polysets. This corresponds to the fact that types form a subclass of the collection of constructors. The poly-functionals are restricted set-theoretical functionals or sets, and the predicative sets are sets having poly-functionals as their elements. The restrictions on poly-functionals are a consequence from the fact that polymorphism is not set-theoretical in the classical sense (see [14]). However, two poly-functionals or two predicative sets are equal if they are set-theoretically equal. Two elements are equal if they are equal via the equality of the underlying weakly-extensional combinatory algebra.

The three collections of semantical objects are built simultaneously, by induction on the structure of typable terms. This is in line with the fact that objects and types cannot be defined separately for systems with dependent types. In such a way a proper direct meaning is obtained for dependent types without disregarding any dependencies.

Impredicativity is modeled in a proper way as well, by using the notion of polystructure over the underlying combinatory algebra. Polystructures poses similar closure properties as PERs, namely closed under products defined on them and intersections, but are simpler - they are just collections of subsets of the combinatory algebra.

An interesting aspect of the models that we obtain is that it is now relatively easy to find counter-models (for proving properties about the syntax). In a separate section we give some applications of this. For example we show that the Axiom of Choice (AC) is not derivable in CC by constructing a model where the type representing AC is interpreted as the empty set. Furthermore, we show how the property of *strong normalization* can be derived directly from a particular model of CC.

2 Some Basic Definitions

2.1 Calculus of Constructions

In this section a precise definition of the Calculus of Constructions (CC) is presented. We adopt the same syntax for CC as in [8, 3]. To present the derivation

rules for CC we first fix the set of *pseudoterms* from which the derivation rules select the (typable) terms.

Definition 2.1 The set of pseudoterms, $\mathcal{T}$, is defined by

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

where $\mathbf{Var}^*$ and $\mathbf{Var}^\Box$ are countable disjoint sets of variables and $\mathbf{Var} = \mathbf{Var}^* \bigcup \mathbf{Var}^\Box$.

Definition 2.2 The Calculus of Constructions is a typed λ-calculus with the following derivation rules:

(axiom)	$\vdash * : \Box$	
(Var)	$\dfrac{\Gamma \vdash T : s}{\Gamma, v{:}T \vdash v : T}$	$s \in \{*, \Box\}$, $v \in \mathbf{Var}^{\mathbf{s}} \setminus FV(\Gamma)$
(weak)	$\dfrac{\Gamma \vdash T : s \quad \Gamma \vdash M : U}{\Gamma, v{:}T \vdash M : U}$	$s \in \{*, \Box\}$, $v \in \mathbf{Var}^{\mathbf{s}} \setminus FV(\Gamma)$
($\prod$)	$\dfrac{\Gamma \vdash T : s_1 \quad \Gamma, v{:}T \vdash U : s_2}{\Gamma \vdash \Pi v : T.U : s_2}$	$s_1, s_2 \in \{*, \Box\}$
(λ)	$\dfrac{\Gamma, v{:}T \vdash M : U \quad \Gamma \vdash \Pi v : T.U : s}{\Gamma \vdash \lambda v : T.M : \Pi v : T.U}$	$s \in \{*, \Box\}$
(app)	$\dfrac{\Gamma \vdash M : \Pi v : T.U \quad \Gamma \vdash N : T}{\Gamma \vdash MN : U[N/v]}$	
(conv)	$\dfrac{\Gamma \vdash M : T \quad \Gamma \vdash U : s}{\Gamma \vdash M : U} T =_\beta U$	$s \in \{*, \Box\}$

For the informal explanation of these rules see, for example, [8, 3]. The set of terms of CC is defined by

$$\mathbf{Term} = \{A \mid \exists \Gamma, B[\Gamma \vdash A : B \vee \Gamma \vdash B : A]\}.$$

It is convenient to divide the typable terms into subsets ([3, 8]) in the following way:

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

Here, $\Gamma \vdash t : \sigma : *$ abbreviates $\Gamma \vdash t : \sigma \wedge \Gamma \vdash \sigma : *$ and $\Gamma \vdash C : A : \Box$ abbreviates $\Gamma \vdash C : A \wedge \Gamma \vdash A : \Box$.

We use x, y and z to denote variables of $\mathbf{Var}^*$, also called *object variables*, and we use α, β, and γ to denote variables of $\mathbf{Var}^\Box$, also called *constructor variables*. The small Greek letters will denote types, the letters A, B, C, P, Q - kinds or constructors and the letters t, m, n - objects.

2.2 Combinatory Algebras

Combinatory algebras are used to model the set of pseudoterms of CC. Below we list the definitions of some notions used in the present paper. Most of the definitions in this section are taken from [2] and [6].

Definition 2.3 A *combinatory algebra* (ca) is an applicative structure $\mathcal{A} = \langle \mathbf{A} , . , \mathbf{k} , \mathbf{s} , =_A \rangle$ with distinguished elements $\mathbf{k}$ and $\mathbf{s}$ satisfying

$$(\mathbf{k}.x).y =_A x , \qquad ((\mathbf{s}.x).y).z =_A (x.z).(y.z)$$

The application (.) is usually not written.

Definition 2.4 The set of *terms over* $\mathcal{A}$ (notation $\mathcal{T}(\mathcal{A})$) is defined as follows.

$$\mathcal{T} ::= \mathbf{Var} \mid \mathbf{A} \mid \mathcal{T}\mathcal{T}$$

Every ca is *combinatory complete*, i.e., for every $T \in \mathcal{T}(\mathcal{A})$ with $\mathrm{FV}(T) \subset \{x\}$, there exists an $f \in \mathbf{A}$ such that

$$f.a =_A T[a/x] \qquad \forall a \in \mathbf{A}.$$

Such an element f will be denoted by $\Delta x.T$ in the sequel. For example, as explained in [2], one can define Δ as the standard abstraction λ^* with the help of the combinators $\mathbf{k}$ and $\mathbf{s}$. In the sequel we refer to Δ as an arbitrary abstraction operation on $\mathbf{A}$, which exists due to combinatory completeness.

The set Λ of pure lambda terms is a combinatory algebra, viz.,

$$\Lambda = \langle \Lambda , . , \lambda xy.x , \lambda xyz.xz(yz) , =_\beta \rangle .$$

One can choose in this case Δ to be just the abstraction operation λ on pure terms.

There is a natural mapping from Λ to any other combinatory algebra $\mathcal{A}$. Let $\rho : \mathbf{Var} \to \mathbf{A}$. The interpretation $[\,]_\rho$ of the lambda-terms into $\mathbf{A}$ is defined as follows.

$$\begin{aligned} [v]_\rho &= \rho(v) \\ [T_1 T_2]_\rho &= [T_1]_\rho [T_2]_\rho \\ [\lambda v.T]_\rho &= \Delta v.[T]_{\rho[v:=v]}. \end{aligned}$$

As was pointed out to us by Th. Altenkirch, it is not true in general that, if $T_1 =_\beta T_2$, then $[T_1]_\rho =_A [T_2]_\rho$. In [2] it is shown that this holds for a special case of combinatory algebras - the so called λ-models where Δ is chosen to be λ^* and in which additional axioms hold (see [2], page 94-95). If one considers an arbitrary abstraction Δ (as we do), then it is convenient to take weakly-extensional combinatory algebras to model Λ.

Let $\sim$ be a binary relation on $\mathbf{A}$. For $T_1, T_2 \in \Lambda$ we say that $T_1 = T_2$ is true in the ca $\mathcal{A}$ w.r.t. $\sim$ (notation $\mathcal{A}, \sim \models T_1 = T_2$), if for every valuation ρ, $[T_1]_\rho \sim [T_2]_\rho$. The above notion of satisfaction is easily extended to arbitrary first-order equational formulas over Λ.

Definition 2.5 The equivalence realtion $\sim$ is *weakly-extensional* over $=_A$ if,

- $=_A \subseteq \sim$;
- if $a_1 \sim a_2$ and $b_1 \sim b_2$, then $a_1 a_2 \sim b_1 b_2$;
- $\mathcal{A}, \sim \models \forall x(T_1 = T_2) \to \lambda x.T_1 = \lambda x.T_2$.

Now we can prove the following lemma.

Lemma 2.6 Let $\mathcal{A}$ be a ca and $\sim$ a weakly-extensional relation over $=_A$. Then, for all ρ,

$$\text{if } T_1, T_2 \in \Lambda \text{ and } T_1 =_\beta T_2, \text{ then } [T_1]_\rho \sim [T_2]_\rho.$$

Examples 2.7
- The relation $\sim = \mathbf{A} \times \mathbf{A}$ is weakly-extensional over $=_A$, because it relates all elements of $\mathbf{A}$;
- Let Δ be the abstraction λ^* defined with the help of $\mathbf{k}$ and $\mathbf{s}$ (see [2], page 90). Any congruence relation which contains $=_A$ and satisfies the equations $\mathbf{A}_\beta$ and Meyer-Scott axiom (see [2], page 94-95) is weakly-extensional over $=_A$;
- In the combinatory algebra $\Lambda = \langle\, \Lambda\ ,\ .\ ,\ \lambda xy.x\ ,\ \lambda xyz.xz(yz)\ ,\ =_\beta \,\rangle$, the β-equality is weakly-extensional over itself (if Δ is taken to be λ).

3 The Model Construction

The notion of *CC-structure* and the interpretations of the typable terms of CC are explained informally in the next paragraphs. For more details about the intuition see [15].

The typable terms of CC are mapped into a (set-theoretical) hierarchical structure (called CC-structure) according to their classification as objects, constructors or kinds. The predicative universe of CC is interpreted as a collection $\mathcal{U}^\Box$ of sets (*predicative structure*) and every kind is mapped to a predicative set. Predicative structures are closed under set-theoretical dependent products. The impredicative universe $*$ is interpreted as a collection $\mathcal{U}^*$ of subsets of the underlying ca. We call this collection *polystructure* and its elements *polysets*. $\mathcal{U}^*$ itself is an element of $\mathcal{U}^\Box$ and is closed under non-empty intersections and dependent products (to be defined). Constructors are interpreted as elements of $\bigcup_{X \in \mathcal{U}^\Box} X$ ($\bigcup \mathcal{U}^\Box$ in short). Their interpretations are called *poly-functionals*. In particular, types are mapped to polysets.

Due to the various dependencies in CC, kinds have two other interpretations, as polysets and as elements of the underlying ca, and constructors have a second interpretation as elements of the ca. Three interpretation functions are defined by simultaneous induction on the structure of typable terms: $[\![\ ]\!]^\Box$ to map kinds to predicative sets, $[\![\ ]\!]^*$ to map constructors and kinds to polyfunctionals, and $(\!|\ |\!)$ to map kinds, constructors and objects to elements of the ca. For these interpretations the following Soundness result is proved:

$$\begin{array}{lll}
\Gamma \vdash A : \Box & \Rightarrow [\![A]\!]^\Box_{\xi,\rho} \in \mathcal{U}^\Box & [\![A]\!]^*_{\xi,\rho} \in \mathcal{U}^* \\
\Gamma \vdash P : A : \Box & \Rightarrow [\![P]\!]^*_{\xi,\rho} \in [\![A]\!]^\Box_{\xi,\rho} & (\!|P|\!)_\rho \in [\![A]\!]^*_{\xi,\rho} \\
\Gamma \vdash \sigma : * & \Rightarrow [\![\sigma]\!]^*_{\xi,\rho} \in \mathcal{U}^* & \\
\Gamma \vdash t : \sigma : * & \Rightarrow (\!|t|\!)_\rho \in [\![\sigma]\!]^*_{\xi,\rho} &
\end{array}$$

Here, ξ and ρ are valuations: ξ assigns a *poly-functional* to every constructor variable and ρ assigns an element of $\mathbf{A}$ to every constructor variable and object variable.

Now we are ready to give a formal definition of a class of mathematical structures which constitute models of CC. Let $\mathcal{A}$ be a ca in the sequel.

Definition 3.1 The operation of *dependent product* Π_A on $\mathcal{A}$ takes as arguments a subset X of $\mathbf{A}$ and a function $F : X \to \wp(\mathbf{A})$ and is defined as:

$$\prod_A(X,F) := \{f \in \mathbf{A} \mid \forall n \in X(f.n \in F(n))\}$$

Note that $X = \emptyset$ implies $\prod_A(X,F) = A$, and if $X \neq \emptyset$ and $F(x) = \emptyset$ for some $x \in X$ then $\prod_A(X,F) = \emptyset$. For convenience $\prod_A(X,F)$ will be denoted by $\prod_A x \in X.F(x)$. Like in CC, if F is a constant function on X, say $F(x) = Y$, then we denote $\prod_A(X,F)$ as a function space $X \to Y$, which is defined as $\{f \in \mathbf{A} \mid \forall n \in X(f.n \in Y)\}$.

The impredicative universe of CC is interpreted as a *polystructure*. The impredicativity (or polymorphism) is modeled by requiring polystructures to be closed under arbitrary intersections.

Definition 3.2 Let $\mathcal{A}$ be a ca and $\star \in \mathbf{A}$. A *sufficient* subset of $\mathbf{A}$ w.r.t. $\star$ is a set $\overline{\mathbf{A}}$, such that

1. $\emptyset \subsetneq \overline{\mathbf{A}} \subseteq \mathbf{A}$;
2. If $t[a] \in \overline{\mathbf{A}}$ for some $a \in \mathbf{A}$, then $\Delta x.t[x] \in \overline{\mathbf{A}}$;
3. If $\overrightarrow{a} \in \overline{\mathbf{A}}$, then $\star\overrightarrow{a} \in \overline{\mathbf{A}}$;
4. If $t[a] \in \overline{\mathbf{A}}$, $a \in \overline{\mathbf{A}}$, then $(\Delta x.t[x])a \in \overline{\mathbf{A}}$.

Examples 3.3 The set $\mathbf{A}$ is a sufficient subset of itself (taking for $\star$ an arbitrary element of $\mathbf{A}$). Furthermore, **SN** , the set of β-strongly-normalizing pure λ-terms, is a sufficient subset of Λ w.r.t. x, for any variable x. To show this, take the ca $\langle \Lambda, \cdot, \lambda xy.x, \lambda xyz.xz(yz), =_\beta \rangle$ and Δ to be λ.

Definition 3.4 Let $\mathcal{A}$ be a ca, $\star \in \mathbf{A}$ and $\overline{\mathbf{A}}$ a sufficient subset of $\mathbf{A}$ w.r.t. $\star$. A *polystructure* over $\mathcal{A}$, $\overline{\mathbf{A}}$ and $\star$ is a collection $\mathcal{P} \subseteq \wp(\overline{\mathbf{A}})$, such that the following conditions hold.

(i) $\overline{\mathbf{A}} \in \mathcal{P}$;
(ii) $\mathcal{P}$ is closed under dependent products, i.e. for every $X \in \mathcal{P}$ and every function $F : X \to \mathcal{P}$, $\prod_A x \in X.F(x) \in \mathcal{P}$.
(iii) $\mathcal{P}$ is closed under non-empty intersections, i.e., if I is a nonempty set and $X_i \in \mathcal{P}$ for every $i \in I$ then $\bigcap_{i \in I} X_i \in \mathcal{P}$;
(iv) for all $X \in \mathcal{P}$, if $t[a] \in X$ for some $a \in \overline{\mathbf{A}}$, then $(\Delta x.t[x])a \in X$;
(v) for all $X \in \mathcal{P}$, if $a \in X$ and $b \in \overline{\mathbf{A}}$, then $\mathbf{k}ab \in X$.

The elements of a polystructure are called *polysets*.

Remark 3.5 If $\emptyset \in \mathcal{P}$, then $\overline{\mathbf{A}} = \mathbf{A}$ due to the requirements that $\mathcal{P} \subseteq \wp(\overline{\mathbf{A}})$ and that polystructures should be closed under dependent products, since $\emptyset \to \overline{A} = \mathbf{A}$.

Examples 3.6 Let $\mathcal{A}$ be a ca.

1. A *saturated set* is a set X of strongly normalizing λ-terms such that $y\overrightarrow{P} \in X$ for every variable y and $\overrightarrow{P} \in \mathbf{SN}$ and, if $M[Q/y]\overrightarrow{P} \in X$ and $Q \in \mathbf{SN}$,

then $(\lambda y.M)Q\overrightarrow{P} \in X$. The set of saturated sets is denoted by **SAT** . **SAT** is a polystructure over the ca $\langle \Lambda, \cdot, \lambda xy.x, \lambda xyz.xz(yz), =_\beta \rangle$, **SN** and x (for any variable x).

2. The set $\{\emptyset, \mathbf{A}\}$ is a polystructure over $\mathcal{A}$, $\mathbf{A}$ and $\star$, for any element $\star \in \mathbf{A}$.
3. The set $\mathcal{P} := \{X \subseteq \mathbf{A} \mid X \text{ is closed under } =_A\}$ is a polystructure over $\mathcal{A}$, $\mathbf{A}$ and $\star$, for any element $\star \in \mathbf{A}$.

We shall often be concerned with 'simple' kinds of polystructures, like the ones in the last two examples, where all the polysets are closed under $=_A$ and the sufficient subset is just $\mathbf{A}$ itself. We therefore give the following definition.

Definition 3.7 Let $\mathcal{A}$ be a ca. A *simple polystructure* over $\mathcal{A}$ is a collection $\mathcal{P} \subseteq \wp(A)$, such that the following conditions hold.

(i) $\mathbf{A} \in \mathcal{P}$;
(ii) $\mathcal{P}$ is closed under arbitrary nonempty intersections;
(iii) $\mathcal{P}$ is closed under dependent products;
(iv) Every element of $\mathcal{P}$ is closed under the equivalence relation $=_A$.

If one just works with simple polystructures, the relation $=_A$ is not really necessary; instead one could just look at the quotient algebra $\mathbf{A}/=_A$. (We are also interested in the polystructure of saturated sets, which is not simple.) Note that simple polystructures are still intensional: if X and Y are polysets and $f, g \in X \to Y$, then $\forall x \in X[fx =_A gx]$ does not necessarily imply $f =_A g$.

The predicative universe $\Box$ is interpreted as a *predicative structure*. The necessary properties of predicative structures are derived from the rules of CC. A predicative structure contains a polystructure as an element and is closed under a restricted set-theoretical product.

Definition 3.8 Let $\mathcal{A}$ be a ca and $\sim$ a binary relation on $\mathcal{A}$. The operation $\widetilde{\Pi}$ takes as arguments a subset X of $\mathbf{A}$ and a function $F : X \to SET$, and is defined by:

$$\widetilde{\Pi}(X, F) := \{f \in \Pi\, x \in X.F(x) \mid \forall x_1, x_2 \in X(x_1 \sim x_2 \Longrightarrow f(x_1) = f(x_2))\}$$

Here, $\Pi\, x \in X.F(x)$ denotes the set of functions f such that for all $x \in X$, $f(x) \in F(x)$ (the set-theoretical dependent product).

Note that, if $X = \emptyset$ then $\widetilde{\Pi}(X, F) = \{\emptyset\}$, where $\emptyset$ ambiguously denotes the empty function. Furthermore, if $X \neq \emptyset$ and $F(x) = \emptyset$ for some $x \in X$, then $\widetilde{\Pi}(X, F) = \emptyset$. (The same holds if $F(x_1) \bigcap F(x_2) = \emptyset$ for some $\sim$-related elements x_1 and x_2.) For convenience, $\widetilde{\Pi}(X, F)$ will be denoted by $\widetilde{\Pi}\, x \in X.F(x)$.

Definition 3.9 A *predicative structure over a polystructure $\mathcal{P}$ and a relation* $\sim$ (on $\mathcal{A}$) is a collection of sets $\mathcal{N}$ such that

(i) $\mathcal{P} \in \mathcal{N}$;
(ii) $\mathcal{N}$ is closed under set-theoretical dependent product, Π, i.e. if $B \in \mathcal{N}$ and $F : B \to \mathcal{N}$, then $\Pi\, b \in B.F(b) \in \mathcal{N}$
(iii) $\mathcal{N}$ is closed under $\widetilde{\Pi}$ for *$\sim$-preserving functions*, i.e. if $X \subseteq \mathbf{A}$ and $F : X \to \mathcal{N}$ such that $\forall x_1, x_2 \in X.x_1 \sim x_2 \Longrightarrow F(x_1) = F(x_2)$, then

$$\widetilde{\Pi}\, x \in X.F(x) \in \mathcal{N}.$$

An example of a predicative structure is the collection SET of all sets.

For convenience we introduce some notations. If $f(b) \in F(b)$ for all $b \in B$, then $\boldsymbol{\lambda}\, b \in B.\, f(b)$ denotes the function $b \longmapsto f(b)$. If $g(b) \in F(b)$ and $g(b_1) = g(b_2)$ whenever $b_1 \sim b_2$, then $\tilde{\boldsymbol{\lambda}}\, x \in B.\, g(x)$ denotes the function $b \longmapsto g(b)$.

Now we are ready to give the definition of CC-structures and to define the interpretations of typable terms into such CC-structures.

Definition 3.10 A *CC-structure* is a tuple $\mathcal{M} = \langle \mathcal{A}, \overline{\mathbf{A}}, \star, \sim, \mathcal{U}^*, \mathcal{U}^\Box \rangle$, where

1. $\mathcal{A}$ is a ca;
2. $\overline{\mathbf{A}}$ is a sufficient subset of $\mathbf{A}$ w.r.t. $\star$;
3. $\star$ is a fixed element of $\mathbf{A}$;
4. $\sim$ is a weakly-extensional equivalence relation over $=_A$ (see def.2.5); [3].
5. $\mathcal{U}^*$ is a polystructure over $\mathcal{A}$, $\overline{\mathbf{A}}$ and $\star$;
6. $\mathcal{U}^\Box$ is a predicative structure over $\mathcal{U}^*$ and $\sim$;

Definition 3.11 An *atom-valuation* of constructor and object variables is any map $\rho : \mathbf{Var}^* \bigcup \mathbf{Var}^\Box \to \mathbf{A}$. A *constructor-valuation* of constructor variables is a map $\xi : \mathbf{Var}^\Box \to \bigcup_{X \in \mathcal{N}} X$.

Definition 3.12 The *atom-interpretations* of the typable terms under an atom-valuation ρ are defined as follows

$$\begin{array}{|rl|}\hline
(\!| * |\!)_\rho & := \star \\
(\!| v |\!)_\rho & := \rho(v) \quad \text{if } v \text{ is a variable} \\
(\!| T_1 T_2 |\!)_\rho & := (\!| T_1 |\!)_\rho . (\!| T_2 |\!)_\rho \\
(\!| \lambda v : T_1 . T_2 |\!)_\rho & := \mathbf{k} . (\Delta v . (\!| T_2 |\!)_{\rho[v:=v]}) . (\!| T_1 |\!)_\rho \\
(\!| \textstyle\prod v : T_1 . T_2 |\!)_\rho & := \star . (\!| T_1 |\!)_\rho . (\Delta v . (\!| T_2 |\!)_{\rho[v:=v]}) \\ \hline
\end{array}$$

Remark 3.13 As usual (see [2]), $(\!| T |\!)_{\rho[v:=v]}$ denotes the term over $\mathcal{A}$ obtained from T by applying the map $(\!|\ |\!)_{\rho'}$ to it, where $\rho' : \mathbf{Var} \to \mathcal{T}\ (\mathbf{A})$ is defined as

$$\rho'(u) = \begin{cases} \rho(u) & \text{if } u \neq v, \\ v & \text{if } u = v \end{cases}$$

Fact 3.14 Due to the fact that $\sim$ simulates the equality on a weakly extensional combinatory algebra, the following holds:

1. If $m_1, m_2 \in \mathbf{A}$ and $m_1 \sim m_2$, then $(\!| T |\!)_{\rho[v:=m_1]} \sim (\!| T |\!)_{\rho[v:=m_2]}$.
2. If $T_1 =_\beta T_2$, then $(\!| T_1 |\!)_\rho \sim (\!| T_2 |\!)_\rho$.
3. $(\!| T_1[T_2/v] |\!)_\rho = (\!| T_1 |\!)_{\rho[v:=(\!| T_2 |\!)_\rho]}$.

Definition 3.15 Let ρ be an atom-valuation and ξ a constructor-valuation. The *$\mathcal{U}^*$-interpretation* of kinds and constructors

$$[\![\ ]\!]^*_{\xi,\rho} : \{\Box\} \cup \mathbf{Kind} \cup \mathbf{Constr} \longrightarrow \bigcup \mathcal{U}^\Box$$

and the *$\mathcal{U}^\Box$-interpretation* of kinds

$$[\![\ ]\!]^\Box_{\xi,\rho} : \{\Box\} \cup \mathbf{Kind} \longrightarrow \mathcal{U}^\Box$$

[3] Note that we do not require $s \nsim k$, i.e. $A/\!\sim$ is not necessarily a (weakly-extensional) ca.

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

$$
\begin{array}{rl}
[\![*]\!]^{\Box}_{\xi,\rho} := [\![\Box]\!]^{\Box}_{\xi,\rho} := & \mathcal{U}^* \\
[\![\Pi\alpha : A.B]\!]^{\Box}_{\xi,\rho} := & \Pi\, a \in [\![A]\!]^{\Box}_{\xi,\rho}.\, \widetilde{\Pi}\, m \in [\![A]\!]^{*}_{\xi,\rho}.[\![B]\!]^{\Box}_{\xi[\alpha:=a],\rho[\alpha:=m]} \quad \text{if } A, B \in \mathbf{Kind} \\
[\![\Pi x : \sigma.B]\!]^{\Box}_{\xi,\rho} := & \widetilde{\Pi}\, m \in [\![\sigma]\!]^{*}_{\xi,\rho}.[\![B]\!]^{\Box}_{\xi,\rho[x:=m]} \quad \text{if } \sigma \in \mathbf{Type}\ ,\ B \in \mathbf{Kind}
\end{array}
$$

$$
\begin{array}{rl}
[\![*]\!]^{*}_{\xi,\rho} := [\![\Box]\!]^{*}_{\xi,\rho} := & \overline{\mathbf{A}} \\
[\![\alpha]\!]^{*}_{\xi,\rho} := & \xi(\alpha) \quad \text{if } \alpha \in \mathbf{Var}^{\Box} \\
[\![\Pi\alpha : A.B]\!]^{*}_{\xi,\rho} := & \bigcap_{a \in [\![A]\!]^{\Box}_{\xi,\rho}} \prod_A m \in [\![A]\!]^{*}_{\xi,\rho}.[\![B]\!]^{*}_{\xi[\alpha:=a],\rho[\alpha:=m]} \quad \text{if } A \in \mathbf{Kind} \\
[\![\Pi x : \sigma.B]\!]^{*}_{\xi,\rho} := & \prod_A m \in [\![\sigma]\!]^{*}_{\xi,\rho}.[\![B]\!]^{*}_{\xi,\rho[x:=m]} \quad \text{if } \sigma \in \mathbf{Type} \\
[\![PQ]\!]^{*}_{\xi,\rho} := & [\![P]\!]^{*}_{\xi,\rho}([\![Q]\!]^{*}_{\xi,\rho})((|Q|)_\rho) \quad \text{if } P, Q \in \mathbf{Constr} \\
[\![Pt]\!]^{*}_{\xi,\rho} := & [\![P]\!]^{*}_{\xi,\rho}((|t|)_\rho) \quad \text{if } P \in \mathbf{Constr}\ ,\ t \in \mathbf{Obj} \\
[\![\lambda\alpha : A.P]\!]^{*}_{\xi,\rho} := & \boldsymbol{\lambda}\ a\ \in\ [\![A]\!]^{\Box}_{\xi,\rho}\ .\ \tilde{\boldsymbol{\lambda}}\ m\ \in\ [\![A]\!]^{*}_{\xi,\rho}.\ [\![P]\!]^{*}_{\xi[\alpha:=a],\rho[\alpha:=m]} \\
& \text{if } A \in \mathbf{Kind}\ , P \in \mathbf{Constr} \\
[\![\lambda x : \sigma.P]\!]^{*}_{\xi,\rho} := & \tilde{\boldsymbol{\lambda}}\ m\ \in\ [\![\sigma]\!]^{*}_{\xi,\rho}.\ [\![P]\!]^{*}_{\xi,\rho[x:=m]} \quad \text{if } \sigma \in \mathbf{Type}\ , P \in \mathbf{Constr}
\end{array}
$$

Remark 3.16 The interpretations $[\![\ \ ]\!]^{\Box}_{\xi,\rho}$ and $[\![\ \ ]\!]^{*}_{\xi,\rho}$ may be undefined. For example, if the first argument of the operation $\widetilde{\Pi}$ is not a subset of $\mathbf{A}$ or the abstraction $\tilde{\boldsymbol{\lambda}}$ has the wrong arguments. We will show that, for well-typed terms, the interpretations are well-defined indeed.

For these interpretations the substitution property, which is stated in the next lemma, holds. The relation $\cong$ is 'Kleene-equality'.

Lemma 3.17 Let $t \in \mathbf{Obj}$, $Q \in \mathbf{Constr}$, $T \in \mathbf{Kind} \bigcup \mathbf{Constr}$ and $s \in \{*, \Box\}$. Then:

$$[\![T[Q/\alpha]]\!]^{s}_{\xi,\rho} \cong [\![T]\!]^{s}_{\xi[\alpha:=[\![Q]\!]^{*}_{\xi,\rho}],\rho[\alpha:=(|Q|)_\rho]} \text{ and } [\![T[t/x]]\!]^{s}_{\xi,\rho} \cong [\![T]\!]^{s}_{\xi[x:=(|t|)_\rho],\rho[x:=.]}$$

Definition 3.18 The constructor valuations ξ and the atom valuation ρ *satisfy the context* Γ (notation $\xi, \rho \vDash \Gamma$) if

(i) for every constructor variable α and kind A such that $(\alpha : A) \in \Gamma$,
$$\xi(\alpha) \in [\![A]\!]^{\Box}_{\xi,\rho} \ \text{ and } \ \rho(\alpha) \in [\![A]\!]^{*}_{\xi,\rho}.$$
(ii) for every object variable x and type σ , such that $(x : \sigma) \in \Gamma$,
$$\rho(x) \in [\![\sigma]\!]^{*}_{\xi,\rho}.$$

Definition 3.19 We say that the CC-structure $\mathcal{M}$ *models* $\Gamma \vdash M : T$ (notation $\Gamma \models_{\mathcal{M}} M : T$) iff for every $\xi, \rho \vDash \Gamma$,

(i) If $M \in \mathbf{Kind}$, then $[\![M]\!]^{\Box}_{\xi,\rho} \in \mathcal{U}^{\Box}$, $[\![M]\!]^{*}_{\xi,\rho} \in \mathcal{U}^{*}$, $(|M|)_\rho \in \overline{\mathbf{A}}$;
(ii) If $M \in \mathbf{Constr}$ then $[\![M]\!]^{*}_{\xi,\rho} \in [\![T]\!]^{\Box}_{\xi,\rho}$ and $(|M|)_\rho \in [\![T]\!]^{*}_{\xi,\rho}$;
(iii) If $M \in \mathbf{Obj}$ then $(|M|)_\rho \in [\![T]\!]^{*}_{\xi,\rho}$.

Definition 3.20 If $\Gamma \vdash M_i : T, i = 1, 2$ and $M_1 =_\beta M_2$,we say that the CC-structure $\mathcal{M}$ *models* $M_1 =_\beta M_2$ (notation $\Gamma \models_{\mathcal{M}} M_1 =_\beta M_2$) if for all ξ, ρ such that $\xi, \rho \models \Gamma$,

$$(\!|M_1|\!)_\rho \sim (\!|M_2|\!)_\rho,$$
$$[\![M_1]\!]^s_{\xi,\rho} \cong [\![M_2]\!]^s_{\xi,\rho},$$

for applicable $s \in \{*, \Box\}$.

Definition 3.21 Let $m_1, m_2 \in \mathbf{A}$, $v \in \mathbf{Var}$. We say that m_1 and m_2 are v-*compatible* in the CC-structure $\mathcal{M}$ with respect to $\Gamma \vdash M : T$ (notation $\Gamma, v := m_1, m_2 \models_{\mathcal{M}} M : T$) if for all valuations ξ and ρ, such that $\xi, \rho[v := m_i] \models \Gamma$ $(i = 1, 2)$,

$$(\!|M|\!)_{\rho[v:=m_1]} \sim (\!|M|\!)_{\rho[v:=m_2]},$$
$$[\![M]\!]^s_{\xi,\rho[v:=m_1]} \cong [\![M]\!]^s_{\xi,\rho[v:=m_2]},$$

for applicable $s \in \{*, \Box\}$.

The next theorem says that every CC-structure is a *model* of CC, namely it models every legal judgment of CC.

Theorem 3.22 (Soundness) Let $\mathcal{M}$ be a CC-structure and let Γ be a context and M and T terms such that $\Gamma \vdash M : T$. Then the following holds.

(i) $\Gamma \models_{\mathcal{M}} M : T$;
(ii) for every $m_1, m_2 \in \mathbf{A}$, such that $m_1 \sim m_2$, $\Gamma, v := m_1, m_2 \models_{\mathcal{M}} M : T$;
(iii) if $M \to_\beta N$, then $\Gamma \models_{\mathcal{M}} M =_\beta N$.

Proof. The proof of (i)-(ii) is by simultaneous induction on derivations. The non-trivial cases are: the (λ)-rule, where property (iii) of polystructures is applied (see def.3.4); the $(\prod)$-rules, where the closure of $\mathcal{U}^*$ under non-empty intersections and dependent products and the closure of $\mathcal{U}^\Box$ under set-theoretical products and under $\widetilde{\Pi}$ are used. Furthermore, in the conversion rule the following property is essential. Two typable terms are β-equal (as pseudoterms) iff they are equal via a reduction-expansion path through the set of well-typed terms. (This property follows from Church-Rosser for β and Subject Reduction for β.) In the end, note that to prove the condition (iii) of the Soundness Theorem, Subject Reduction for β is necessary. □

4 Applications

In this section we treat some examples of models of CC that fit in the framework described above. Our main goal hereby is to prove properties about the syntax by employing the models. Typical statements that we can prove in this way are e.g. that the Axiom of Choice is not derivable in CC and that Classical Logic is a consistent extension of CC. The first is proved by constructing a model in which the type that represents the Axiom of Choice is empty and the second is proved by constructing a model in which the type representing the double negation law is inhabited and the interpretation of $\bot$ is empty. The examples

that we show are in the same realm (and sometimes the same) as the ones in [17]. We think (and hope) however that in many cases counterexamples can be constructed more easily using our model construction.

Before going into details, we first compute the interpretations of some logical formulas to observe that their interpretation in the model expresses - roughly - what the formula states. For example, it is easy to check that the interpretation of $\exists x{:}\sigma.\tau$ is not empty iff there exists an element t in $[\![\sigma]\!]^*_{\xi,\rho}$ such that $[\![\tau]\!]^*_{\xi,\rho[x:=t]} \neq \emptyset$.

In this section we restrict ourselves to simple polystructures. (So, $\overline{\mathbf{A}} = \mathbf{A}$, $\emptyset \in \mathcal{P}$ and all polyset are closed under $=_A$.)

Lemma 4.1 In CC-structures with simple polystructures the following holds.

1. $[\![\exists x{:}\sigma.\tau]\!]^*_{\xi,\rho} \neq \emptyset$ iff there exists $t \in [\![\sigma]\!]^*_{\xi,\rho}$ such that $[\![\tau]\!]^*_{\xi,\rho[x:=t]} \neq \emptyset$.
2. If CL is the statement $\Pi\alpha{:}*.\neg\neg\alpha{\to}\alpha$ of Classical Logic, then $[\![\mathrm{CL}]\!]^*_{\xi,\rho} \neq \emptyset$ iff $\bigcap_{X\in\mathcal{U}^*}((X{\to}\emptyset){\to}\emptyset){\to}X \neq \emptyset$.
3. $[\![t =_\sigma q]\!]^*_{\xi,\rho} \neq \emptyset$ iff $(\!|t|\!)_\rho \sim (\!|q|\!)_\rho$, where $=_\sigma$ represents Leibniz equality on σ.
4. The statement PI of Proof-Irrelevance is defined as $\Pi\alpha{:}*.\Pi x,y{:}\alpha.x =_\alpha y$. Then $[\![\mathrm{PI}]\!]^*_{\xi,\rho} \neq \emptyset$ iff for all $t,q \in \mathbf{A}$, $t \sim q$.

It is not true that every formula of higher order predicate logic has such a direct interpretation in the models. As an example we look at the statement of extensionality for propositions, EXT. It is defined as

$$\mathrm{EXT} := \Pi\alpha,\beta{:}*.(\alpha{\leftrightarrow}\beta){\to}\alpha =_* \beta.$$

Here, $=_*$ denotes Leibniz equality on the kind $*$. The interpretation of EXT is

$$\bigcap_{X,Y\in\mathcal{U}^*}\prod_A m,n\in\mathbf{A}.(X{\leftrightarrow}Y){\to}\bigcap_{Q\in\mathcal{U}^*\to\mathbf{A}\to\mathcal{U}^*}(\mathbf{A}{\to}\mathbf{A}){\to}QXm{\to}QYn\,.$$

$$[\![\mathrm{EXT}]\!]^*_{\xi,\rho} \neq \emptyset \text{ iff } \bigcap_{X,Y\in\mathcal{U}^*}(X{\leftrightarrow}Y){\to}\prod_A m,n\in\mathbf{A}.\bigcap_{Q\in\mathcal{U}^*\to\mathbf{A}\to\mathcal{U}^*}QXm{\to}QYn \neq \emptyset$$

$$\text{iff } {\sim} = \mathbf{A}\times\mathbf{A} \text{ and } \mathcal{U}^* = \{\emptyset,\mathbf{A}\}.$$

The fact that $\mathcal{U}^* = \{\emptyset,\mathbf{A}\}$ indeed somehow expresses extensionality of propositions in the model, but ${\sim} = \mathbf{A}\times\mathbf{A}$ does not in any way.

4.1 Classical Logic and Proof-irrelevance

Adding Classical Logic to CC is done by putting $x : \Pi\alpha{:}*.\neg\neg\alpha{\to}\alpha$ as a declaration in the context. It is not difficult to find a polystructure in which $[\![\mathrm{CL}]\!]^*_{\xi,\rho}$ is nonempty, while $[\![\bot]\!]^*_{\xi,\rho}$ is empty ($\bot = \Pi\alpha{:}*.\alpha$). Consider the polystructure $\mathcal{U}^* := \{\emptyset,\mathbf{A}\}$. In this model, $[\![\mathrm{CL}]\!]^*_{\xi,\rho} = \mathbf{A}$, because $((\mathbf{A}{\to}\emptyset){\to}\emptyset){\to}\mathbf{A} = \mathbf{A}$ and $((\emptyset{\to}\emptyset){\to}\emptyset){\to}\emptyset = \mathbf{A}$.

Lemma 4.2 $x : \mathrm{CL}$ is a consistent context of CC.

The statement of Proof-Irrelevance, PI, says that every two elements of a type are equal. Above we have seen that $[\![\mathrm{PI}]\!]^*_{\xi,\rho} \neq \emptyset$ iff $\forall t,q \in \mathbf{A}[t \sim q]$ Now consider the polystructure $\mathcal{U}^* := \{\emptyset,\mathbf{A}\}$ where $\mathbf{A}$ is a weakly-extensional ca and $\sim$ is simply the equality $=_A$ on $\mathbf{A}$, so it does not identify all elements. We find that $[\![\mathrm{PI}]\!]^*_{\xi,\rho} = \emptyset$. Hence we can conclude the following.

Lemma 4.3 In CC there is no term M such that $\vdash M : \mathrm{PI}$. Moreover, there is no term M such that $\vdash M : \mathrm{CL}{\to}\mathrm{PI}$.

The second part of this Lemma can also be reversed. Consider therefore the polystructure $\mathcal{U}^* := \{X \subset \mathbf{A} \mid X \text{ closed under } =_A\}$ and let the equivalence relation $\sim$ be the relation that identifies all elements ($\sim = \mathbf{A} \times \mathbf{A}$). (This makes that $\mathbf{A}/\sim$ is a degenerate λ-algebra, but that is no problem for our construction.) Now, $[\![\mathrm{PI}]\!]^*_{\xi,\rho} \neq \emptyset$, because $\sim$ relates all elements. On the other hand, $[\![\mathrm{CL}]\!]^*_{\xi,\rho} = \emptyset$: take X and Y such that $\emptyset \subsetneq X, Y \subsetneq \mathbf{A}$ and $X \cap Y \neq \emptyset$; then $((X{\to}\emptyset){\to}\emptyset){\to}X = \mathbf{A}{\to}X$ and $((Y{\to}\emptyset){\to}\emptyset){\to}Y = \mathbf{A}{\to}Y$, so $[\![\mathrm{CL}]\!]^*_{\xi,\rho} \subset (\mathbf{A}{\to}X) \cap (\mathbf{A}{\to}Y) = \emptyset$. We have obtained the following result.

Lemma 4.4 In CC there is no term M such that $\vdash M : \mathrm{CL}$. Moreover, there is no term M such that $\vdash M : \mathrm{PI}{\to}\mathrm{CL}$.

In [4], it is shown that there is a term M such that

$$x{:}\mathrm{EXT}, \alpha{:}*, c, c'{:}\alpha, h{:}c \neq_\alpha c' \vdash M{:}\text{`every } f{:}\alpha{\to}\alpha \text{ has a fixed point'}.$$

The statement that 'every $f{:}\alpha{\to}\alpha$ has a fixed point' is written formally as $\Pi f{:}\sigma{\to}\sigma.\exists x{:}\sigma.fx =_\sigma x$. In the models we are looking at here, this is even stronger: we can show that $[\![\mathrm{EXT}{\to}\mathrm{PI}]\!]^*_{\xi,\rho}$ is not empty (and from $[\![\mathrm{PI}]\!]^*_{\xi,\rho} \neq \emptyset$ it easily follows that 'every function has a fixed point' is true in the model). We have seen that $[\![\mathrm{EXT}]\!]^*_{\xi,\rho} \neq \emptyset$ iff $\sim = \mathbf{A} \times \mathbf{A}$ and $\mathcal{U}^* = \{\emptyset, \mathbf{A}\}$. Furthermore, we have seen in Lemma 4.1 that $[\![\mathrm{PI}]\!]^*_{\xi,\rho} \neq \emptyset$ iff for all $t, q \in \mathbf{A}$, $t \sim q$. So, we can conclude that, if $[\![\mathrm{EXT}]\!]^*_{\xi,\rho} \neq \emptyset$, then $[\![\mathrm{PI}]\!]^*_{\xi,\rho} \neq \emptyset$. Hence $[\![\mathrm{EXT}{\to}\mathrm{PI}]\!]^*_{\xi,\rho} \neq \emptyset$.

The interpretation of 'every $f{:}\sigma{\to}\sigma$ has a fixed point' is (writing Y for $[\![\sigma]\!]^*_{\xi,\rho}$) $\prod_A g \in Y{\to}Y.[\![\exists x{:}\sigma.fx =_\sigma x]\!]^*_{\xi,\rho[f:=g]}$.

If $[\![\mathrm{PI}]\!]^*_{\xi,\rho} \neq \emptyset$ in the model, then for all $t \in Y$ and $g \in Y{\to}Y$ we have $[\![fx =_\sigma x]\!]^*_{\xi,\rho[x,f:=t,g]}$, simply because $[\![y =_\sigma x]\!]^*_{\xi,\rho[x,y:=t,q]} \neq \emptyset$ for all $t, q \in \mathbf{A}$.

So, from $[\![\mathrm{PI}]\!]^*_{\xi,\rho} \neq \emptyset$ we conclude that every $f{:}\sigma{\to}\sigma$ (for any σ) has a fixed point.

Lemma 4.5 The sets $[\![\mathrm{PI}{\to}\text{'for every type } \sigma \text{ every } f{:}\sigma{\to}\sigma \text{ has a fixed point'}]\!]^*_{\xi,\rho}$ and $[\![\mathrm{EXT}{\to}\mathrm{PI}]\!]^*_{\xi,\rho}$ are not empty in our models.

4.2 Axiom of Choice

We now show that the Axiom of Choice is not inhabited in CC by giving a model in which the type AC (representing the Axiom of Choice) is empty. Define

$$\mathrm{AC} := (\Pi x{:}\sigma.\exists y{:}\tau.Rxy){\to}(\exists f{:}\sigma{\to}\tau.\Pi x{:}\sigma.Rx(fx)).$$

Here, σ and τ are two inhabited types and R is a variable of type $\sigma{\to}\tau{\to}*$. (We could have formalized AC in a more general way, by abstracting over σ, τ and R, but if AC above is not inhabited, then a more abstract version of the Axiom of Choice is also not inhabited.) To simplify notation we write AC_1 for $\Pi x{:}\sigma.\exists y{:}\tau.Rxy$ and AC_2 for $\exists f{:}\sigma{\to}\tau.\Pi x{:}\sigma.Rx(fx)$.

We consider the combinatory algebra $\mathbf{A} := \Lambda/\beta\eta$, consisting of $\beta\eta$-equivalence classes of λ-terms. Take as polystructure $\mathcal{U}^* := \{\emptyset, \mathbf{A}\}$ and let $\sim \subset \mathbf{A} \times \mathbf{A}$ be

equality (of $\beta\eta$-equivalence classes). Now,

$$\begin{aligned}[\![\mathrm{AC}_1]\!]^*_{\xi,\rho} = \mathbf{A} \text{ iff } & \textstyle\prod_A m \in [\![\sigma]\!]^*_{\xi,\rho}.[\![\exists y{:}\tau.Rxy]\!]^*_{\xi,\rho[x:=m]} = \mathbf{A} \\ \text{iff } & \text{for all } m \in \mathbf{A},\ [\![\exists y{:}\tau.Rxy]\!]^*_{\xi,\rho[x:=m]} = \mathbf{A} \\ \text{iff } & \text{for all } m \in \mathbf{A} \text{ there is a } t \in \mathbf{A} \text{ with } \xi(R)(m,t) = \mathbf{A}\end{aligned}$$

(Note that $[\![\sigma]\!]^*_{\xi,\rho} = \mathbf{A}$, because σ is inhabited and furthermore note that, if for all $m \in \mathbf{A}$, $P(m) = \mathbf{A}$, then $\prod_A m \in \mathbf{A}.P(m) = \mathbf{A}$.) We also find that

$$[\![\mathrm{AC}_2]\!]^*_{\xi,\rho} = \mathbf{A} \text{ iff for some } f \in \mathbf{A}{\to}\mathbf{A},\ \xi(R)(m,fm) = \mathbf{A} \text{ for all } m \in \mathbf{A}.$$

If we define $\xi(R)(m,t) := \mathbf{A}$ iff $m \neq t$, then $\forall m \in \mathbf{A}\exists t \in \mathbf{A}[m \neq t]$, so $[\![\mathrm{AC}_1]\!]^*_{\xi,\rho} = \mathbf{A}$, but not $\exists f \in \mathbf{A}{\to}\mathbf{A}\forall m \in \mathbf{A}[fm \neq m]$ (because of the fixed point theorem for the lambda calculus), so $[\![\mathrm{AC}_2]\!]^*_{\xi,\rho} = \emptyset$. We conclude that $[\![\mathrm{AC}]\!]^*_{\xi,\rho} = \emptyset$ in this model.

Lemma 4.6 There is no closed term of type AC in CC.

One may wonder what happens if one makes the type AC more concrete, e.g. is AC inhabited for all *closed* types σ and τ and all *closed* predicates R? By adapting the construction above a little bit it can be shown that this question has to be answered in the negative.

Consider the model described above (with $\mathbf{A} = \Lambda/\beta\eta$, $\mathcal{U}^* = \{\emptyset, \mathbf{A}\}$ and $\sim$ the equality between $\beta\eta$-equivalence classes). Take for both σ and τ the type of Church numerals, N, and take for R the predicate $\lambda x, y{:}N.x \neq_N y$. The interpretation of N is $\mathbf{A}$ and, using Lemma 4.1 we see that for $m, t \in \mathbf{A}$, $[\![Rxy]\!]^*_{\xi,\rho[x,y:=m,t]} = \mathbf{A}$ iff $m \neq t$. Similarly as above, we find that $[\![AC_1]\!]^*_{\xi,\rho} \neq \emptyset$ and $[\![AC_2]\!]^*_{\xi,\rho} = \emptyset$.

Lemma 4.7 There are closed types σ and τ and a closed predicate R such that there is no closed term of type AC.

Of course, this still leaves the question open whether the Axiom of Choice holds as a rule, that is, whether the following holds. If $\vdash M : \Pi x{:}\sigma.\exists y{:}\tau.Rxy$, is there a closed term N of type $\exists f{:}\sigma{\to}\tau.\Pi x{:}\sigma.Rx(fx))$? This question is not addressed here.

5 Strong Normalization

In this subsection we explain how strong normalization can be proved by using a specific model of CC. The approach used here differs from the ones in [10] and [1], where strong normalization is also derived from particular models. Hyland and Ong (see [10]) point out that there are some complications resulting from the fact that $\langle \mathbf{SN}\ , =_\beta \rangle$ is not a conditional partial combinatory algebra (c-pca). Hence, instead of considering β-equality they work with so-called 'conditionally-weak equality', which is the equality relation generated from the reduction relation $\to_{cw}$, defined by (β) and the rule 'if $M \to_{cw} N$ and $C[-]$ is a term-with-hole such that no free variable of M becomes bound in $C[M]$, then $C[M] \to_{cw} C[N]$'. So, $\to_{cw}$ is not compatible with abstraction; in fact $\to_{cw}$ is a way of restricting

the reduction 'under a λ'. This leads to much additional work for studying properties of cw-equality and the c-pca $\langle \mathbf{SN}\ , =_{cw}\rangle$, while the only equality we are really interested in is the β-equality. Moreover,in [10] models which contain the empty set as a possible interpretation of a type are excluded.

Altenkirch [1] presents a simpler solution, by taking the intersection of the collection of partial equivalence relations over the pure λ-terms and the collection of saturated sets (with a modified definition of the notion of 'saturated set', slightly different from 3.6) as interpretation of $*$.

Our approach is based on the fact that the full collection of saturated sets (see 3.6, or [3]) is a polystructure over the set of pure λ-terms Λ. $v\,\overrightarrow{P} \in X$; The following CC-structure is used to prove strong normalization.

$$\mathcal{M} = \langle \Lambda \bigcup \{\star\},\ \mathbf{SN}\ ,\ \star,\ =_\beta,\ \mathbf{SAT}\ ,\ SET\rangle,$$

where Λ is the set of untyped λ-terms and $\mathbf{SN}$ is the set of β-strongly-normalizing λ-terms. In 3.3 it is shown how $\mathbf{SN}$ can be seen as a sufficient subset of Λ. This CC-structure models CC according to the Soundness Theorem 3.22.

Theorem 5.1 (Strong Normalization for CC). If $\Gamma \vdash M : T$, then $M \in SN$.

Proof. We define a maximum element of $[\![A]\!]^{\Box}_{\xi,\rho}$ for every kind A in the following way.

$$\begin{array}{ll} max(\mathbf{SAT}\ ) & = \mathbf{SN} \\ max([\![\prod \alpha : A.B]\!]^{\Box}_{\xi,\rho}) & = \boldsymbol{\lambda}\ a\ \in\ [\![A]\!]^{\Box}_{\xi,\rho}\ .\ \tilde{\boldsymbol{\lambda}}\ m\ \in\ [\![A]\!]^{*}_{\xi,\rho}.\ max([\![B]\!]^{\Box}_{\xi[\alpha:=a],\rho[\alpha:=m]}) \\ max([\![\prod x : \sigma.B]\!]^{\Box}_{\xi,\rho}) & = \tilde{\boldsymbol{\lambda}}\ m\ \in\ [\![\sigma]\!]^{*}_{\xi,\rho}.\ max([\![B]\!]^{\Box}_{\xi,\rho[x:=m]}) \end{array}$$

Let $\rho(v) = v$ for every variable v, and $\xi(\alpha) = max([\![A]\!]^{\Box}_{\xi,\rho})$ for every $(\alpha : A) \in \Gamma$. (This is possible due to the linearity of the legal contexts). It is immediately verified that, for all terms N, $N \in \mathbf{SN}$ iff $(\!|N|\!)_\rho \in \mathbf{SN}$. Furthermore, the so-chosen valuation ρ and ξ obviously satisfy Γ. From the Soundness Theorem it follows that $(\!|M|\!)_\rho \in [\![T]\!]^{*}_{\xi,\rho} \subseteq \mathbf{SN}$. Hence $(\!|M|\!)_\rho \in \mathbf{SN}$, and so $M \in \mathbf{SN}$. □

For a more detailed presentation of the proof of strong normalization see [15].

6 Related Research

The present paper combines and develops further the ideas in [7] and [15]. This results in constructing a relatively simple set-theoretical notion of model of CC being a PTS. It has been shown how syntactical properties of the system can be studied in a semantical way. Furthermore, such an essential property as SN has been shown to be in a close relation with the semantics of CC. The resulting proof of SN is very flexible in a sense that it can be adapted in a modular way to various extensions of CC, such as inductive types and kinds (see [15]). An interesting question is whether the whole model construction can be extended in a modular way to give semantics of richer systems than CC.

We compare our notion of model with the following.

- **Categorical Models**(see for example [11]). We do not use the abstract machinery of category theory and instead present a simple, intuitively grounded notion of model for CC being a PTS.

- **Standard Realizability Models** (see [12, 13]). The differences here are conceptual. As has been mentioned before, realizability models are a convenient tool for describing semantics for impredicative systems in which the type-dependency rule $(*, \Box)$ of PTSs is "encoded" by explicit "lifting" of every type to a special small kind (see [16, 1]. Such models are usually extensional. A semantics of the PTS CC can be obtained from these models via a syntactic mapping from CC-PTS-style to CC-with-lifting. The model described here is intensional and presents a direct meaning of the Calculus of Construction *as a PTS*.

- **Abstract non-categorical model-constructions.** The only such model-construction we know is the one described in [1]. It is a non-trivial presentation of categorical models without using categorical tools. A non-trivial instance of it is the class of standard realizability models. Note, that this abstract notion of model is also for a system with "lifted" types.

In fact the principle difference between our notion of model and the above three classes of models is that we give a direct interpretation of the rules of Pure Type Systems. We present a new class of concrete models, which are intensional. This makes us believe that these models cannot be viewed as a particular instance of the abstract scheme, as for example presented in [1]. In fact we have tried to organize these concrete models in a more general scheme to cover the PERs as well, but we have so far not succeeded. However one can use PERs instead of polystructures as interpretations of $*$ and 'redo' the rest of the construction.

- **Other (partial) models of the PTS CC** (see [8, 4]). In the literature there are models of CC employed for proving strong normalization, in which CC is interpreted via an explicit or implicit syntactical mapping into Girard's system $F\omega$ (see [9, 8]) . Furthermore, there are models in which type-dependencies are not fully disregarded as in [4] where dependencies are eliminated only in the interpretation of kinds. The interpretations in such models are not straightforwardly extendible to richer systems, for example with inductive types, and our notion of models is more flexible in this sense.

Acknowledgments

We would like to thank Thorsten Altenkirch for some illuminating discussions on the subject of (weakly extensional) combinatory algebras. We are also very grateful to Henk Barendregt, Erik Barendsen and Stefano Berardi for helpful discussions on topics related to the subject of this paper. Further, the comments of the two anonymous referees have been very helpful for us to improve the contents of this paper. Finally, we want to thank Erik Barendsen for his help with LaTeX.

References

1. T. Altenkirch. *Constructions, Inductive Types and Strong Normalization.* PhD thesis, Laboratory for the Foundations of Computer Science, University of Edinburgh, 1993.

2. H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics.* North-Holland, Amsterdam, second, revised edition, 1984.
3. H. P. Barendregt. Typed lambda calculi. In Abramski, editor, *Handbook of Logic in Computer Science.* Oxford University Press, 1992.
4. S. Berardi. Encoding of data types in pure construction calculus: a semantic justification. In G. Plotkin and G. Huet, editors, *Logical Enviroments*, pages 30–60, Edinburgh, 1992.
5. S. Berardi. An application of per models to program extraction. *Mathematical Structures in Computer Science*, 3:309–331, 1993.
6. I. Bethke and J. W Klop. Collapsing partial combinatory algebras. Technical report, CWI, The Netherlands, 1995.
7. J. H. Geuvers. Semantics for dependent types (the calculus of constructions) by a 'double' model construction. Technical report, Department of Computer Science, University of Eindhoven, 1995.
8. J. H. Geuvers. A short and flexible proof of strong normalization for the calculus of constructions. In P. Dybjer, B. Nordström, and J. Smith, editors, *Types for Proofs and Programs, Int. Workshop TYPES '94, Båstad, Sweden, LNCS 996*, pages 14–38, Edinburgh, 1995.
9. J.H. Geuvers and M.J. Nederhof. A modular proof of strong normalization for the calculus of constructions. *Journal of Functional Programming*, 1(2):155–189, 1991.
10. J .M. E. Hyland and C.-H. L. Ong. Modified realizability and strong normalization proofs. In M. Bezem and J. F. Groote, editors, *Typed Lambda Calculi and Applications*, 1993.
11. J. M. E. Hyland and M. Pitts. The theory of constructions: Categorical semantics and topos-theoretic models. In Boulder, editor, *AMS notes*, 1987.
12. G. Longo and E. Moggi. Constructive natural deduction and its 'ω-set' interpretation. *Mathematical Structures in Computer Science*, 1:215–254, 1991.
13. Z. Luo. A higher-order calculus and theory abstraction. *IC*, 90:107–137, 1991.
14. J. C. Reynolds. Polymorphism is not set-theoretic. In G. Kahn, D. B. McQueen, and G. Plotkin, editors, *Lecture Notes in Computer Science 173*, 1984.
15. M.T. Stefanova. Schematic proof of strong normalization for barendregt's-cube, 1995. Submitted, also available at http://www.cs.kun.nl/~milena.
16. T. Streicher. *Semantics of Type Theory. Correctness, Completeness and Independence Results.* Progress in Theoretical Computer Science. Birkhäuser, Boston, 1991.
17. T. Streicher. Independence of the induction principle and the axiom of choice in the pure calculus of constructions. *TCS*, 103(2):395–409, 1992.

This article was processed using the LaTeX macro package with LLNCS style