Models and termination of proof-reduction in the $\lambda\Pi$-calculus modulo theory

Gilles Dowek

Models and termination of proof-reduction in the $\lambda\Pi$-calculus modulo theory

Gilles Dowek*

Abstract

We define a notion of model for the $\lambda\Pi$-calculus modulo theory, a notion of super-consistent theory, and prove that proof-reduction terminates in the $\lambda\Pi$-calculus modulo a super-consistent theory. We prove this way the termination of proof-reduction in two theories in the $\lambda\Pi$-calculus modulo theory, and their consistency: an embedding of Simple type theory and an embedding of the Calculus of constructions.

1 Introduction

1.1 Models and algebras

In Predicate logic and in *Deduction modulo theory* [5, 7], a model is defined by a set $\mathcal{M}$, the *domain* of the model, a set $\mathcal{B}$ of *truth values*, and a function, parametrized by a valuation ϕ, mapping each term t to an element $[\![t]\!]_\phi$ of $\mathcal{M}$, and each proposition A to an element $[\![A]\!]_\phi$ of $\mathcal{B}$.

In the usual definition of the notion of model, the set $\mathcal{B}$ is a two-element set $\{0,1\}$, but this notion can be extended to a notion of *many-valued model*, where $\mathcal{B}$ is an arbitrary Boolean algebra, a Heyting algebra, a pre-Boolean algebra [1], or a pre-Heyting algebra [4]. Boolean algebras permit to introduce intermediate truth values for propositions that are neither provable nor disprovable, Heyting algebras permit to consider models where the excluded middle is not necessarily valid, that is models of constructive Predicate logic, and pre-Boolean and pre-Heyting algebras, where the order $\leq$ is replaced by a pre-order relation, permit to distinguish a notion of weak equivalence: for all valuations ϕ, ($[\![A]\!]_\phi \leq [\![B]\!]_\phi$ and $[\![B]\!]_\phi \leq [\![A]\!]_\phi$), from a notion of strong equivalence: for all valuations ϕ, $[\![A]\!]_\phi = [\![B]\!]_\phi$. The first corresponds to the provability of $A \Leftrightarrow B$ and the second to the *congruence* defining the *computational equality* in Deduction modulo theory [5, 7], also known as *definitional equality* in Constructive type theory [10, 11].

In a model valued in a Boolean algebra, a Heyting algebra, a pre-Boolean algebra, or a pre-Heyting algebra, a proposition A is *valid* when it is weakly equivalent to the proposition $\top$, that is when for all valuations ϕ, $[\![A]\!]_\phi \geq \tilde{\top}$, and this condition boils down to $[\![A]\!]_\phi = \tilde{\top}$ in Boolean and Heyting algebras. A congruence $\equiv$ defined on propositions is *valid* when for all A and B such that $A \equiv B$, A and B are strongly equivalent, that is for all valuations ϕ, $[\![A]\!]_\phi = [\![B]\!]_\phi$. Note that the relation $\leq$ is used in the definition of the validity of a proposition, but not in the definition of the validity of a congruence.

1.2 Termination of proof-reduction

Proof-reduction terminates in Deduction modulo a theory defined by a set of axioms $\mathcal{T}$ and a congruence $\equiv$, if this theory has a model valued in the pre-Heyting algebra of reducibility

*Inria, 23 avenue d'Italie, CS 81321, 75214 Paris Cedex 13, France, gilles.dowek@inria.fr.

candidates [7, 4]. As a consequence, proof-reduction terminates if the theory is *super-consistent*, that is if for all pre-Heyting algebras $\mathcal{B}$, it has a model valued in $\mathcal{B}$.

For the termination of proof-reduction, the congruence matters, but the axioms do not. Thus, the pre-order relation $\leq$ is immaterial in the algebra of reducibility candidates and it is possible to define it as the trivial relation such that $C \leq C'$ for all C and C', which is a pre-order, but not an order. Such a pre-Heyting algebra is said to be *trivial*. As the pre-order is degenerated, all the conditions defining pre-Heyting algebras, such as $a \mathbin{\tilde{\wedge}} b \leq a$, $a \mathbin{\tilde{\wedge}} b \leq b$, etc. are always satisfied in a trivial pre-Heyting algebra, and a trivial pre-Heyting algebra is just a set equipped with arbitrary operations $\tilde{\wedge}$, $\tilde{\Rightarrow}$, etc. Thus, in order to prove that proof-reduction terminates in Deduction modulo a theory defined by a set of axioms $\mathcal{T}$ and a congruence $\equiv$, it is sufficient to prove that for all trivial pre-Heyting algebras $\mathcal{B}$, the theory has a model valued in $\mathcal{B}$.

1.3 Models of the $\lambda\Pi$-calculus modulo theory

In Deduction modulo theory, like in Predicate logic, terms, propositions, and proofs belong to three distinct languages. But, it is also possible to consider a single language, such as the $\lambda\Pi$-calculus modulo theory [3], which is implemented in the Dedukti system [13], or Martin-Löf's Logical Framework [11], and express terms, propositions, and proofs, in this language. For instance, in Deduction modulo theory, 0 is a term, $P(0) \Rightarrow P(0)$ is a proposition and $\lambda\alpha : P(0)\ \alpha$ is a proof of this proposition. In the $\lambda\Pi$-calculus modulo theory, all these expressions are terms of the calculus. Only their types differ: 0 has type nat, $P(0) \Rightarrow P(0)$ has type $Type$ and $\lambda\alpha : P(0)\ \alpha$ has type $P(0) \Rightarrow P(0)$.

The goal of this paper is to define a notion of model for the $\lambda\Pi$-calculus modulo theory, define a notion of super-consistent theory and prove that proof-reduction terminates in the $\lambda\Pi$-calculus modulo a super-consistent theory. We shall this way prove the termination of proof-reduction in two theories in the $\lambda\Pi$-calculus modulo theory: an embedding of Simple type theory [5] and an embedding of the Calculus of constructions [3] in the $\lambda\Pi$-calculus modulo theory.

1.4 Double interpretation

Extending the notion of model to many-sorted predicate logic requires to consider not just one domain $\mathcal{M}$, but a family of domains $\mathcal{M}_s$ indexed by the sorts of the theory, for instance, in a model of Simple type theory, the family of domains is indexed by Simple types. Then, to each term t of sort s is associated an element $[\![t]\!]_\phi$ of $\mathcal{M}_s$ and to each proposition A an element $[\![A]\!]_\phi$ of $\mathcal{B}$.

In the $\lambda\Pi$-calculus modulo theory, the sorts also are just terms of the calculus. Thus, we shall define a model of the $\lambda\Pi$-calculus modulo theory by a family of sets $\mathcal{M}_t$ indexed by the terms of the calculus and a function mapping each term t of type A to an object $[\![t]\!]_\phi$ of $\mathcal{M}_A$. As propositions are just some terms of type $Type$, we shall require that $\mathcal{M}_{Type} = \mathcal{B}$, so that if A is a proposition, then $[\![A]\!]_\phi$ is an element of $\mathcal{B}$.

1.5 Proof-reduction

In Deduction modulo theory, it is possible to define a congruence with a set of rewrite rules that does not terminate, without affecting the termination of proof-reduction. For instance, consider the trivial set of rewrite rules $\mathcal{R}$ containing only the rule $c \longrightarrow c$. Obviously, the congruence

defined by this set of rewrite rules is the identity and proofs modulo this theory are just proofs is pure Predicate logic. Thus, proof-reduction in Deduction modulo this theory terminates. This means that in the $\lambda\Pi$-calculus modulo this theory, the β-reduction terminates. But the $\beta\mathcal{R}$-reduction does not terminate, as the $\mathcal{R}$-reduction alone does not terminate.

Thus, in this paper, we shall restrict to prove the termination of β-reduction, not $\beta\mathcal{R}$-reduction. In some cases the termination of the $\beta\mathcal{R}$-reduction is a simple corollary of the termination of the β-reduction. In some others it is not.

2 The $\lambda\Pi$-calculus modulo theory

2.1 The $\lambda\Pi$-calculus

Definition 2.1 (The syntax of the $\lambda\Pi$-calculus) *The syntax of the $\lambda\Pi$-calculus is*

$$t = x \mid Type \mid Kind \mid \Pi x : t\ t \mid \lambda x : t\ t \mid t\ t$$

As usual, we write $A \to B$ for $\Pi x : A\ B$ when x does not occur in B.

The α-equivalence relation is defined as usual and terms are identified modulo α-equivalence.

The relation β (one step β-reduction at the root) is defined as usual.

As usual, if r is a relation on terms, we write $\longrightarrow_r^1$ for the subterm extension of r, $\longrightarrow_r^+$ for the transitive closure of the relation $\longrightarrow_r^1$, $\longrightarrow_r^*$ for its reflexive-transitive closure, and $\equiv_r$ for its reflexive-symmetric-transitive closure.

Definition 2.2 (The typing rules of the $\lambda\Pi$-calculus) *The typing rules of the $\lambda\Pi$-calculus are*

$$\frac{}{[\,]\ \textit{well-formed}}\ \textbf{Empty}$$

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A\ \ \textit{well-formed}}\ \textbf{Declaration}\ x\ \textit{not in}\ \Gamma$$

$$\frac{\Gamma\ \textit{well-formed}}{\Gamma \vdash Type : Kind}\ \textbf{Sort}$$

$$\frac{\Gamma\ \textit{well-formed}}{\Gamma \vdash x : A}\ \textbf{Variable}\ x : A \in \Gamma$$

$$\frac{\Gamma \vdash A : Type\ \ \Gamma, x : A \vdash B : s}{\Gamma \vdash \Pi x : A\ B : s}\ \textbf{Product}$$

$$\frac{\Gamma \vdash A : Type\ \ \Gamma, x : A \vdash B : s\ \ \Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A\ t : \Pi x : A\ B}\ \textbf{Abstraction}$$

$$\frac{\Gamma \vdash t : \Pi x : A\ B\ \ \Gamma \vdash u : A}{\Gamma \vdash (t\ u) : (u/x)B}\ \textbf{Application}$$

$$\frac{\Gamma \vdash A : s\ \ \Gamma \vdash B : s\ \ \Gamma \vdash t : A}{\Gamma \vdash t : B}\ \textbf{Conversion}\ A \equiv_\beta B$$

where in each rule s is either $Type$ or $Kind$.

It can be proved that types are preserved by β-reduction, that β-reduction is confluent and strongly terminating and that each term has a unique type modulo β-equivalence [9].

Definition 2.3 (Object) *A term t is said to be an* object *in a context Γ, if t has a type A, and A has type $Type$.*

2.2 The $\lambda\Pi$-calculus modulo theory

Recall that if Σ, Γ, and Δ are contexts, a substitution θ, binding the variables declared in Γ, is said to be *of type* $\Gamma \rightsquigarrow \Delta$ *in* Σ if for all x declared of type T in Γ, we have $\Sigma, \Delta \vdash \theta x : \theta T$. In this case, if $\Sigma, \Gamma \vdash u : U$, then $\Sigma, \Delta \vdash \theta u : \theta U$.

Definition 2.4 (Rewrite rule) *A* rewrite rule *is a quadruple* $l \longrightarrow^{\Gamma,T} r$ *where* Γ *is a context and* l*,* r*, and* T *are* β*-normal terms. Such a rule is said to be* well-typed *in the context* Σ *if, in the* $\lambda\Pi$*-calculus, the context* Σ, Γ *is well-formed and the terms* l *and* r *have type* T *in this context.*

If Σ is a context, $l \longrightarrow^{\Gamma,T} r$ is a rewrite rule well-typed in Σ and θ is a substitution of type $\Gamma \rightsquigarrow \Delta$ in Σ then the terms θl and θr both have type θT in the context Σ, Δ.

The relation $\mathcal{R}$ (one step $\mathcal{R}$-reduction at the root) is defined by: $t \ \mathcal{R} \ u$ is there exists a rewrite rule $l \longrightarrow^{\Gamma,T} r$ and a substitution θ such that $t = \theta l$ and $u = \theta r$. The relation $\beta\mathcal{R}$ (one step $\beta\mathcal{R}$-reduction at the root) is the union of β and $\mathcal{R}$.

Definition 2.5 (Theory) *A* theory *is a pair formed with a context* Σ*, well-formed in the* $\lambda\Pi$*-calculus and a set of rewrite rules* $\mathcal{R}$*, well-typed in* Σ *in the* $\lambda\Pi$*-calculus.*

The variables declared in Σ are called *constants* rather than variables. They replace the sorts, the function symbols, the predicate symbols, and also the axioms of Predicate logic.

Definition 2.6 (The $\lambda\Pi$-calculus modulo theory) *The* $\lambda\Pi$-calculus modulo $\Sigma, \mathcal{R}$ *is the extension of the* $\lambda\Pi$*-calculus obtained modifying the* **Declaration** *and* **Variable** *rules to allow the use of constants as well as variables*

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \ \ \textit{well-formed}} \ \textbf{Declaration} \ x \textit{ not in } \Sigma, \Gamma$$

$$\frac{\Gamma \ \textit{well-formed}}{\Gamma \vdash x : A} \ \textbf{Variable} \ x : A \in \Sigma, \Gamma$$

and by replacing the relation $\equiv_\beta$ *by* $\equiv_{\beta\mathcal{R}}$ *in the* **Conversion** *rule*

$$\frac{\Gamma \vdash A : s \ \ \Gamma \vdash B : s \ \ \Gamma \vdash t : A}{\Gamma \vdash t : B} \ \textbf{Conversion} \ A \equiv_{\beta\mathcal{R}} B$$

3 Examples of theories

3.1 Simple type theory

In [6], we have given a presentation of Simple type theory in Deduction modulo theory. This presentation can easily be adapted to the $\lambda\Pi$-calculus modulo theory.

Definition 3.1 (The language of Simple type theory)

$$\iota : Type$$

$$o : Type$$

$$\varepsilon : o \to Type$$

$$\dot{\Rightarrow} : o \to o \to o$$

$$\dot{\forall}_A : (A \to o) \to o$$

for a finite number of Simple types A, where Simple types *are inductively defined by*

- ι *and* o *are Simple types,*
- *if* A *and* B *are Simple types, then* $A \to B$ *is a Simple type.*

Definition 3.2 (The rules of Simple type theory)

$$\varepsilon(\dot{\Rightarrow}\ X\ Y) \longrightarrow \varepsilon(X) \to \varepsilon(Y)$$

$$\varepsilon(\dot{\forall}_A\ X) \longrightarrow \Pi z : A\ \varepsilon(X\ z)$$

3.2 The Calculus of constructions

In [3], we have introduced an embedding of the Calculus of constructions [2] in the $\lambda\Pi$-calculus modulo theory.

Definition 3.3 (The language of the Calculus of constructions)

$$U_{Type} : Type$$

$$U_{Kind} : Type$$

$$\dot{Type} : U_{Kind}$$

$$\varepsilon_{Type} : U_{Type} \to Type$$

$$\varepsilon_{Kind} : U_{Kind} \to Type$$

$$\dot{\Pi}_{\langle Type,Type,Type\rangle} : \Pi X : U_{Type}\ (((\varepsilon_{Type}\ X) \to U_{Type}) \to U_{Type})$$

$$\dot{\Pi}_{\langle Type,Kind,Kind\rangle} : \Pi X : U_{Type}\ (((\varepsilon_{Type}\ X) \to U_{Kind}) \to U_{Kind})$$

$$\dot{\Pi}_{\langle Kind,Type,Type\rangle} : \Pi X : U_{Kind}\ (((\varepsilon_{Kind}\ X) \to U_{Type}) \to U_{Type})$$

$$\dot{\Pi}_{\langle Kind,Kind,Kind\rangle} : \Pi X : U_{Kind}\ (((\varepsilon_{Kind}\ X) \to U_{Kind}) \to U_{Kind})$$

Definition 3.4 (The rules of the Calculus of constructions)

$$\varepsilon_{Kind}(\dot{Type}) \longrightarrow U_{Type}$$

$$\varepsilon_{Type}(\dot{\Pi}_{\langle Type,Type,Type\rangle}\ X\ Y) \longrightarrow \Pi z : (\varepsilon_{Type}\ X)\ (\varepsilon_{Type}\ (Y\ z))$$

$$\varepsilon_{Kind}(\dot{\Pi}_{\langle Type,Kind,Kind\rangle}\ X\ Y) \longrightarrow \Pi z : (\varepsilon_{Type}\ X)\ (\varepsilon_{Kind}\ (Y\ z))$$

$$\varepsilon_{Type}(\dot{\Pi}_{\langle Kind,Type,Type\rangle}\ X\ Y) \longrightarrow \Pi z : (\varepsilon_{Kind}\ X)\ (\varepsilon_{Type}\ (Y\ z))$$

$$\varepsilon_{Kind}(\dot{\Pi}_{\langle Kind,Kind,Kind\rangle}\ X\ Y) \longrightarrow \Pi z : (\varepsilon_{Kind}\ X)\ (\varepsilon_{Kind}\ (Y\ z))$$

4 Super-consistency

4.1 Π-algebras

The notion of Π-algebra is an adaptation to the λΠ-calculus of the notion of (trivial) pre-Heyting algebra.

Definition 4.1 (Π-algebra) *A* Π*-algebra is formed with*

- *a set* $\mathcal{B}$*,*
- *an element* $\tilde{T}$ *of* B*,*
- *a subset* $\mathcal{A}$ *of* $\mathcal{P}(\mathcal{B})$*,*
- *a function* $\tilde{\Pi}$ *from* $\mathcal{B} \times \mathcal{A}$ *to* $\mathcal{B}$*.*

If w and w' are two elements of $\mathcal{B}$, we write $w \tilde{\to} w'$ for $\tilde{\Pi}(w, \{w'\})$.

Definition 4.2 (Full Π-algebra) *A* Π*-algebra is* full *if* $\mathcal{A} = \mathcal{P}(\mathcal{B})$*, that is if* $\tilde{\Pi}$ *is defined for all subsets of* $\mathcal{B}$*.*

Definition 4.3 (Ordered, complete Π-algebra) *A* Π*-algebra is* ordered *if it is equipped with an order relation* $\sqsubseteq$ *such that the operation* $\tilde{\Pi}$ *is left anti-monotonous and right monotonous with respect to* $\sqsubseteq$*, that is*

- *if* $x \sqsubseteq y$*, then for all* S $\tilde{\Pi}(y, S) \sqsubseteq \tilde{\Pi}(x, S)$*,*
- *if* $S \sqsubseteq T$*, then for all* x $\tilde{\Pi}(x, S) \sqsubseteq \tilde{\Pi}(x, T)$*, where* $S \sqsubseteq T$ *is defined as: for all* y *in* S*, there exists a* z *in* T *such that* $y \sqsubseteq z$*.*

It is complete *if every subset of* $\mathcal{B}$ *has a least upper bound for the relation* $\sqsubseteq$*.*

4.2 Models valued in a Π-algebra $\mathcal{B}$

Definition 4.4 (Family of domains valued in a Π-algebra $\mathcal{B}$) *Let* $\mathcal{B} = \langle \mathcal{B}, \tilde{T}, \mathcal{P}(\mathcal{B}), \tilde{\Pi} \rangle$ *be a full* Π*-algebra. A* family of domains valued in $\mathcal{B}$ *is a family* $(\mathcal{M}_t)_t$ *indexed by terms of* λΠ*-calculus modulo theory, such that* $\mathcal{M}_{Kind} = \mathcal{M}_{Type} = \mathcal{B}$*.*

Definition 4.5 (Valuation) *Let* $(\mathcal{M}_t)_t$ *be a family of domains. Let* $\Gamma = x_1 : A_1, ..., x_n : A_n$ *be a well-formed context. A* Γ-valuation onto $\mathcal{M}$ *is a function mapping every variable* x_i *to an element of* $\mathcal{M}_{A_i}$*.*

Definition 4.6 (Model valued in a Π-algebra $\mathcal{B}$) *A* model *is a function mapping each pair formed with a term* t *of type* A *in a context* Γ *and each* Γ*-valuation onto* $\mathcal{M}$ ϕ *to a element* $[\![t]\!]_\phi$ *of* $\mathcal{M}_A$*, such that* $[\![Kind]\!]_\phi = [\![Type]\!]_\phi = \tilde{T}$*,* $[\![x]\!]_\phi = \phi x$*, and* $[\![\Pi x : C\ D]\!]_\phi = \tilde{\Pi}([\![C]\!]_\phi, \{[\![D]\!]_{\phi, x=c} \mid c \in \mathcal{M}_C\})$*.*

Definition 4.7 (Validity) *A theory* $\Sigma, \mathcal{R}$ *is said to be* valid *in a model* $\mathcal{M}$ *if for all* A *and* B *such that* $A \equiv_{\beta\mathcal{R}} B$*, we have* $\mathcal{M}_A = \mathcal{M}_B$ *and* $[\![A]\!]_\phi = [\![B]\!]_\phi$*.*

Definition 4.8 (Super-consistency) *A theory $\Sigma, \mathcal{R}$, is said to be* super-consistent *if for every full, ordered and complete Π-algebra $\mathcal{B}$, there exists a model $\mathcal{M}$ valued in $\mathcal{B}$ such that $\Sigma, \mathcal{R}$ is valid in $\mathcal{M}$.*

To define the notion of validity of a type (axiom) declared in Σ, we should, in supplement, add a pre-order relation $\leq$ on $\mathcal{B}$, with some compatibility conditions between $\tilde{T}$, $\mathcal{A}$, $\tilde{\Pi}$, $\sqsubseteq$ and $\leq$, so that the validity of a type A can be defined as $[\![A]\!]_\phi \geq \tilde{T}$. As our main focus in this paper is proof termination, we leave this for future work and we implicitly consider that all types are valid in all models.

5 Super-consistency of the $\lambda\Pi$-calculus modulo Simple type theory and modulo the Calculus of constructions

5.1 Simple type theory

Let $\langle \mathcal{B}, \tilde{T}, \mathcal{P}(\mathcal{B}), \tilde{\Pi} \rangle$ be a full Π-algebra and $\{e\}$ be an arbitrary one-element set.

Definition 5.1 *We define a family of set $(\mathcal{M}_t)_t$ indexed by terms of the $\lambda\Pi$-calculus modulo theory as follows.*

- $\mathcal{M}_{Kind} = \mathcal{M}_{Type} = \mathcal{M}_o = \mathcal{B}$,
- $\mathcal{M}_\iota = \mathcal{M}_\varepsilon = \mathcal{M}_{\Rightarrow} = \mathcal{M}_{\dot{\forall}_A} = \mathcal{M}_x = \{e\}$,
- $\mathcal{M}_{\lambda x:C\ t} = \mathcal{M}_t$,
- $\mathcal{M}_{(t\ u)} = \mathcal{M}_t$,
- $\mathcal{M}_{\Pi x:C\ D}$ *is the set of functions f from $\mathcal{M}_C$ to $\mathcal{M}_D$, except if $\mathcal{M}_D = \{e\}$, in which case* $\mathcal{M}_{\Pi x:C\ D} = \{e\}$.

Lemma 5.1 *If t does not contain any occurrence of $Kind$, $Type$, or o then $\mathcal{M}_t = \{e\}$.*

Proof. By induction on the structure of t.

Lemma 5.2 *If u does not contain any occurrence of $Kind$, $Type$, or o, then $\mathcal{M}_{(u/x)t} = \mathcal{M}_t$.*

Proof. By induction on the structure of t. If $t = x$ then, by Lemma 5.1, $\mathcal{M}_{(u/x)t} = \mathcal{M}_u = \{e\} = \mathcal{M}_t$. If t is $Kind$, $Type$, ι, o, ε, $\Rightarrow$, $\dot{\forall}_A$, or a variable different from x, then x does not occur in t. If t is an application, an abstraction, or a product, we use the induction hypothesis.

Lemma 5.3 (Conversion) *If $t \equiv_{\beta\mathcal{R}} u$ then $\mathcal{M}_t = \mathcal{M}_u$.*

Proof. If $t = ((\lambda x : C\ t')\ u')$, then u' is an object and it does not contain any occurrence of $Kind$, $Type$, or o. By Lemma 5.2, $\mathcal{M}_{((\lambda x:C\ t')\ u')} = \mathcal{M}_{t'} = \mathcal{M}_{(u'/x)t'}$.

Then, as for all v, $\mathcal{M}_{(\varepsilon\ v)} = \mathcal{M}_\varepsilon = \{e\}$, and if $\mathcal{M}_D = \{e\}$, then $\mathcal{M}_{\Pi x:C\ D} = \{e\}$, we have $\mathcal{M}_{(\varepsilon\ C)\Rightarrow(\varepsilon\ D)} = \{e\} = \mathcal{M}_{(\varepsilon\ (C \Rightarrow D))}$ and $\mathcal{M}_{\Pi x:C\ (\varepsilon\ (D\ x))} = \{e\} = \mathcal{M}_{(\varepsilon\ (\dot{\forall}_C\ D))}$.

We prove, by induction on t, that if $t \longrightarrow^1_{\beta\mathcal{R}} u$ then $\mathcal{M}_t = \mathcal{M}_u$ and we conclude with a simple induction on the structure of the derivation of $t \equiv_{\beta\mathcal{R}} u$.

Definition 5.2 *Let t be a term of type A in a context Γ or that is equal to $Kind$. Let ϕ be a Γ-valuation onto $\mathcal{M}$. The element $[\![t]\!]_\phi$ of $\mathcal{M}_A$ is defined as follows.*

- $[\![Kind]\!]_\phi = [\![Type]\!]_\phi = [\![\iota]\!]_\phi = [\![o]\!]_\phi = \tilde{T}$,
- $[\![x]\!]_\phi = \phi x$,
- $[\![\lambda x : C\ t]\!]_\phi$ *is the function mapping c in $\mathcal{M}_C$ to $[\![t]\!]_{\phi,x=c}$, except if for all c in $\mathcal{M}_C$ $[\![t]\!]_{\phi,x=c} = e$, in which case* $[\![\lambda x : C\ t]\!]_\phi = e$,
- $[\![(t\ u)]\!]_\phi = [\![t]\!]_\phi([\![u]\!]_\phi)$, *except if* $[\![t]\!]_\phi = e$, *in which case* $[\![(t\ u)]\!]_\phi = e$,
- $[\![\Pi x : C\ D]\!]_\phi = \tilde{\Pi}([\![C]\!]_\phi, \{[\![D]\!]_{\phi,x=c} \mid c \in \mathcal{M}_C\})$,
- $[\![\varepsilon]\!]_\phi$ *is the identity on $\mathcal{B}$,*
- $[\![\dot{\Rightarrow}]\!]_\phi$ *is the function mapping w and w' in $\mathcal{B}$ to $w \tilde{\to} w'$,*
- $[\![\dot{\forall}_C]\!]_\phi$ *is the function mapping the function f from $\mathcal{M}_C$ to $\mathcal{B}$ to the element of $\mathcal{B}$* $\tilde{\Pi}([\![C]\!]_\phi, \{fc \mid c \in \mathcal{M}_C\})$.

Lemma 5.4 (Substitution) $[\![(u/x)t]\!]_\phi = [\![t]\!]_{\phi + x = [\![u]\!]_\phi}$

Proof. By induction over the structure of t.

Lemma 5.5 (Conversion) *If $t \equiv_{\beta\mathcal{R}} u$ then $[\![t]\!]_\phi = [\![u]\!]_\phi$.*

Proof. If $t = ((\lambda x : C\ t')\ u')$, then let D be the type of t', if $\mathcal{M}_D = \{e\}$ then $[\![((\lambda x : C\ t')\ u')]\!]_\phi = e = [\![(u'/x)t']\!]_\phi$. Otherwise $[\![((\lambda x : C\ t')\ u')]\!]_\phi = [\![t']\!]_{\phi,x=[\![u']\!]_\phi} = [\![(u'/x)t']\!]_\phi$.

Then $[\![\varepsilon(\dot{\Rightarrow}\ t'\ u')]\!]_\phi = [\![t']\!]_\phi \tilde{\to} [\![u']\!]_\phi = [\![\varepsilon(t') \to \varepsilon(u')]\!]_\phi$ and $[\![\varepsilon(\dot{\forall}_C\ t')]\!]_\phi = \tilde{\Pi}([\![C]\!]_\phi, \{[\![t']\!]_\phi c \mid c \in \mathcal{M}_C\}) = [\![\Pi y : C\ \varepsilon(t'\ y)]\!]_\phi$.

We prove, by induction on t, that if $t \longrightarrow^1_{\beta\mathcal{R}} u$ then $[\![t]\!]_\phi = [\![u]\!]_\phi$ and we conclude with a simple induction on the structure of the derivation of $t \equiv_{\beta\mathcal{R}} u$.

5.2 The Calculus of constructions

In the model of Simple type theory, we had $\mathcal{M}_t = \{e\}$ for all objects t. This allowed to define $\mathcal{M}_{(t\ u)}$ and $\mathcal{M}_{\lambda x:C\ t}$ as $\mathcal{M}_t$ and validate β-reduction trivially. In the model built in this section, we need to take $\mathcal{M}_{\dot{Type}} = \mathcal{B}$ although $\dot{Type}$ is an object, as $\dot{Type} : U_{Kind} : Type$. Thus, we need to define $\mathcal{M}_{\lambda x:A\ t}$ as a function. This leads to define first another family of domains $(\mathcal{N}_t)_t$ and parametrize the definition of $\mathcal{M}_t$ itself by a valuation onto $\mathcal{N}$.

Let $\langle \mathcal{B}, \tilde{T}, \mathcal{P}(\mathcal{B}), \tilde{\Pi} \rangle$ be a full Π-algebra and $\{e\}$ be an arbitrary one-element set. Let E be a set containing $\mathcal{B}$ and $\{e\}$, and closed by function space and arbitrary unions. The existence of such a set can be proved with the replacement scheme.

Definition 5.3 *We define a family $(\mathcal{N}_t)_t$ indexed by terms of the $\lambda\Pi$-calculus modulo theory as follows.*

- $\mathcal{N}_{Kind} = \mathcal{N}_{Type} = \mathcal{N}_{U_{Kind}} = E$,

- $\mathcal{N}_{\Pi x:C\ D}$ *is the set of functions* f *from* $\mathcal{N}_C$ *to* $\mathcal{N}_D$*, except if* $\mathcal{N}_D = \{e\}$*, in which case* $\mathcal{N}_{\Pi x:C\ D} = \{e\}$.
- $\mathcal{N}_{U_{Type}} = \mathcal{N}_{\dot{Type}} = \mathcal{N}_{\varepsilon_{Type}} = \mathcal{N}_{\varepsilon_{Kind}} = \mathcal{N}_{\dot{\Pi}_{\langle Type,Type,Type\rangle}} = \mathcal{N}_{\dot{\Pi}_{\langle Type,Kind,Kind\rangle}} = \mathcal{N}_{\dot{\Pi}_{\langle Kind,Type,Type\rangle}} = \mathcal{N}_{\dot{\Pi}_{\langle Kind,Kind,Kind\rangle}} = \mathcal{N}_x = \{e\}$,
- $\mathcal{N}_{\lambda x:C\ t} = \mathcal{N}_t$,
- $\mathcal{N}_{(t\ u)} = \mathcal{N}_t$.

Lemma 5.6 *If* t *does not contain any occurrence of* $Kind$*,* $Type$*, or* U_{Kind}*, then* $\mathcal{N}_t = \{e\}$.

Proof. By induction on the structure of t.

Lemma 5.7 *If* u *does not contain any occurrence of* $Kind$*,* $Type$*, or* U_{Kind}*, then* $\mathcal{N}_{(u/x)t} = \mathcal{N}_t$.

Proof. By induction on the structure of t. If $t = x$ then, by Lemma 5.6, $\mathcal{N}_{(u/x)t} = \mathcal{N}_u = \{e\} = \mathcal{N}_t$. If t is $Kind$, $Type$, U_{Kind}, U_{Type}, $\dot{Type}$, ε_{Kind}, ε_{Type}, $\dot{\Pi}_{\langle Type,Type,Type\rangle}$, $\dot{\Pi}_{\langle Type,Kind,Kind\rangle}$, $\dot{\Pi}_{\langle Kind,Type,Type\rangle}$, $\dot{\Pi}_{\langle Kind,Kind,Kind\rangle}$, or a variable different from x, then x does not occur in t. If t is an application, an abstraction, or a product, we use the induction hypothesis.

Lemma 5.8 (Conversion) *If* $t \equiv_{\beta\mathcal{R}} u$ *then* $\mathcal{N}_t = \mathcal{N}_u$.

Proof. If $t = ((\lambda x : C\ t')\ u')$, then u' is an object and it does not contain any occurrence of $Kind$, $Type$, or U_{Kind}. By Lemma 5.7, $\mathcal{N}_{((\lambda x:C\ t')\ u')} = \mathcal{N}_{t'} = \mathcal{N}_{(u'/x)t'}$.

We have $\mathcal{N}_{(\varepsilon_{Kind}\ \dot{Type})} = \{e\} = \mathcal{N}_{U_{Type}}$.

Then, as for all v, $\mathcal{N}_{(\varepsilon_s\ v)} = \mathcal{N}_{\varepsilon_s} = \{e\}$, and if $\mathcal{N}_D = \{e\}$, then $\mathcal{N}_{\Pi x:C\ D} = \{e\}$, we have $\mathcal{N}_{\varepsilon_{s_2}(\dot{\Pi}_{\langle s_1,s_2,s_2\rangle}\ C\ D)} = \{e\} = \mathcal{N}_{\Pi x:(\varepsilon_{s_1}\ C)\ (\varepsilon_{s_2}\ (D\ x))}$.

We prove, by induction on t, that if $t \longrightarrow^1_{\beta\mathcal{R}} u$ then $\mathcal{N}_t = \mathcal{N}_u$ and we conclude with a simple induction on the structure of the derivation of $t \equiv_{\beta\mathcal{R}} u$.

Definition 5.4 *We define a family* $(\mathcal{M}_t)_{t,\psi}$ *indexed by terms of the* $\lambda\Pi$*-calculus modulo theory and* Γ*-valuations onto* $\mathcal{N}$*, in such a way that if* t *has type* A *in* Γ*, then* $\mathcal{M}_{t,\psi}$ *is an element of* $\mathcal{N}_A$.

- $\mathcal{M}_{Kind,\psi} = \mathcal{M}_{Type,\psi} = \mathcal{M}_{U_{Kind},\psi} = \mathcal{M}_{U_{Type},\psi} = \mathcal{M}_{\dot{Type},\psi} = \mathcal{B}$,
- $\mathcal{M}_{\varepsilon_{Kind},\psi}$ *is the identity on* E,
- $\mathcal{M}_{\varepsilon_{Type},\psi}$ *is the function from* $\{e\}$ *to* E *mapping* e *to* $\{e\}$,
- $\mathcal{M}_{x,\psi} = \psi x$,
- $\mathcal{M}_{\lambda x:C\ t,\psi}$ *is the function mapping* c *in* $\mathcal{N}_C$ *to* $\mathcal{M}_{t,\psi+x=c}$*, except if for all* c *in* $\mathcal{N}_C$*,* $\mathcal{M}_{t,\psi+x=c} = e$ *in which case* $\mathcal{M}_{\lambda x:C\ t,\psi} = e$,
- $\mathcal{M}_{(t\ u),\psi} = \mathcal{M}_{t,\psi}(\mathcal{M}_{u,\psi})$*, except if* $\mathcal{M}_{t,\psi} = e$ *in which case* $\mathcal{M}_{(t\ u),\psi} = e$,
- $\mathcal{M}_{\Pi x:C\ D,\psi}$ *is the set of functions from* $\mathcal{M}_C$ *to* $\bigcup_{c\in\mathcal{N}_C} \mathcal{M}_{D,\psi+x=c}$*, except if* $\bigcup_{c\in\mathcal{N}_C} \mathcal{M}_{D,\psi+x=c} = \{e\}$ *in which case* $\mathcal{M}_{\Pi x:C\ D,\psi} = \{e\}$,

- $\mathcal{M}_{\dot{\Pi}_{\langle Type,Type,Type\rangle}} = \mathcal{M}_{\dot{\Pi}_{\langle Kind,Type,Type\rangle}} = e$,
- $\mathcal{M}_{\dot{\Pi}_{\langle Type,Kind,Kind\rangle}}$ *is the function mapping* e *and* h *from* $\{e\}$ *to* E *to the set of functions from* $\{e\}$ *to* $(h\ e)$,
- $\mathcal{M}_{\dot{\Pi}_{\langle Kind,Kind,Kind\rangle}}$ *is the function mapping* a *in* E *and* h *from* $\{e\}$ *to* E *to the set of functions from* a *to* $(h\ e)$.

Lemma 5.9 (Substitution) $\mathcal{M}_{(u/x)t,\psi} = \mathcal{M}_{t,\psi+x=\mathcal{M}_u}$

Proof. By induction on the structure of t.

Lemma 5.10 (Conversion) *If* $t \equiv_{\beta\mathcal{R}} u$ *then* $\mathcal{M}_{t,\psi} = \mathcal{M}_{u,\psi}$.

Proof. The term $((\lambda x : C\ t')\ u')$ reduces to $(u'/x)t'$. Let D be the type of t'. Both terms $((\lambda x : C\ t')\ u')$ and $(u'/x)t'$ have type $(u'/x)D$. If $\mathcal{N}_D = \{e\}$, then $\mathcal{M}_{((\lambda x:C\ t')\ u'),\psi} = e = \mathcal{M}_{(u'/x)t',\psi}$, otherwise $\mathcal{M}_{((\lambda x:C\ t')\ u'),\psi} = \mathcal{M}_{t',\psi+x=\mathcal{M}_{u',\psi}} = \mathcal{M}_{(u'/x)t',\psi}$.

We have $\mathcal{M}_{\varepsilon_{Kind}(\dot{Type}),\psi} = \mathcal{M}_{\varepsilon_{Kind},\psi}(\mathcal{M}_{\dot{Type},\psi}) = \mathcal{M}_{\dot{Type},\psi} = \mathcal{B} = \mathcal{M}_{U_{Type},\psi}$.

We have $\mathcal{M}_{\varepsilon_{Kind}(\dot{\Pi}_{\langle Kind,Kind,Kind\rangle}\ C\ D),\psi} = \mathcal{M}_{\dot{\Pi}_{\langle Kind,Kind,Kind\rangle},\psi}\mathcal{M}_{C,\psi}\mathcal{M}_{D,\psi}$ is the set of functions from $\mathcal{M}_{C,\psi}$ to $\mathcal{M}_{D,\psi}e$. And $\mathcal{M}_{\Pi x:(\varepsilon_{Kind}\ C)\ (\varepsilon_{Kind}\ (D\ x)),\psi}$ is the set of functions from $\mathcal{M}_{(\varepsilon_{Kind}\ C),\psi}$ to $\bigcup_{c\in\mathcal{N}_{(\varepsilon_{Kind}\ C)}} \mathcal{M}_{\varepsilon_{Kind}\ (D\ x),\psi+x=c}$, that is the set of functions from $\mathcal{M}_{C,\psi}$ to $\mathcal{M}_{(D\ x),\psi+x=e}$ that is the set of functions from $\mathcal{M}_{C,\psi}$ to $\mathcal{M}_{D,\psi}e$ as well.

We have $\mathcal{M}_{\varepsilon_{Kind}(\dot{\Pi}_{\langle Type,Kind,Kind\rangle}\ C\ D),\psi} = \mathcal{M}_{\dot{\Pi}_{\langle Type,Kind,Kind\rangle},\psi}\mathcal{M}_{C,\psi}\mathcal{M}_{D,\psi}$ is the set of functions from $\{e\}$ to $\mathcal{M}_{D,\psi}e$. And $\mathcal{M}_{\Pi x:(\varepsilon_{Type}\ C)\ (\varepsilon_{Kind}\ (D\ x)),\psi}$ is the set of functions from $\mathcal{M}_{(\varepsilon_{Type}\ C)}$ to $\bigcup_{c\in\mathcal{N}_{(\varepsilon_{Type}\ C)}} \mathcal{M}_{\varepsilon_{Kind}(D\ x),\psi+x=c}$, that is the set of functions from $\{e\}$ to $\mathcal{M}_{(D\ x),\psi+x=e}$ that is the set of functions from $\{e\}$ to $\mathcal{M}_{D,\psi}e$ as well.

We have $\mathcal{M}_{\varepsilon_{Type}(\dot{\Pi}_{\langle s,Type,Type\rangle}\ C\ D)} = \{e\}$ and as $\bigcup_{c\in\mathcal{N}_{(\varepsilon\ C)}} \mathcal{M}_{\varepsilon_{Type}(D\ x),\psi+x=c} = \{e\}$, we have $\mathcal{M}_{\Pi x:(\varepsilon\ C)\ (\varepsilon_{Type}\ (D\ x))} = \{e\}$ as well.

We prove, by induction on t, that if $t \longrightarrow^1_{\beta\mathcal{R}} u$ then $\mathcal{M}_{t,\psi} = \mathcal{M}_{u,\psi}$ and we conclude with a simple induction on the structure of the derivation of $t \equiv_{\beta\mathcal{R}} u$.

From now on, consider a fixed valuation onto $\mathcal{N}$ ψ and write $\mathcal{M}_A$ for $\mathcal{M}_{A,\psi}$.

Definition 5.5 *Let* t *be a term of type* A *in a context* Γ *or that is equal to* $Kind$. *Let* ϕ *be a* Γ*-valuation onto* $\mathcal{M}$. *The element* $[\![t]\!]_\phi$ *of* $\mathcal{M}_A$ *is defined as follows.*

- $[\![Kind]\!]_\phi = [\![Type]\!]_\phi = [\![U_{Kind}]\!]_\phi = [\![U_{Type}]\!]_\phi = [\![\dot{Type}]\!]_\phi = \tilde{T}$,
- $[\![x]\!]_\phi = \phi x$,
- $[\![\lambda x : C\ t]\!]_\phi$ *is the function mapping* c *in* $\mathcal{M}_C$ *to* $[\![t]\!]_{\phi,x=c}$, *except if for all* c *in* $\mathcal{M}_C$ $[\![t]\!]_{\phi,x=c} = e$, *in which case* $[\![\lambda x : C\ t]\!]_\phi = e$, *(note that* $[\![t]\!]_{\phi,x=c}$ *is in* $\mathcal{M}_D$, *that is* $\mathcal{M}_{D,\psi}$, *hence it is in* $\bigcup_{d\in\mathcal{N}_C} \mathcal{M}_{D,\psi+x=d}$*),*
- $[\![(t\ u)]\!]_\phi = [\![t]\!]_\phi([\![u]\!]_\phi)$, *except if* $[\![t]\!]_\phi = e$, *in which case* $[\![(t\ u)]\!]_\phi = e$,
- $[\![\Pi x : C\ D]\!]_\phi = \tilde{\Pi}([\![C]\!]_\phi, \{[\![D]\!]_{\phi,x=c} \mid c \in \mathcal{M}_C\})$,
- $[\![\varepsilon_{Type}]\!]_\phi = [\![\varepsilon_{Kind}]\!]_\phi$ *is the identity on* $\mathcal{B}$,

- $[\![\dot{\Pi}_{\langle Type,Type,Type\rangle}]\!]_\phi = [\![\dot{\Pi}_{\langle Type,Kind,Kind\rangle}]\!]_\phi = [\![\dot{\Pi}_{\langle Kind,Type,Type\rangle}]\!]_\phi = [\![\dot{\Pi}_{\langle Kind,Kind,Kind\rangle}]\!]_\phi$ *is the function mapping an element C in $\mathcal{B}$ and a function f from some set S in E to $\mathcal{B}$ to the element $\tilde{\Pi}(C, \{f\,s \mid s \in S\})$ of $\mathcal{B}$.*

Lemma 5.11 (Substitution) $[\![(u/x)t]\!]_\phi = [\![t]\!]_{\phi+x=[\![u]\!]_\phi}$

Proof. By induction over the structure of t.

Lemma 5.12 (Conversion) *If $t \equiv_{\beta\mathcal{R}} u$ then $[\![t]\!]_\phi = [\![u]\!]_\phi$.*

Proof. If $t = ((\lambda x : B\ t')\ u')$, then let C be the type of t', if $\mathcal{M}_C = \{e\}$ then $[\![((\lambda x : B\ t')\ u')]\!]_\phi = e = [\![(u'/x)t']\!]_\phi$, otherwise $[\![((\lambda x : B\ t')\ u')]\!]_\phi = [\![t']\!]_{\phi,x=[\![u']\!]_\phi} = [\![(u'/x)t']\!]_\phi$.

We have $[\![\varepsilon_{Kind}(\dot{Type})]\!]_\phi = \tilde{T} = [\![U_{Type}]\!]_\phi$.

If $s_1 = Kind$, then $[\![\varepsilon_{s_2}(\dot{\Pi}_{\langle s_1,s_2,s_2\rangle}\ C\ D)]\!]_\phi = \tilde{\Pi}([\![C]\!]_\phi, \{[\![D]\!]_\phi c \mid c \in \mathcal{M}_C\}) = [\![\Pi y : (\varepsilon_{s_1}\ C)\ (\varepsilon_{s_2}\ (D\ y))]\!]_\phi$.

If $s_1 = Type$, then $[\![\varepsilon_{s_2}(\dot{\Pi}_{\langle s_1,s_2,s_2\rangle}\ C\ D)]\!]_\phi = \tilde{\Pi}([\![C]\!]_\phi, \{[\![D]\!]_\phi e\}) = [\![\Pi y : (\varepsilon_{s_1}\ C)\ (\varepsilon_{s_2}\ (D\ y))]\!]_\phi$.

We prove, by induction on t, that if $t \longrightarrow^1_{\beta\mathcal{R}} u$ then $[\![t]\!]_\phi = [\![u]\!]_\phi$ and we conclude with a simple induction on the structure of the derivation of $t \equiv_{\beta\mathcal{R}} u$.

6 The termination of β-reduction in super-consistent theories

We now prove that proof-reduction terminates in the $\lambda\Pi$-calculus modulo a super-consistent theory such as Simple type theory or the Calculus of constructions. We use here the notion of *reducibility candidate* introduced by Girard [8]. Our definition, however, follows that of Parigot [12].

6.1 The candidates

Definition 6.1 (Operations on set of terms) *The set $\tilde{T}$ is defined as the set of strongly terminating terms.*

Let C be a set of terms and S be a set of sets of terms. The set $\tilde{\Pi}(C, S)$ is defined as the set of strongly terminating terms t such that if $t \longrightarrow^*_\beta \lambda x : A\ t'$ then for all t'' in C, and for all D in S, $(t''/x)t' \in D$.

The main property of the operation $\tilde{\Pi}$ is expressed by the following Lemma.

Lemma 6.1 *Let C be a set of terms and S be a set of sets of terms, t_1, t_2, and u be terms such that $t_1 \in \tilde{\Pi}(C, S)$, $t_2 \in C$, and $(t_1\ t_2) \longrightarrow^1_\beta u$, n_1 and n_2 be natural numbers such that n_1 is the maximum length of a reduction sequence issued from t_1, and t_2 is the maximum length of a reduction sequence issued from t_2, and D be an element of S. Then, $u \in D$.*

Proof. By induction on $n_1 + n_2$. If the reduction is at the root of the term, then t_1 has the form $\lambda x : A\ t'$ and $u = (t_2/x)t'$. By the definition of $\tilde{\Pi}(C, S)$, $u \in D$. Otherwise, the reduction takes place in t_1 or in t_2, and we we apply the induction hypothesis.

Definition 6.2 (Candidates) Candidates *are inductively defined by the three rules*

- *the set $\tilde{T}$ of all strongly terminating terms is a candidate,*

- *if C is a candidate and S is a set of candidates, then $\tilde{\Pi}(C, S)$ is a candidate,*
- *if S is a set of candidates, then $\bigcap S$ is a candidate.*

We write $\mathcal{C}$ for the set of all candidates.

The Π-algebra $\langle \mathcal{C}, \tilde{T}, \mathcal{P}(\mathcal{C}), \tilde{\Pi} \rangle$ is full, it is ordered by the subset relation and complete for this order.

Lemma 6.2 (Termination) *If C is a candidate, then all the elements of C strongly terminate.*

Proof. By induction on the construction of C.

Lemma 6.3 (Variables) *If C is a candidate and x is a variable, then $x \in C$.*

Proof. By induction on the construction of C.

Lemma 6.4 (Closure by reduction) *If C is a candidate, $t \in C$, and $t \longrightarrow_\beta^* t'$, then $t' \in C$.*

Proof. By induction on the construction of C.

If $C = \tilde{T}$, then as t is an element of C, it strongly terminates, thus t' strongly terminates, and $t' \in C$.

If $C = \tilde{\Pi}(D, S)$, then as t is an element of C, it strongly terminates, thus t' strongly terminates. If moreover $t' \longrightarrow_\beta^* \lambda x : A\ t_1$, then $t \longrightarrow_\beta^* \lambda x : A\ t_1$, and for all u in D, and for all E in S, $(u/x)t_1 \in E$. Thus, $t' \in C$.

If $C = \bigcap_i C_i$, then for all i, $t \in C_i$ and by induction hypothesis $t' \in C_i$. Thus, $t' \in C$.

Lemma 6.5 (Applications) *Let C be a candidate and S be a set of candidates, t_1 and t_2 such that $t_1 \in \tilde{\Pi}(C, S)$ and $t_2 \in C$, and D be an element of S. Then $(t_1\ t_2) \in D$.*

Proof. As $t_1 \in \tilde{\Pi}(C, S)$ and and $t_2 \in C$, t_1 and t_2 strongly terminate. Let n_1 be the maximum length of a reduction sequence issued from t_1 and n_2 be the maximum length of a reduction sequence issued from t_2. By Lemma 6.1, all the one step reducts of $(t_1\ t_2)$ are in D.

To conclude that $(t_1\ t_2)$ itself is in D, we prove, by induction on the construction of D, that if D is a candidate and all the one-step reducts of the term $(t_1\ t_2)$ are in D, then $(t_1\ t_2)$ is in D.

- If $D = \tilde{T}$, then as all the one-step reducts of the term $(t_1\ t_2)$ strongly terminate, the term $(t_1\ t_2)$ strongly terminates, and $(t_1\ t_2) \in D$.
- If $D = \tilde{\Pi}(C, S)$, then as all the one-step reducts of the term $(t_1\ t_2)$ strongly terminate, the term $(t_1\ t_2)$ strongly terminates. If moreover $(t_1\ t_2) \longrightarrow_\beta^* \lambda x : A\ v$, then let $(t_1\ t_2) = u_1, u_2, \ldots, u_n = \lambda x : A\ v$ be a reduction sequence from $(t_1\ t_2)$ to $\lambda x : A\ v$. As $(t_1\ t_2)$ is an application and $\lambda x : A\ v$ is not, $n \geq 2$. Thus, $(t_1\ t_2) \longrightarrow_\beta^1 u_2 \longrightarrow_\beta^* \lambda x : A\ v$. We have $u_2 \in D$ and $u_2 \longrightarrow_\beta^* \lambda x : A\ v$, thus for all w in C and F in S, $(w/x)v \in F$. Thus, $(t_1\ t_2) \in \tilde{\Pi}(C, S) = D$.
- If $D = \bigcap_i D_i$, then for all i, all the one step reducts of $(t_1\ t_2)$ are in D_i, and, by induction hypothesis $(t_1\ t_2) \in D_i$. Thus, $(t_1\ t_2) \in D$.

6.2 Termination

Consider a super-consistent theory $\Sigma, \mathcal{R}$. We want to prove that β-reduction terminates in the $\lambda\Pi$-calculus modulo this theory, while $\beta\mathcal{R}$-reduction may or may not terminate. As this theory is super-consistent, it has a model $\mathcal{M}$ valued in the Π-algebra $\langle \mathcal{C}, \tilde{T}, \mathcal{P}(\mathcal{C}), \tilde{\Pi} \rangle$. Consider this model.

If a term t has type B in some context Γ, then B has type $Type$ in Γ, B has type $Kind$ in Γ, or $B = Kind$. Thus, $[\![B]\!]_\phi$ is an element of $\mathcal{M}_{Type} = \mathcal{C}$, $[\![B]\!]_\phi$ is an element of $\mathcal{M}_{Kind} = \mathcal{C}$, or $[\![B]\!]_\phi = \tilde{T}$. In all these cases $[\![B]\!]_\phi$ is a candidate.

Lemma 6.6 *Let $\Gamma = x_1 : A_1, ..., x_n : A_n$ be a context, ϕ be a Γ-valuation onto $\mathcal{M}$, σ be a substitution mapping every x_i to an element of $[\![A_i]\!]_\phi$ and t a term of type B in Γ. Then $\sigma t \in [\![B]\!]_\phi$.*

Proof. By induction on the structure of t.

- If $t = Type$, then $B = Kind$, $[\![B]\!]_\phi = \tilde{T}$ and $\sigma t = Type \in [\![B]\!]_\phi$.
- If $t = x$ is a variable, then by definition of σ, $\sigma t \in [\![A]\!]_\phi$.
- If $t = \Pi x : C\ D$, then $B = Type$ or $B = Kind$, and $[\![B]\!]_\phi = \tilde{T}$, $\Gamma \vdash C : Type$ and $\Gamma, x : C \vdash D : Type$ or $\Gamma, x : C \vdash D : Kind$, by induction hypothesis $\sigma C \in [\![Type]\!]_\phi = \tilde{T}$, that is σC strongly terminates and $\sigma D \in [\![Type]\!]_\phi = \tilde{T}$ or $\sigma D \in [\![Kind]\!]_\phi = \tilde{T}$, that is σD strongly terminates. Thus, $\sigma(\Pi x : C\ D) = \Pi x : \sigma C\ \sigma D$ strongly terminates also and it is an element of $\tilde{T} = [\![B]\!]_\phi$.
- If $t = \lambda x : C\ u$ where u has type D. Then $B = \Pi x : C\ D$ and $[\![B]\!]_\phi = [\![\Pi x : C\ D]\!]_\phi = \tilde{\Pi}([\![C]\!]_\phi, \{[\![D]\!]_{\phi, x=c} \mid c \in \mathcal{M}_C\})$ is the set of terms s such that s strongly terminates and if s reduces to $\lambda x : E\ s_1$ then for all s' in $[\![C]\!]_\phi$ and all a in $\mathcal{M}_C$, $(s'/x)s_1$ is an element of $[\![D]\!]_{\phi, x=a}$.

  We have $\sigma t = \lambda x : \sigma C\ \sigma u$, consider a reduction sequence issued from this term. This sequence can only reduce the terms σC and σu. By induction hypothesis, the term σC is an element of $[\![Type]\!]_\phi = \tilde{T}$ and the term σu is an element of $[\![D]\!]_\phi$, thus the reduction sequence is finite.

  Furthermore, every reduct of σt has the form $\lambda x : C'\ v$ where C' is a reduct of σC and v is a reduct of σu. Let w be any term of $[\![C]\!]_\phi$, and a be any element of $\mathcal{M}_C$, the term $(w/x)v$ can be obtained by reduction from $((w/x) \circ \sigma)u$. By induction hypothesis, the term $((w/x) \circ \sigma)u$ is an element of $[\![D]\!]_{\phi, x=a}$. Hence, by Lemma 6.4 the term $(w/x)v$ is an element of $[\![D]\!]_{\phi, x=a}$. Therefore, the term $\sigma\lambda x\ u$ is an element of $[\![B]\!]_\phi$.
- If the term t has the form $(u_1\ u_2)$ then u_1 is a term of type $\Pi x : C\ D$, u_2 a term of type C and $B = (u_2/x)D$. We have $\sigma t = (\sigma u_1\ \sigma u_2)$, and by induction hypothesis $\sigma u_1 \in [\![\Pi x : C\ D]\!]_\phi = \tilde{\Pi}([\![C]\!]_\phi, \{[\![D]\!]_{\phi + x = a} \mid a \in \mathcal{M}_A\})$ and $\sigma u_2 \in [\![C]\!]_\phi$. By Lemma 6.5, $(\sigma u_1\ \sigma u_2) \in [\![D]\!]_{\phi + x = [\![u_2]\!]_\phi} = [\![(u_2/x)D]\!]_\phi = [\![B]\!]_\phi$.

Theorem 6.1 *Let Γ be a context and t be a term well-typed in Γ. Then t strongly terminates.*

Proof. Let B be the type of t in Γ, let ϕ be any Γ-valuation onto $\mathcal{M}$, σ be the substitution mapping every x_i to itself. Note that, by Lemma 6.3, this variable is an element of $[\![A_i]\!]_\phi$. Then $t = \sigma t \in [\![B]\!]_\phi$. Hence it strongly terminates.

7 Consistency of the $\lambda\Pi$-calculus modulo Simple type theory and modulo the Calculus of constructions

Lemma 7.1 (Consistency of the $\lambda\Pi$-calculus modulo Simple type theory) *In the context $x : o$ there is no term of type $\varepsilon(x)$.*

Proof. Assume there exists a term t of type $\varepsilon(x)$ in the context $x : o$ and let t' be its β-normal form. The term t' would have the form $(h\ u_1\ ...\ u_n)$ for some constant or variable h. A case analysis shows that no constant or variable can yield a term of type $\varepsilon(x)$.

A similar argument applies to the $\lambda\Pi$-calculus modulo the Calculus of constructions with the context $x : U_{Type}$ and the type $\varepsilon_{Type}(x)$.

8 Termination of the $\beta\mathcal{R}$-reduction

We finally prove the termination of the $\beta\mathcal{R}$-reduction for Simple type theory and for the Calculus of constructions. The rules $\mathcal{R}$ of Simple type theory are

$$\varepsilon(\dot{\Rightarrow}\ X\ Y) \longrightarrow \varepsilon(X) \to \varepsilon(Y)$$

$$\varepsilon(\dot{\forall}_A\ X) \longrightarrow \Pi z : A\ \varepsilon(X\ z)$$

This set $\mathcal{R}$ of rewrite rules terminates, as each reduction step reduces the number of symbols $\dot{\Rightarrow}$ and $\dot{\forall}_A$ in the term. Then, $\mathcal{R}$-reduction can create β-redices, but only β-redices on the form $((\lambda x : A\ t)\ z)$ where z is a variable. Thus, any term can be (weakly) $\beta\mathcal{R}$-reduced by β-reducing it first, then $\mathcal{R}$-reducing it, then β-reducing the trivial β-redices created by the $\mathcal{R}$-reduction.

A similar argument applies to the Calculus of constructions.

References

[1] A. Brunel, O. Hermant, and C. Houtmann, Orthogonality and Boolean Algebras for Deduction Modulo, *Typed Lambda Calculus and Applications* Lecture Notes in Computer Science 6990, 2011, pp. 76-90.

[2] T. Coquand and G. Huet, The Calculus of Constructions, *Information and Computation*, 76, 1988, pp. 95-120.

[3] D. Cousineau and G. Dowek, Embedding Pure Type Systems in the lambda-Pi-calculus modulo, S. Ronchi Della Rocca, *Typed lambda calculi and applications*, Lecture Notes in Computer Science 4583, Springer-Verlag, 2007, pp. 102-117.

[4] G. Dowek, Truth values algebras and proof normalization, Th. Altenkirch and C. McBride, *Types for proofs and programs*, Lecture Notes in Computer Science 4502, 2007, pp. 110-124.

[5] G. Dowek, Th. Hardin, and C. Kirchner, Theorem proving modulo, *Journal of Automated Reasoning*, 31, 2003, pp. 33-72.

[6] G. Dowek, Th. Hardin, and C. Kirchner, HOL-lambda-sigma: an intentional first-order expression of higher-order logic, *Mathematical Structures in Computer Science*, 11, 2001, pp. 1-25.

[7] G. Dowek and B. Werner, Proof normalization modulo, *The Journal of Symbolic Logic*, 68, 4, 2003, pp. 1289-1316.

[8] J.Y. Girard, Interprétation Fonctionnelle et Élimination des Coupures dans l'Arithmétique d'Ordre Supérieur, *Thèse de Doctorat*, Université Paris VII, 1972.

[9] R. Harper, F. Honsell, and G. Plotkin, A framework for defining logics, *Journal of the ACM*, 40, 1, 1993, pp. 143-184.

[10] P. Martin-Löf, Intuitionistic Type Theory, *Bibliopolis*, 1984.

[11] B. Nordström, K. Petersson, and J.M. Smith, Martin-Löf's type theory. *Handbook of Logic in Computer Science*, S. Abramsky, D. Gabbay, and T. Maibaum (eds.), Clarendon Press, 2000, pp. 1-37.

[12] M. Parigot, Strong normalization for the second orclassical natural de duction. *Logic in Computer Science*, 1993, pp. 39âĂŞ46.

[13] R. Saillard, Towards explicit rewrite rules in the ÎźÎă-calculus modulo, International Workshop on the Implementation of Logics, 2013.