Models and termination of proof reduction in the λΠ-calculus modulo theory

Gilles Dowek*

Abstract

We define a notion of model for the λΠ-calculus modulo theory and prove a soundness theorem. We then use this notion to define a notion of super-consistent theory and prove that proof reduction terminates in the λΠ-calculus modulo any super-consistent theory. We prove this way the termination of proof reduction in several theories including Simple type theory and the Calculus of constructions.

1 Introduction

1.1 Models and termination

In Predicate logic and in Deduction modulo theory [9, 10], a model is defined by a domain \( \mathcal{M} \), a set \( B \) of truth values, and an interpretation function, parametrized by a valuation \( \phi \), mapping each term \( t \) to an element \( \llbracket t \rrbracket_\phi \) of \( \mathcal{M} \), and each proposition \( A \) to an element \( \llbracket A \rrbracket_\phi \) of \( B \).

In the usual definition of the notion of model, the set \( B \) is a two-element set \( \{0, 1\} \), but this notion can be extended to a notion of many-valued model, where \( B \) is an arbitrary Boolean algebra, a Heyting algebra, a pre-Boolean algebra [4], or a pre-Heyting algebra [7]. Boolean algebras permit to introduce intermediate truth values for propositions that are neither provable nor disprovable, Heyting algebras permit to construct models of constructive Predicate logic, where the excluded middle is not necessarily valid, and pre-Boolean and pre-Heyting algebras, where the order relation \( \leq \) is replaced by a pre-order relation, permit to distinguish a notion of weak equivalence: \( \llbracket A \rrbracket_\phi \leq \llbracket B \rrbracket_\phi \) and \( \llbracket B \rrbracket_\phi \leq \llbracket A \rrbracket_\phi \), for all \( \phi \), from a notion of strong equivalence: \( \llbracket A \rrbracket_\phi = \llbracket B \rrbracket_\phi \), for all \( \phi \). The first corresponds to the provability of \( A \iff B \) and the second to the congruence defining the computational equality in Deduction modulo theory [9, 10], also known as definitional equality in Constructive type theory [14].

In a model valued in a Boolean algebra, a Heyting algebra, a pre-Boolean algebra, or a pre-Heyting algebra, a proposition \( A \) is said to be valid when it is weakly equivalent to the proposition \( \top \), that is when for all \( \phi \), \( \llbracket A \rrbracket_\phi \geq \top \), and this condition can be rephrased as \( \llbracket A \rrbracket_\phi = \top \) in Boolean and Heyting algebras. A congruence \( \equiv \) defined on propositions is said to be valid when for all \( A \) and \( B \) such that \( A \equiv B \), \( A \) and \( B \) are strongly equivalent, that is \( \llbracket A \rrbracket_\phi = \llbracket B \rrbracket_\phi \), for all \( \phi \). 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.

Proof reduction terminates in Deduction modulo a theory defined by a set of axioms \( \mathcal{T} \) and a congruence \( \equiv \), when this theory has a model valued in the pre-Heyting algebra of reducibility candidates [10, 7]. 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 \( \mathcal{B} \) [7]. This theorem permits to completely separate the semantic and the syntactic aspects that are often mixed in the usual proofs of termination of proof reduction. The semantic aspect is in the proof of super-consistency of the theory and the syntactic in the universal proof that super-consistency implies termination of proof reduction.

For the termination of proof reduction, the congruence matters, but the axioms do not. Thus, the pre-order relation \( \leq \) does not matter in the algebra of reducibility candidates and it is possible

*Inria and École Normale Supérieure de Paris-Saclay, 61, avenue du Président Wilson, 94235 Cachan Cedex, France, gilles.dowek@ens-paris-saclay.fr
to define it as the trivial pre-order relation such that \( C \leq C' \), for all \( C \) and \( C' \). Such a pre-Heyting algebra is said to be trivial. As the pre-order is trivial, all the conditions defining pre-Heyting algebras, such as \( a \land b \leq a, a \land b \leq b \ldots \) are always satisfied in a trivial pre-Heyting algebra, and a trivial pre-Heyting algebra is just a set equipped with arbitrary operations \( \land, \Rightarrow \ldots \). Thus, in order to prove that proof reduction terminates in Deduction modulo a theory defined by a set of axioms \( T \) and a congruence \( \equiv \), it is sufficient to prove that for all trivial pre-Heyting algebras \( B \), the theory has a model valued in \( B \).

1.2 The \( \lambda \Pi \)-calculus modulo theory

In Predicate logic and in Deduction modulo theory, terms, propositions, and proofs belong to three distinct languages. But, it more thrifty to consider a single language, such as the \( \lambda \Pi \)-calculus modulo theory [6], which is implemented in the DEDUKTI system [1], or Martin-Löf’s Logical Framework [17], and express terms, propositions, and proofs, in this language.

For instance, in Predicate logic, 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 \( \text{nat} \), \( P(0) \Rightarrow P(0) \) has type \( \text{Type} \) and \( \lambda \alpha : P(0) \alpha \) has type \( P(0) \Rightarrow P(0) \).

1.3 From pre-Heyting algebras to \( \Pi \)-algebras

The first goal of this paper is to extend the notion of pre-Heyting algebra to a notion of \( \Pi \)-algebra, adapted to the \( \lambda \Pi \)-calculus modulo theory.

In Predicate logic and in Deduction modulo theory, the propositions are built from atomic propositions with the connectors and quantifiers \( \top, \bot, \land, \lor, \Rightarrow, \forall, \exists \). Accordingly, the operations of a pre-Heyting algebra are \( \top, \bot, \land, \lor, \Rightarrow, \forall, \exists \). In the \( \lambda \Pi \)-calculus and in the \( \lambda \Pi \)-calculus modulo theory, the only connector is \( \Pi \). Thus, a \( \Pi \)-algebra mainly has an operation \( \Pi \). As expected, its properties are a mixture of the properties of the implication and of the universal quantifier of the pre-Heyting algebras.

1.4 Layered models

The second goal of this paper is to extend the usual notion of model to the \( \lambda \Pi \)-calculus modulo theory.

Extending the notion of model to many-sorted predicate logic requires to consider not just one domain \( M \), but a family of domains \( M_s \), indexed by the sorts. For instance, in a model of Simple type theory, the family of domains is indexed by simple types. 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 domains \( (M_s)_s \) indexed by the terms of the calculus and a function \( [\cdot] \) mapping each term \( t \) of type \( A \) and valuation \( \phi \) to an element \( [t]_\phi \) of \( M_A \).

The functions \( M \) and \( [\cdot] \) are similar, in the sense that both their domains is the set of terms of the calculus. The goal of the model construction is to define the function \( [\cdot] \) and the function \( M \) can be seen as a tool helping to define this function. For instance, assume \( f \) is a symbol of type \( A \rightarrow A \), where \( A \) is a term of type \( Type \), and we want to construct a model of the rule

\[
 f(x) \rightarrow x
\]

In this model, we want to define the interpretation \( [f] \) as the identity function over some set, but to state which, we must first define the function \( M \) that maps the term \( A \) to a set \( M_A \), and then define \( [f]_\phi \) as the identity function over the set \( M_A \).

In Predicate logic, terms and propositions may be typed with sorts, but sorts themselves have no type. In the \( \lambda \Pi \)-calculus modulo theory, in contrast, terms have types that have types, that have... This explains that, in some cases, constructing the function \( M \) itself requires to define first another function \( N \), that is used as a tool helping to define this function. This can be iterated to a several layer model, where the function \( [\cdot] \) is defined with the help of a function \( M \), that is defined with the help of a function \( N \), that is defined with the help of... Such layered constructions are common in proofs of termination of proof reduction [11, 15, 3].
Note that, in this definition of the notion of model, when a term \( t \) has type \( A \), we do not require \( [t]_\phi \) to be an element of \( [A]_\phi \), but of \( \mathcal{M}_A \). This is consistent with the notion of model of many-sorted predicate logic, where we require \( [\ell]_\phi \) to be an element of \( \mathcal{M}_s \) and where \( [s]_\phi \) is often not even defined.

Valuations must be handled with care in such layered models. In a three layer model, for instance, the definition of \( \mathcal{N}_f \) is absolute, the definition of \( \mathcal{M}_s \) is relative to a valuation \( \psi \), mapping each variable of type \( A \) to an element of \( \mathcal{N}_A \), and the definition of \( [t]_\phi \) is relative to a valuation \( \psi \) and to a valuation \( \phi \) mapping each variable of type \( A \) to an element of \( \mathcal{M}_A, \psi \).

### 1.5 Super-consistency and proof reduction

The third goal of this paper is to use this notion of \( \Pi \)-algebra to define a notion of super-consistency and to prove that proof reduction terminates in the \( \lambda \Pi \)-calculus modulo any super-consistent theory.

We prove this way the termination of proof reduction in several theories expressed in the \( \lambda \Pi \)-calculus modulo theory, including Simple type theory [9] and the Calculus of constructions [6]. Together with confluence, this termination of proof reduction is a property required to define these theories in the system Dedukti [1].

In Section 2, we recall the definition of the \( \lambda \Pi \)-calculus modulo theory and give three examples of theories expressed in this framework. In Section 3, we introduce the notion of \( \Pi \)-algebra and that of model for the \( \lambda \Pi \)-calculus modulo theory and we prove a soundness theorem. In Section 4, we define the notion of super-consistent theory and prove that the three theories introduced in Section 2 are super-consistent. In Section 5, we prove that proof reduction terminates in the \( \lambda \Pi \)-calculus modulo any super-consistent theory.

### 2 The \( \lambda \Pi \)-calculus modulo theory

#### 2.1 The \( \lambda \Pi \)-calculus

The syntax of the \( \lambda \Pi \)-calculus is

\[
t = x \mid Type \mid Kind \mid \Pi x : t t \mid \lambda x : t t \mid t t
\]

and the typing rules are given in Figure 1.

As usual, we write \( A \rightarrow B \) for \( \Pi x : A B \) when \( x \) does not occur in \( B \). The \( \alpha \)-equivalence relation is defined as usual and terms are identified modulo \( \alpha \)-equivalence. The relation \( \beta \)—one step \( \beta \)-reduction at the root—is defined as usual. If \( r \) is a relation on terms, we write \( \longrightarrow^1 \), \( \longrightarrow^+ \) for the monotonic closure of \( \longrightarrow \), \( \longrightarrow^+ \) for the transitive closure of \( \longrightarrow^1 \), \( \longrightarrow^* \) for its reflexive-transitive closure, and \( \Xi \), for its reflexive-symmetric-transitive closure.

If \( \Sigma, \Gamma, \) and \( \Delta \) are contexts, a substitution \( \theta \), binding the variables of \( \Gamma \), is said to have type \( \Gamma \overset{\sim}{\rightarrow} \Delta \) in \( \Sigma \) if for all \( x : A \) in \( \Gamma \), we have \( \Sigma, \Delta \vdash \theta x : \theta A \). In this case, if \( \Sigma, \Gamma \vdash t : B \), then \( \Sigma, \Delta \vdash \theta t : \theta B \).

Types are preserved by \( \beta \)-reduction. The \( \beta \)-reduction relation is confluent and strongly terminating. And each term has a unique type modulo \( \beta \)-equivalence [13].

A term \( t \), well-typed in some context \( \Gamma \) is said to be a kind if its type in this context is Kind. For instance, \( Type \) and \( nat \rightarrow Type \) are kinds. It is said to be a type family if its type is a kind. In particular, it is said to be a type if its type is \( Type \). For instance, \( nat \), \( array \), and \( (array \ 0) \) are type families, among which \( nat \) and \( (array \ 0) \) are types. It is is said to be an object if its type is a type. For instance, \( 0 \) and \( [0] \) are objects.

#### 2.2 The \( \lambda \Pi \)-calculus modulo theory

**Definition 2.1 (Rewrite rule)** A rewrite rule is a triple \( \longrightarrow^\Gamma \) where \( \Gamma \) is a context and \( l \) and \( r \) are \( \beta \)-normal terms. Such a rule is said to be well-typed in the context \( \Sigma \) if, in the \( \lambda \Pi \)-calculus, the context \( \Sigma, \Gamma \) is well-formed and there exists a term \( A \) such that the terms \( l \) and \( r \) both have type \( A \) in this context.
If $\Sigma$ is a context, $l \rightarrow^\beta r$ is a rewrite rule well-typed in $\Sigma$ and $\theta$ is a substitution of type $\Gamma \sim \Delta$ in $\Sigma$, then the terms $\theta l$ and $\theta r$ both have type $\theta A$ in the context $\Sigma, \Delta$.

The relation $R$—one step $R$-reduction at the root—is defined by: $t R u$ there exists a rewrite rule $l \rightarrow^R r$ and a substitution $\theta$ such that $t = \theta l$ and $u = \theta r$. The relation $\beta R$—one step $\beta R$-reduction at the root—is the union of $\beta$ and $R$.

**Definition 2.2 (Theory)** A theory is a pair formed with a context $\Sigma$, well-formed in the $\lambda\Pi$-calculus, and a set of rewrite rules $R$, well-typed in $\Sigma$ in the $\lambda\Pi$-calculus.

The variables declared in $\Sigma$ are called constants. They replace the sorts, the function symbols, the predicate symbols, and the axioms of a theory in Predicate logic.

**Definition 2.3 (The $\lambda\Pi$-calculus modulo theory)** The $\lambda\Pi$-calculus modulo $\Sigma, R$ is the extension of the $\lambda\Pi$-calculus obtained modifying the Declaration rules to replace the condition $x \notin \Sigma, \Gamma$, the Variable rules to replace the condition $x : A \in \Gamma$ by $x : A \in \Sigma, \Gamma$, and the Conversion rules to replace the condition $A \equiv_B B$ with $A \equiv_{\beta R} B$.

In all this paper, we assume that the relation $\rightarrow^{\parallel}_{\beta R}$ is confluent. This way, $A \equiv_{\beta R} B$ if and only if there exists a $C$ such that $A \rightarrow_{*\beta R}^{\parallel} C$ and $B \rightarrow_{\beta R}^{*} C$. Confluence and subject reduction are indeed needed to build models and prove termination of proof reduction. This is consistent with the methodology defined in [2]: first prove confluence, then subject reduction, then termination. With the strong restriction we have on the type of rules, the subject reduction property is trivial.

**2.3 Examples of theories**

Simple type theory can be expressed in Deduction modulo theory [8]. The main idea in this presentation is to distinguish terms of type $o$ from propositions. If $t$ is a term of type $o$, the corresponding proposition is written $\varepsilon(t)$. The term $t$ is a propositional content or a code of the proposition $\varepsilon(t)$. This way, it is not possible to quantify over propositions, but it is possible to quantify over codes of propositions: there is no proposition

$$\forall X \ (X \Rightarrow X)$$
\[
\begin{array}{ll}
\iota & : \text{Type} \\
o & : \text{Type} \\
\Rightarrow & : o \rightarrow o \rightarrow o \\
\forall_A & : (A \rightarrow o) \rightarrow o \\
\varepsilon & : o \rightarrow \text{Type} \\
(\varepsilon (\Rightarrow x y)) & \rightarrow (\varepsilon x) \rightarrow (\varepsilon y) \\
(\varepsilon (\forall_A x)) & \rightarrow \Pi z : A \rightarrow (\varepsilon (x z)) \\
\end{array}
\]

with a finite number of quantifiers \( \forall_A \)

Figure 2: Simple type theory

but there is a proposition

\[ \forall x (\varepsilon(x) \Rightarrow \varepsilon(x)) \]

respecting the syntax of Predicate logic, where the predicate symbol \( \varepsilon \) is applied to the variable \( x \) to form a proposition.

In this presentation, each simple type is a sort and, for each simple type \( A \), there is a quantifier \( \forall_A \). Thus, the language contains an infinite number of sorts and an infinite number of constants.

This presentation can be adapted to the \( \lambda \Pi \)-calculus modulo theory. To avoid declaring an infinite number of constants for simple types, we can just declare two constants \( \iota \) and \( o \) of type \( \text{Type} \) and use the product of the \( \lambda \Pi \)-calculus modulo theory to represent the simple types \( \iota \rightarrow \iota \), \( \iota \rightarrow \iota \rightarrow \iota \), \( \iota \rightarrow o \) ... We should declare an infinite number of quantifiers \( \forall_A \), indexed by simple types, but this can be avoided as, in each specific proof, only a finite number of such quantifiers occur. This leads to the theory presented in Figure 2.

Another possibility is to add the type \( A \) as an extra argument of the universal quantifier \( \forall \). To do so, we need to introduce a type \( \text{type} \) for codes of simple types, two constants \( \iota \) and \( o \), of type \( \text{type} \), and not \( \text{Type} \), a constant \( \text{arrow} \) of type \( \text{type} \rightarrow \text{type} \rightarrow \text{type} \), and a decoding function \( \eta \) of type \( \text{type} \rightarrow \text{Type} \) to embed \( \text{type} \) into \( \text{Type} \). This way, the universal quantifier can be given the type \( \Pi a : \text{type} (((\eta a) \rightarrow (\eta o)) \rightarrow (\eta o)) \). This leads to the theory presented in Figure 3.

The Calculus of constructions [5] can also be expressed in the \( \lambda \Pi \)-calculus modulo theory [6] as the theory presented in Figure 4.

Note that this presentation slightly differs from that of [6]: the symbol \( U_{\text{Type}} \) has been replaced everywhere by \( \varepsilon_{\text{Kind}}(T_{\text{Type}}) \) allowing to drop the rule

\[ \varepsilon_{\text{Kind}}(T_{\text{Type}}) \rightarrow U_{\text{Type}} \]

Then, to keep the notations similar to those of Simple type theory, we write \( \text{type} \) for \( U_{\text{Kind}} \), \( o \) for \( T_{\text{Type}} \), \( \eta \) for \( \varepsilon_{\text{Kind}} \), and \( \varepsilon \) for \( \varepsilon_{\text{Type}} \). We also write \( \Pi_{KK} \) for \( \Pi_{(\text{Kind},\text{Kind},\text{Kind})} \), \( \Pi_{TT} \) for \( \Pi_{(\text{Type},\text{Type},\text{Type})} \), \( \Pi_{KT} \) for \( \Pi_{(\text{Kind},\text{Type},\text{Type})} \), and \( \Pi_{TK} \) for \( \Pi_{(\text{Type},\text{Kind},\text{Kind})} \).

Note that the symbol \( \Pi_{KT} \) is exactly the parametric universal quantifier of Simple type theory, the symbol \( \Pi_{TT} \) is a dependent version of the symbol \( \Rightarrow \) and \( \Pi_{KK} \) a dependent version of the symbol \( \Rightarrow \). The symbol \( \Pi_{TK} \), in contrast, is new.

3 Algebras and Models

3.1 \( \Pi \)-algebras

The notion of \( \Pi \)-algebra is an adaptation to the \( \lambda \Pi \)-calculus of the notion of pre-Heyting algebra.

Definition 3.1 (\( \Pi \)-algebra) A \( \Pi \)-algebra is formed with

- a set \( B \),
- a pre-order relation \( \leq \) on \( B \),

5
\begin{align*}
type & : Type \\
\iota & : type \\
o & : type \\
arrow & : type \to type \to type \\
\eta & : type \to Type \\
\Rightarrow & : (\eta o) \to (\eta o) \to (\eta o) \\
\forall & : \Pi a : type \ (((\eta a) \to (\eta o)) \to (\eta o)) \\
\varepsilon & : (\eta o) \to Type \\
(\eta (arrow x y)) & \to (\eta x) \to (\eta y) \\
(\varepsilon (\Rightarrow x y)) & \to (\varepsilon x) \to (\varepsilon y) \\
(\varepsilon (\forall x y)) & \to \Pi z : (\eta x) (\varepsilon (y z))
\end{align*}

Figure 3: Simple type theory with a parametric quantifier

- an element $\top$ of $B$,
- a function $\wedge$ from $B \times B$ to $B$,
- a subset $A$ of $\mathcal{P}^+(B)$, the set of non-empty subsets of $B$,
- a function $\Pi$ from $B \times A$ to $B$,

such that

- $\top$ is a maximal element for $\leq$, that is for all $a$ in $B$, $a \leq \top$,
- $a \wedge b$ is a greatest lower bound of $\{a, b\}$ for $\leq$, that is $a \wedge b \leq a$, $a \wedge b \leq b$, and for all $c$, if $c \leq a$ and $c \leq b$, then $c \leq a \wedge b$,
- $a \leq \Pi(b, S)$ if and only if for all $c$ in $S$, $a \wedge b \leq c$.

Definition 3.2 (Full $\Pi$-algebra) A $\Pi$-algebra is full if $A = \mathcal{P}^+(B)$, that is if $\Pi$ is total on $B \times \mathcal{P}^+(B)$.

Example. The algebra $\langle \{0, 1\}, 1, \wedge, \Pi^+\{\{0, 1\}\}, \Pi\rangle$, where $\wedge$ and $\Pi$ are defined by the tables below, is a $\Pi$-algebra.

\begin{center}
\begin{tabular}{c|c|c}
& 0 & 1 \\
\hline
0 & 0 & 0 \\
1 & 0 & 1 \\
\end{tabular}
\begin{tabular}{c|c|c|c}
& \{0\} & \{0, 1\} & \{1\} \\
\hline
\{0\} & 0 & 1 & 1 \\
\{1\} & 1 & 0 & 1 \\
\end{tabular}
\end{center}

Note that, dropping the middle column of the table of $\Pi$, we get the table of implication and, dropping the first line, we get that of the universal quantifier.

Independently of the pre-order relation $\leq$, we sometimes need to define elements of $B$ as fixed points of monotonic functions. To do so, we introduce an order relation $\sqsubseteq$ and a notion of completeness. This order relation extends to sets of elements of $B$ in a trivial way: $S \sqsubseteq T$ if for all $a$ in $S$, there exists a $b$ in $T$ such that $a \sqsubseteq b$.

Definition 3.3 (Ordered, complete $\Pi$-algebra) A $\Pi$-algebra is ordered if it is equipped with an order relation $\sqsubseteq$ such that the operation $\Pi$ is left anti-monotonic and right monotonic with respect to $\sqsubseteq$, that is

- if $a \sqsubseteq b$, then for all $S$, $\Pi(b, S) \sqsubseteq \Pi(a, S)$,
- if $S \sqsubseteq T$, then for all $a$, $\Pi(a, S) \sqsubseteq \Pi(a, T)$.

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

6
3.2 Models valued in a Π-algebra \( B \)

**Definition 3.4 (Model)** A model is a family of interpretation functions \( \mathcal{D}^1, ..., \mathcal{D}^n \) such that for all \( i \), \( \mathcal{D}^i \) is a function mapping each term \( t \) of type \( B \) in some context \( \Gamma \), function \( \phi_1 \) mapping each variable \( x : A \) of \( \Gamma \) to an element of \( \mathcal{D}^i_A \), ..., and function \( \phi_{i-1} \) mapping each variable \( x : A \) of \( \Gamma \) to an element of \( \mathcal{D}^{i-1}_{A,\phi_1, ..., \phi_{i-2}} \), to some \( \mathcal{D}^{i-1}_{B,\phi_1, ..., \phi_{i-2}} \) and for all \( t, u, \phi_1, ..., \phi_{i-1} \):

\[
\mathcal{D}^n(u/x)t,\phi_1, ..., \phi_{n-1} = \mathcal{D}^n_{t,\phi_1, ..., \phi_{n-1}}((\phi_{n-1}, x=\mathcal{D}^n_{t,\phi_1, ..., \phi_{n-1}}^n) \ldots ((\phi_1, x=\mathcal{D}^n_{t,\phi_1, ..., \phi_{n-1}}^1)) \ldots ((\phi_1, x=\mathcal{D}^n_{t,\phi_1, ..., \phi_{n-1}}))))
\]

For the last function \( \mathcal{D}^n \), we write \( [t]_{\phi_1, ..., \phi_{n-1}} \) instead of \( \mathcal{D}^n_{t,\phi_1, ..., \phi_{n-1}} \).

In the examples presented in this paper, we use the cases \( n = 2 \) and \( n = 3 \) only. The general definition then specializes as follows.

**Example.** When \( n = 2 \), a model is given by two functions \( \mathcal{M} \) and \([] \) such that

- \( \mathcal{M} \) is a function mapping each term \( t \) of type \( B \) in \( \Gamma \) to some \( \mathcal{M}_t \),
- \([] \) is a function mapping each term \( t \) of type \( B \) in \( \Gamma \) and function \( \phi \) mapping each variable \( x : A \) of \( \Gamma \) to an element of \( \mathcal{M}_A \), to some \( [t]_\phi \) in \( \mathcal{M}_B \), such that for all \( t, u \) and \( \phi \):

\[
[(u/x)t]_\phi = [t]_{\phi, x=[u]_\phi}
\]

This is the usual definition of model for many-sorted predicate logic.

**Remark.** If \( f \) is a constant of type \( A \to A \to A \), we can define the function \( f \) mapping each variable \( x : A \) of \( \Gamma \) to an element of \( \mathcal{N}_A \), to some \( \mathcal{N}_t, \psi \) in \( \mathcal{N}_B \), such that for all \( t, u, \psi \) and \( \phi \):

\[
[(u/x)t]_{\phi, \psi} = [t]_{\psi, x=[u]_\psi}
\]

which is the usual definition of the notion of denotation.

**Example.** When \( n = 3 \), a model is given by three functions \( \mathcal{N}, \mathcal{M}, \text{ and } [ ] \) such that

- \( \mathcal{N} \) is a function mapping each term \( t \) of type \( B \) in \( \Gamma \) to some \( \mathcal{N}_t \),
- \( \mathcal{M} \) is a function mapping each term \( t \) of type \( B \) in \( \Gamma \) and function \( \psi \) mapping each variable \( x : A \) of \( \Gamma \) to an element of \( \mathcal{N}_A \), to some \( \mathcal{M}_t, \psi \) in \( \mathcal{N}_B \),
- \([] \) is a function mapping each term \( t \) of type \( B \) in \( \Gamma \), function \( \psi \) mapping each variable \( x : A \) of \( \Gamma \) to an element of \( \mathcal{N}_A \), and function \( \phi \) mapping each variable \( x : A \) of \( \Gamma \) to an element of \( \mathcal{M}_A, \psi \), to some \( [t]_{\psi, \phi} \) in \( \mathcal{M}_B, \psi \), such that for all \( t, u, \psi \), and \( \phi \):

\[
[(u/x)t]_{\phi, \psi} = [t]_{\psi, x=[u]_\psi}
\]
Definition 3.5 (Model valued in a Π-algebra B) Let $B = (B, \bar{\top}, \bar{\Lambda}, \bar{R})$ be a Π-algebra. A model is valued in $B$ if

- $D^{n-1}_{\text{Kind}, \phi_1, \ldots, \phi_{n-2}} = D^{n-1}_{\text{Type}, \phi_1, \ldots, \phi_{n-2}} = B$,
- $[\text{Kind}]_{\phi_1, \ldots, \phi_{n-1}} = [\text{Type}]_{\phi_1, \ldots, \phi_{n-1}} = \bar{\top}$
- $[\Pi x : C \ D]_{\phi_1, \ldots, \phi_{n-1}} = \bar{\Pi}([C]_{\phi_1, \ldots, \phi_{n-1}}, \{[D]_{(\phi_1, x = c_1), \ldots, (\phi_{n-1}, x = c_{n-1})} \mid c_1 \in D^{1}_{C}, \ldots, c_{n-1} \in D^{n}_{C, \phi_1, \ldots, \phi_{n-1}}\})$

We often write $\vec{c}$ for a sequence $\phi_1, \ldots, \phi_n$ and, if $\bar{c} = c_1, \ldots, c_n$, we write $\vec{c}, x = \bar{c}$ for the sequence $(\phi_1, x = c_1), \ldots, (\phi_n, x = c_n)$.

Definition 3.6 (Validity) A model $\mathcal{M}$ valued in some Π-algebra $B$, of a theory $\Sigma, \mathcal{R}$, or the theory is said to be valid in the model, if

- for all constants $c : A$ in $\Sigma$, we have $[A] \geq \bar{\top}$,
- and for all $A$ and $B$ well-typed in a context $\Gamma$, such that $A \equiv_{\beta \eta} B$, we have for all $i$, for all $\vec{c}$, $D_{A, \vec{c}} = D_{B, \vec{c}}$.

Theorem 3.1 (Soundness) Let $\mathcal{M}$ be a model, valued in some Π-algebra $B$, of a theory $\Sigma, \mathcal{R}$. Then, for all judgments $x_1 : A_1, \ldots, x_p : A_p \vdash t : B$ derivable in $\Sigma, \mathcal{R}$, and for all $\vec{c}$, we have

$\llbracket A_1 \rrbracket_{\vec{c}} \land \ldots \land \llbracket A_p \rrbracket_{\vec{c}} \triangleq \llbracket B \rrbracket_{\vec{c}}$

Proof. By induction on the structure of the derivation of $x_1 : A_1, \ldots, x_p : A_p \vdash t : B$.

- If the last rule is Sort or Product, then $B = \text{Type}$ or $B = \text{Kind}$, $[B]_{\vec{c}} = \bar{\top}$ and

$\llbracket A_1 \rrbracket_{\vec{c}} \land \ldots \land \llbracket A_p \rrbracket_{\vec{c}} \leq \llbracket B \rrbracket_{\vec{c}}$

- If the last rule is Variable, with a constant of $\Sigma$, then $[B]_{\vec{c}} \geq \bar{\top}$ and

$\llbracket A_1 \rrbracket_{\vec{c}} \land \ldots \land \llbracket A_p \rrbracket_{\vec{c}} \leq \llbracket B \rrbracket_{\vec{c}}$

- If the last rule is Variable, with a variable of $\Gamma$, then $B = A_i$ and

$\llbracket A_1 \rrbracket_{\vec{c}} \land \ldots \land \llbracket A_p \rrbracket_{\vec{c}} \leq \llbracket B \rrbracket_{\vec{c}}$

- If the last rule is Abstraction, then $B = \Pi x : C \ D$ and by induction hypothesis, for all $\bar{c}$ in $D^{1}_{C} \times D^{2}_{C, \phi_1} \times \ldots \times D^{n}_{C, \phi_1, \ldots, \phi_{n-1}}$, we have

$\llbracket A_1 \rrbracket_{\vec{c}} \land \ldots \land \llbracket A_p \rrbracket_{\vec{c}} \land \llbracket C \rrbracket_{\vec{c}} \leq \llbracket D \rrbracket_{\vec{c}, x = \bar{c}}$

thus

$\llbracket A_1 \rrbracket_{\vec{c}} \land \ldots \land \llbracket A_p \rrbracket_{\vec{c}} \leq \bar{\Pi}([C]_{\vec{c}}, \{[D]_{\vec{c}, x = \bar{c}} \mid \bar{c} \in D^{1}_{C} \times D^{2}_{C, \phi_1} \times \ldots \times D^{n}_{C, \phi_1, \ldots, \phi_{n-1}}\})$

$\llbracket A_1 \rrbracket_{\vec{c}} \land \ldots \land \llbracket A_p \rrbracket_{\vec{c}} \leq \llbracket \Pi x : C \ D \rrbracket_{\vec{c}}$

that is

$\llbracket A_1 \rrbracket_{\vec{c}} \land \ldots \land \llbracket A_p \rrbracket_{\vec{c}} \leq \llbracket B \rrbracket_{\vec{c}}$
If the last rule is Application, then we have $B = (u/x)D$ and by, induction hypothesis

$$[A_1]_\sigma \ldots \tilde{\lambda} [A_p]_\sigma \leq [C]_\sigma$$

and

$$[A_1]_\sigma \ldots \tilde{\lambda} [A_p]_\sigma \leq [\Pi x : C D]_\sigma$$

Thus, for all $\tau$ in $D_{\sigma 1}^1 \times D_{\sigma 1, \phi_1}^2 \times \ldots \times D_{\sigma 1, \phi_1, \ldots, \phi_{n-1}}^n$, we have

$$[A_1]_\tau \ldots \tilde{\lambda} [A_p]_\tau \tilde{\lambda} [C]_\tau \leq [D]_{\tau, x=\tau}$$

In particular, for $\tau = D_{\sigma 1}^1, D_{\sigma 1, \phi_1}^2, \ldots, D_{\sigma 1, \phi_1, \ldots, \phi_{n-1}}^n$, we get

$$[A_1]_\tau \ldots \tilde{\lambda} [A_p]_\tau \tilde{\lambda} [C]_\tau \leq [(u/x)D]_\tau$$

$$[A_1]_\tau \ldots \tilde{\lambda} [A_p]_\tau \tilde{\lambda} [C]_\tau \leq [B]_\tau$$

Hence, as $[A_1]_\tau \ldots \tilde{\lambda} [A_p]_\tau \leq [C]_\tau$, we have

$$[A_1]_\tau \ldots \tilde{\lambda} [A_p]_\tau \leq [B]_\tau$$

If the last rule is Conversion, then we use the fact that the model is a model of $\mathcal{R}$.

**Corollary 3.1** Let $M$ be a model, valued in some $\Pi$-algebra $B$, of a theory $\Sigma, \mathcal{R}$. Then, for all judgments $\vdash t : B$ derivable in $\Sigma, \mathcal{R}$, we have $[B]_\tau \geq \top$.

**Corollary 3.2** Let $M$ be a model, valued in the two-element $\Pi$-algebra of Example 3.1, of a theory $\Sigma, \mathcal{R}$. Then, for all judgments $\vdash t : B$, derivable in this theory, we have $[B]_\tau = 1$.

**Corollary 3.3 (Consistency)** Let $\Sigma, \mathcal{R}$ be a theory that has a model, valued in the two-element $\Pi$-algebra of Example 3.1. Then, there is no term $t$ such that the judgment $P : Type \vdash t : P$ is derivable in $\Sigma, \Gamma$.

## 4 Super-consistency

### 4.1 Super-consistency

**Definition 4.1 (Super-consistency)** A theory $\Sigma, \mathcal{R}$, is said to be super-consistent if for every full, ordered and complete $\Pi$-algebra $B$, there exists a model $M$, valued in $B$, of $\Sigma, \mathcal{R}$.

In the remainder of this section, we prove that the three theories presented in Section 2.3 are super-consistent.

### 4.2 Simple type theory

Let $B = (\mathcal{B}, \tilde{\tau}, \tilde{\lambda}, \mathcal{P}^+(\mathcal{B}), \tilde{\Pi})$ be a full $\Pi$-algebra. We construct a model of Simple type theory, valued in $B$, in two steps. The first is the construction of the interpretation function $M$ and the proof of the validity of the congruence for this function. The second is the construction of the interpretation function $\tilde{\lambda}$ and the proof of the validity of the congruence for this function. The key idea in this construction is to take $M_0 = B$, to interpret $\varepsilon$ as the identity over $B$, and $\Rightarrow$ like $\rightarrow$ in order to validate the rewrite rule

$$\varepsilon (\Rightarrow x y) \quad \rightarrow (\varepsilon x) \rightarrow (\varepsilon y)$$

**Definition 4.2** Let $S$ and $T$ be two sets, we write $\mathcal{F}(S,T)$ for the set of functions from $S$ to $T$. 9
4.2.1 The interpretation function $M$

The first step of the proof is the construction of the interpretation function $M$.

Let $\{e\}$ be an arbitrary one-element set such that $e$ is not in $B$.

**Definition 4.3** The interpretation function $M$ is defined as follows

- $M_{\text{Kind}} = M_{\text{Type}} = B$,
- $M_{\Pi \in C \ D} = F(M_C, M_D)$, except if $M_D = \{e\}$, in which case $M_{\Pi \in C \ D} = \{e\}$,
- $M_t = M = M_{\forall x} = M_e = \{e\}$,
- $M_v = B$,
- $M_x = \{e\}$,
- $M_{\lambda x \in C \ t} = M_t$,
- $M_{(t \ u)} = M_t$.

We first prove the two following lemmas.

**Lemma 4.1** If the term $t$ is an object, then

$$M_t = \{e\}$$

*Proof.* By induction on the structure of the term $t$. The term $t$ is neither $\text{Kind}$, $\text{Type}$, nor $o$. It is not a product. If it has the form $\lambda x : C \ t'$, then $t'$ is an object. If it has the form $(t' \ t'')$, then $t'$ is an object.

**Lemma 4.2** If $u$ is an object then

$$M_{(u/x)t} = M_t$$

*Proof.* By induction on the structure of the term $t$. If $t = x$ then, by Lemma 4.1

$$M_{(u/x)t} = M_u = \{e\} = M_t$$

If $t$ is $\text{Kind}$, $\text{Type}$, a constant, or a variable different from $x$, then $x$ does not occur in $t$. If it is a product, an abstraction, or an application, we use the induction hypothesis.

**Lemma 4.3** (Validity of the congruence) If $t \equiv_{\betaR} u$ then

$$M_t = M_u$$

*Proof.* If $t = (\lambda x : C \ t') \ u'$, then $u'$ is an object and by Lemma 4.2

$$M_{(\lambda x : C \ t') \ u'} = M_{u'} = M_{(u'/x)t'}$$

Then, as for all $v$, $M_v = M_e = \{e\}$, and if $M_D = \{e\}$, then $M_{\Pi \in C \ D} = \{e\}$, we have

$$M_{(\varepsilon (C \Rightarrow D))} = \{e\} = M_{(\varepsilon (C \Rightarrow D))}$$

and

$$M_{(\varepsilon (D \ x))} = \{e\} = M_{(\varepsilon (\forall x \ C \ D))}$$

We prove, by induction on $t$, that if $t \rightarrow^{1}_{\betaR} u$ then $M_t = M_u$, and we conclude with a simple induction on the structure of a reduction of $t$ and $u$ to a common term.
4.2.2 The interpretation function \([\cdot]\)

The second step of the proof is the construction of the interpretation function \([\cdot]\) and the proof of the validity of the congruence for this function.

**Definition 4.4** The interpretation function \([\cdot]\) is defined as follows

- \([\text{Kind}]\) = \([\text{Type}]\) = \(\top\),
- \([\Pi x : C \ D] = \tilde{\Pi}(\{\text{Type}\}_{\phi}, \{\text{Kind}\}_{\phi, x=c} \mid c \in M_C\})
- \([\epsilon] = \tilde{\top}\),
- \([\circ] = \tilde{\top}\),
- \([\Rightarrow] = \text{the function mapping } a \text{ and } b \text{ in } B \text{ to } \tilde{\Pi}(\{a\}, \{\{b\}\})
- \([\forall C] = \text{the function mapping } f \text{ in } F(M_C, B) \text{ to } \tilde{\Pi}(\{C\}_{\phi}, \{(f \ c) \mid c \in M_C\})
- \([\epsilon] = \text{the identity on } B,
- \([\lambda x : C \ t] = \text{the function mapping } c \in M_C \text{ to } \{t\}_{\phi, x=c}, \text{ except if for all } c \in M_C, \{t\}_{\phi, x=c} = e \text{ in which case } \{t\}_{\phi} = e,
- \([t \ u] = \{t\}_{\phi} \{u\}_{\phi}, \text{ except if } \{t\}_{\phi} = e, \text{ in which case } \{t \ u\}_{\phi} = e.

**Lemma 4.4 (Well-typedness)** If \(\Gamma \vdash t : B\), and \(\phi\) is a function mapping variables \(x : A\) of \(\Gamma\) to elements of \(M_A\), then

\[\{t\}_{\phi} \in M_B\]

**Proof.** We check each case of the definition of \([\cdot]\).

**Lemma 4.5 (Substitution)** For all \(t, u\) and \(\phi\)

\[\{u/x\} t = \{t\}_{\phi, x=\{u\}_{\phi}}\]

**Proof.** By induction on the structure of the term \(t\).

**Lemma 4.6 (Validity of the congruence)** If \(t \equiv_{\beta R} u\) then

\[\{t\}_{\phi} = \{u\}_{\phi}\]

**Proof.** If \(t = (\lambda x : C \ t') \ u')\), then if for all \(c \in M_C\), we have \(\{t'\}_{\phi, x=c} = e\), then

\[\{(\lambda x : C \ t') \ u')\}_{\phi} = e = \{t'\}_{\phi, x=[u']_{\phi}} = \{u'/x\} t'_{\phi}\]

Otherwise

\[\{(\lambda x : C \ t') \ u')\}_{\phi} = \{t'\}_{\phi, x=[u']_{\phi}} = \{u'/x\} t'_{\phi}\]

Then

\[\{\epsilon \Rightarrow t' \ u')\}_{\phi} = \tilde{\Pi}(\{t'\}_{\phi}, \{\{u'\}_{\phi}\}) = \{\epsilon \rightarrow (t' \ u')\}_{\phi}\]

and

\[\{\epsilon \ (\forall C \ t')\}_{\phi} = \tilde{\Pi}(\{C\}_{\phi}, \{(t'\}_{\phi} \ e \mid c \in M_C\}) = \\Pi y : C \ (\epsilon \ (t' \ y))_{\phi}\]

We prove, by induction on \(t\), that if \(t \rightarrow_{\beta R} u\) then

\[\{t\}_{\phi} = \{u\}_{\phi}\]

and we conclude with a simple induction on the structure of a reduction of \(t\) and \(u\) to a common term.

We thus get the following theorem.

**Theorem 4.1** Simple type theory is super-consistent.
4.3 Simple type theory with a parametric quantifier

In a model of Simple type theory with a parametric quantifier, like in the previous section, we want to take $\mathcal{M}_o = \mathcal{B}$. But, unlike in the previous section, we do not have $o : Type$, but $o : type : Type$. So $o$ is now an object.

In the previous section, we took $\mathcal{M}_t = \{ e \}$ for all objects. This permitted to define $\mathcal{M}_{(t,u)}$ and $\mathcal{M}_{\lambda x.C.$ as $\mathcal{M}_t$ and validate $\beta$-reduction trivially. But this is not possible anymore in Simple type theory with a parametric quantifier, where $\mathcal{M}_o$ is $\mathcal{B}$ and $\mathcal{M}_{\text{arrow}(o,o)}$ is $\mathcal{F}(\mathcal{B}, \mathcal{B})$. So, we cannot define $\mathcal{M}_{\lambda x.type} x$ to be $\mathcal{M}_x$, hence always the same set, but we need to define it as a function. To help to construct this function, we need to construct first another interpretation function ($\mathcal{N}_t$), and parametrize the definition of $\mathcal{M}_t$ itself by a function $\psi$ mapping variables of type $A$ to elements of $\mathcal{N}_A$.

We thus construct the model in three steps. The first is the construction of the interpretation function $\mathcal{N}$. The second is the construction of the interpretation function $\mathcal{M}$. The third is the construction of the interpretation function $\llbracket \rrbracket$.

Like in the previous section, we want to define $\mathcal{M}_{\Pi x.C, D, \psi}$, as the set of functions from $\mathcal{M}_{C, \psi}$ to $\mathcal{M}_D$. But to define this set $\mathcal{M}_D$, we need to extend the function $\psi$, mapping $x$ to an element of $\mathcal{N}_C$. To have such an element of $\mathcal{N}_C$, we need to define $\mathcal{M}_{\Pi x.C, D, \psi}$ as the set of functions mapping $(e', c)$ in $\mathcal{N}_C \times \mathcal{M}_{C, \psi}$ to an element of $\mathcal{M}_{D,(\psi,x=e')}$. As a consequence, if $\phi$ is a function mapping $x$ of type $A$ to some element of $\mathcal{M}_A$, we need to define $\llbracket (t \; u) \rrbracket_{\phi}$ not as $\llbracket t \rrbracket_{\phi} \llbracket u \rrbracket_{\phi}$ but as $\llbracket t \rrbracket \llbracket u \rrbracket$. As a consequence $\llbracket \cdot \rrbracket$ must be parametrized by both $\psi$ and $\phi$.

Let $\mathcal{B} = (\mathcal{B}, \tilde{\top}, \tilde{\bot}, \mathcal{P}^+(\mathcal{B}), \tilde{\Pi})$ be a full $\Pi$-algebra.

4.3.1 The interpretation function $\mathcal{N}$

The first step of the proof is the definition of the interpretation function $\mathcal{N}$ and the proof of the validity of the congruence for this function.

Let $\{ e \}$ be an arbitrary one-element set. Let $\mathcal{U}$ be a set containing $\mathcal{B}$ and $\{ e \}$, and closed by function space and Cartesian product, that is such that if $S$ and $T$ are in $\mathcal{U}$ then so are $S \times T$ and $\mathcal{F}(S,T)$. Such a set can be constructed, with the replacement scheme, as follows.

$$\mathcal{U}_0 = \{ \mathcal{B}, \{ e \} \}$$

$$\mathcal{U}_{n+1} = \mathcal{U}_n \cup \{ S \times T \mid S, T \in \mathcal{U}_n \} \cup \{ \mathcal{F}(S,T) \mid S, T \in \mathcal{U}_n \}$$

$$\mathcal{U} = \bigcup_n \mathcal{U}_n$$

Then, let $\mathcal{V}$ be the smallest set containing $\{ e \}$, $\mathcal{B}$, and $\mathcal{U}$, and closed by Cartesian product and dependent function space, that is, if $S$ is in $\mathcal{V}$ and $T$ is a family of elements of $\mathcal{V}$ indexed by $S$, then the set of functions mapping an element $s$ of $S$ to an element of $T_s$ is an element of $\mathcal{V}$. As noted in [16], the construction of the set $\mathcal{V}$, unlike that of $\mathcal{U}$, requires an inaccessible cardinal. Note that $\mathcal{U}$ is both an element and a subset of $\mathcal{V}$.

Definition 4.5 The interpretation function $\mathcal{N}$ is defined as follows

- $\mathcal{N}_{\text{Type}} = \mathcal{N}_{\text{Kind}} = \mathcal{V}$,
- $\mathcal{N}_{\Pi x.C, D}$ is the set $\mathcal{F}(\mathcal{N}_C, \mathcal{N}_D)$, except if $\mathcal{N}_D = \{ e \}$, in which case $\mathcal{N}_{\Pi x.C, D} = \{ e \}$,
- $\mathcal{N}_{\text{type}} = \mathcal{U}$,
- $\mathcal{N}_e = \mathcal{N}_o = \mathcal{N}_{\text{arrow}} = \mathcal{N}_o = \mathcal{N}_\psi = \mathcal{N}_\eta = \mathcal{N}_e = \{ e \}$,
- $\mathcal{N}_{\lambda x.C} t = \mathcal{N}_t$,
- $\mathcal{N}_{(t,u)} = \mathcal{N}_t$. 

12
We first prove the two following lemmas.

**Lemma 4.7** If the term $t$ is an object, then

$$N_t = \{e\}$$

*Proof.* By induction on the structure of the term $t$. The term $t$ is neither Kind, Type, nor type. It is not a product. If it has the form $\lambda x : C \ t'$, then $t'$ is an object. If it has the form $(t' \ t'')$, then $t'$ is an object.

**Lemma 4.8** If $u$ is an object, then

$$N_{(u/x)t} = N_t$$

*Proof.* By induction on the structure of the term $t$. If $t = x$ then, by Lemma 4.7

$$N_{(u/x)t} = N_u = \{e\} = N_t$$

If $t$ is Kind, Type, a constant, or a variable different from $x$, then $x$ does not occur in $t$. If it is a product, an abstraction, or an application, we use the induction hypothesis.

**Lemma 4.9 (Validity of the congruence)** If $t \equiv_{\beta R} u$ then

$$N_t = N_u$$

*Proof.* If $t = ((\lambda x : C \ t') \ u')$, then $u'$ is an object and by Lemma 4.8

$$N_{((\lambda x : C \ t') \ u')} = N_{u'} = N_{(u'/x)t'}$$

Then, as for all $v, N_{(\eta \ v)} = N_\eta = \{e\}$ and if $N_D = \{e\}$, then $N_{\Pi x : C \ D} = \{e\}$, we have

$$N_{(\eta \ (\text{arrow} \ C \ D))} = \{e\} = N_{((\eta \ C) \to (\eta \ D))}$$

As for all $v, N_{(\varepsilon \ v)} = N_\varepsilon = \{e\}$, and if $N_D = \{e\}$, then $N_{\Pi x : C \ D} = \{e\}$, we have

$$N_{(\varepsilon \ (\Rightarrow \ C \ D))} = \{e\} = N_{((\varepsilon \ C) \to (\varepsilon \ D))}$$

and

$$N_{(\varepsilon \ (\forall \ C \ D))} = \{e\} = N_{\Pi x : (\eta \ C) \ (\varepsilon \ (D \ x))}$$

We prove, by induction on $t$, that if $t \to_{\beta R}^1 u$ then $N_t = N_u$ and we conclude with a simple induction on the structure of a reduction of $t$ and $u$ to a common term.

### 4.3.2 The interpretation function $M$

The second step of the proof is the definition of the interpretation function $M$ and the proof of the validity of the congruence for this function.

**Definition 4.6** The interpretation function $M$ is defined as follows

- $M_{\text{Kind}, \psi} = M_{\text{Type}, \psi} = B$,
- $M_{\Pi x : C \ D, \psi, \phi}$ is the set of functions $f$ mapping $(c', c)$ in $N_C \times M_{C, \psi}$ to an element of $M_{D, (\phi, x = c')}$, except if for all $c'$ in $N_C, M_{D, (\phi, x = c')} = \{e\}$, in which case $M_{\Pi x : C \ D, \psi} = \{e\}$,
- $M_{\text{Type}, \psi} = B$,
- $M_{\eta, \psi}$ is the function of $\mathcal{F}(\mathcal{U}, \mathcal{V})$ mapping $S$ to $S$,
- $M_{\varepsilon, \psi}$ is the function of $\mathcal{F}(\{e\}, \mathcal{V})$, mapping $e$ to $\{e\}$,
- $M_{\iota, \psi} = \{e\}$. 

13
• \( M_{\psi, \psi} = B \),
• \( M_{\downarrow, \psi} \) is the function mapping \( S \) and \( T \) in \( U \) to the set \( F(\{e\} \times S, T) \), except if \( T = \{e\} \) in which case it maps \( S \) and \( T \) to \( F(\{e\} \times S, T) \),
• \( M_{\psi, \psi} = M_{\psi, \psi} = e \),
• \( M_{x, \psi} = \psi x \),
• \( M_{\lambda x : C t, \psi} \) is the function mapping \( c \) in \( N_C \) to \( M_{t, \psi x = c} \), except if for all \( c \) in \( N_C \), \( M_{t, \psi x = c} = e \), in which case \( M_{\lambda x : C t, \psi} = e \),
• \( M_{(t u), \psi} = M_{t, \psi} M_{u, \psi} \), except if \( M_{t, \psi} = e \) in which case \( M_{(t u), \psi} = e \).

**Lemma 4.10** If \( \Gamma \vdash C : Type \), then \( N_C \in V \)

**Proof.** By induction on the structure of the term \( C \). As this term has type \( Type \), it is neither \( Kind \) nor \( Type \).

**Lemma 4.11 (Well-typedness)** If \( \Gamma \vdash t : B \) and \( \psi \) is a function mapping the variables \( x : A \) of \( \Gamma \) to elements of \( N_A \), then \( M_{t, \psi} \in N_B \)

**Proof.** We check each case of the definition of \( M \).

**Lemma 4.12 (Substitution)** For all \( t, u \) and \( \psi \)

\[
M_{(u / x) t, \psi} = M_{t, \psi x = M_{u, \psi}}
\]

**Proof.** By induction on the structure of the term \( t \).

**Lemma 4.13 (Validity of the congruence)** If \( t \equiv B \) then

\[
M_{t, \psi} = M_{u, \psi}
\]

**Proof.** If \( t = ((\lambda x : C t') u') \), then if for all \( e \) in \( N_C \) \( M_{t', \psi x = e} = e \), then

\[
M_{((\lambda x : C t') u'), \psi} = e = M_{t', \psi x = M_{u', \psi}} = M_{(u' / x) t', \psi}
\]

Otherwise

\[
M_{((\lambda x : C t') u'), \psi} = M_{t', \psi x = M_{u', \psi}} = M_{(u' / x) t', \psi}
\]

The set \( M_{(\eta (\text{arrow } C D)), \psi} \) is the set \( F(\{e\} \times M_{C, \psi}, M_{D, \psi}) \), except if \( M_{D, \psi} = \{e\} \), in which case \( M_{(\eta (\text{arrow } C D)), \psi} = \{e\} \). The set \( M_{((\eta C) \rightarrow (\eta D)), \psi} \) is this same set. Thus

\[
M_{(\eta (\text{arrow } C D)), \psi} = M_{((\eta C) \rightarrow (\eta D)), \psi}
\]

We have

\[
M_{(\varepsilon \Rightarrow C D), \psi} = \{e\} = M_{((\varepsilon C) \rightarrow (\varepsilon D)), \psi}
\]

and

\[
M_{(\varepsilon \forall C D), \psi} = \{e\} = M_{\Pi x : (\varepsilon C) (\varepsilon D), \psi}
\]

We prove, by induction on \( t \), that if \( t \rightarrow B \) then \( M_{t, \psi} = M_{u, \psi} \) and we conclude with a simple induction on the structure of a reduction of \( t \) and \( u \) to a common term.
The last step of the proof is the definition of the interpretation function $\llbracket \cdot \rrbracket$ and the proof of the validity of the congruence for this function.

**Definition 4.7** The interpretation function $\llbracket \cdot \rrbracket$ is defined as follows

- $[\text{Kind}]_{\psi,\phi} = [\text{Type}]_{\psi,\phi} = \top$,
- $[\Pi x : C \ D]_{\psi,\phi} = \ubar{\Pi}(\{C\}_{\psi,\phi}, \{[D]_{(\psi,x=c'),(\phi,x=c)} \mid c' \in \mathcal{N}_C, c \in \mathcal{M}_C,\phi\})$,
- $[\text{type}]_{\psi,\phi} = \top$,
- $[\text{let}]_{\psi,\phi} = \top$,
- $[\text{let}]_{\psi,\phi} = \top$,
- $[\text{arrow}]_{\psi,\phi} = \text{the function from } U \times B$ and $U \times B$ to $B$ mapping $\langle S, a \rangle$ and $\langle T, b \rangle$ to $\ubar{\Pi}(a, \{b\})$,
- $[\text{address}]_{\psi,\phi} = \text{the function mapping } \langle S, a \rangle$ to $a$,
- $[\text{call}]_{\psi,\phi} = \text{the function mapping } \langle e, a \rangle$ to $a$,
- $[\text{call}]_{\psi,\phi} = \text{the function mapping } \langle e, a \rangle$ to $a$,
- $[\text{call}]_{\psi,\phi} = \text{the function mapping } \langle e, a \rangle$ to $a$,
- $[\text{call}]_{\psi,\phi} = \text{the function mapping } \langle e, a \rangle$ to $a$,
- $[\text{call}]_{\psi,\phi} = \text{the function mapping } \langle e, a \rangle$ to $a$,
- $[\text{call}]_{\psi,\phi} = \text{the function mapping } \langle e, a \rangle$ to $a$,
- $[\text{call}]_{\psi,\phi} = \text{the function mapping } \langle e, a \rangle$ to $a$,
- $[\text{call}]_{\psi,\phi} = \text{the function mapping } \langle e, a \rangle$ to $a$,
- $[\text{call}]_{\psi,\phi} = \text{the function mapping } \langle e, a \rangle$ to $a$.

**Lemma 4.14 (Well-typedness)** If $\Gamma \vdash t : B$, $\psi$ is a function mapping variables $x : A$ of $\Gamma$ to elements of $\mathcal{N}_A$, and $\phi$ is a function mapping variables $x : A$ of $\Gamma$ to elements of $\mathcal{M}_A,\psi$, then

$$[t]_{\psi,\phi} \in \mathcal{M}_B,\psi$$

**Proof.** We check each case of the definition of $\llbracket \cdot \rrbracket$.

Let us check, for instance, that if $t = (t_1 \ t_2)$, $t_1$ has type $\Pi x : C \ D$ and $t_2$ has type $C$, hence $(t_1 \ t_2)$ has type $(t_2/x)D$, then $[t_1 \ t_2]_{\psi,\phi}$ is in $\mathcal{M}_{(t_2/x)D,\psi}$. We have $\mathcal{M}_{t_2,\psi}$ is in $\mathcal{N}_C$ and, by induction hypothesis, $[t_1]_{\psi,\phi}$ is in $\mathcal{M}_{\Pi x : C \ D,\psi}$, and $[t_2]_{\psi,\phi}$ is in $\mathcal{M}_{\psi,\phi}$. We have $[t_1 \ t_2]_{\psi,\phi} = [t_1]_{\psi,\phi} \ \mathcal{M}_{t_2,\psi}, [t_2]_{\psi,\phi}$ and, by definition of $\mathcal{M}_{\Pi x : C \ D,\psi}$, this term is in $\mathcal{M}_{D,\psi}$, so $\mathcal{M}_{(t_2/x)D,\psi}$.

**Lemma 4.15 (Substitution)** For all $t$, $u$, $\psi$, and $\phi$

$$[u/x]t_{\psi,\phi} = [t]_{\psi,x = M_u,\psi,\phi,x = [u]_{\psi,\phi}}$$

**Proof.** By induction on the structure of the term $t$.

**Lemma 4.16 (Validity of the congruence)** If $t \equiv_{\beta_R} u$ then

$$[t]_{\psi,\phi} = [u]_{\psi,\phi}$$
Remark. The set $\mathcal{V}$, and hence an inaccessible cardinal, are not really needed to prove the consistency and the super-consistency of Simple type theory with a parametric quantifier. It is indeed possible to adapt the notion of model in such a way that the family $\mathcal{V}$ is defined for type families only. Then, Lemma 4.11 is proved for objects only. This is sufficient to define $\mathcal{M}_{\lambda x : a \rightarrow b}$ as the identity on $U$ and, more generally, the function $\mathcal{M}$. In this case, the class of sets $\mathcal{M}_{a \rightarrow b}$ would not be a set, which is common in models of many sorted Predicate logic with an infinite number of sorts.

We leave the systematic development of this notion of model with partial interpretation functions for future work.

4.4 The Calculus of constructions

A very similar proof can be made for the Calculus of constructions.

In the construction of the interpretation functions $\mathcal{V}$, $\mathcal{M}$, and $\llbracket \rrbracket$, we drop the clauses associated to the symbols $\top$, $\Rightarrow$, $\forall$ and $\rightarrow$ and we add the clauses.

- $\mathcal{N}_{\Pi_{TT}} = \mathcal{N}_{\Pi_{TK}} = \mathcal{N}_{\Pi_{KK}} = \{ e \}$
- $\mathcal{M}_{\Pi_{TT}}$ is the function mapping $S$ in $U$ and $h$ in $\mathcal{F}(\{ e \}, U)$ to the set $\mathcal{F}(\{ e \} \times S, \langle h, e \rangle)$, except if $\langle h, e \rangle = \{ e \}$ in which case it maps $S$ and $h$ to $\{ e \}$,
- $\mathcal{M}_{\Pi_{TT}} = e$,
- $\mathcal{M}_{\Pi_{TK}} = e$,
- $\mathcal{M}_{\Pi_{KK}}$ is the function mapping $e$ and $h$ in $\mathcal{F}(\{ e \}, U)$ to the set $\mathcal{F}(\{ e \} \times \{ e \}, \langle h, e \rangle)$, except if $\langle h, e \rangle = \{ e \}$ in which case it maps $e$ and $h$ to $\{ e \}$,
- $\llbracket \Pi_{TT} \rrbracket_{\psi, \phi}$ is the function mapping $\langle S, a \rangle$ in $U \times \mathcal{B}$, $\langle f, g \rangle$ in $\mathcal{F}(\{ e \}, U) \times \mathcal{F}(\{ e \} \times S, \mathcal{B})$ to $\Pi(a, \{ \langle g, \langle e, s \rangle \rangle \mid s \in S \})$,
- $\llbracket \Pi_{TT} \rrbracket_{\psi, \phi}$ is the function mapping $\langle e, a \rangle$ in $\{ e \} \times \mathcal{B}$, and $\langle e, g \rangle$ in $\{ e \} \times \mathcal{F}(\{ e \} \times \{ e \}, \mathcal{B})$ to $\Pi(a, \{ \langle g, \langle e, e \rangle \rangle \})$,
- $\llbracket \Pi_{KK} \rrbracket_{\psi, \phi}$ is the function mapping $\langle S, a \rangle$ in $U \times \mathcal{B}$, and $\langle e, g \rangle$ in $\{ e \} \times \mathcal{F}(\{ e \} \times S, \mathcal{B})$ to $\Pi(a, \{ \langle g, \langle e, s \rangle \rangle \mid s \in S \})$,
- $\llbracket \Pi_{KK} \rrbracket_{\psi, \phi}$ is the function mapping $\langle e, a \rangle$ in $\{ e \} \times \mathcal{B}$, and $\langle f, g \rangle$ in $\mathcal{F}(\{ e \}, U) \times \mathcal{F}(\{ e \} \times \{ e \}, \mathcal{B})$ to $\Pi(a, \{ \langle g, \langle e, e \rangle \rangle \})$.
The proof of Lemmas 4.7 and 4.8 are similar.
The proof of Lemma 4.9 is similar, except for the case of rewrite rules.

\[
N_{(\eta (\Pi_{KK} C D))} = \{e\} = N_{\Pi_{x: (\eta C)} (\eta (D z))}
\]

\[
N_{(\varepsilon (\Pi_{TT} C D))} = \{e\} = N_{\Pi_{x: (\varepsilon C)} (\varepsilon (D z))}
\]

\[
N_{(\varepsilon (\Pi_{KT} C D))} = \{e\} = N_{\Pi_{x: (\varepsilon C)} (\varepsilon (D z))}
\]

\[
N_{(\eta (\Pi_{KK} C D))} = \{e\} = N_{\Pi_{x: (\eta C)} (\eta (D z))}
\]

The proof of Lemma 4.10 is similar.
The proof of Lemma 4.11 must be adapted to check the case of the symbols \(\tilde{\Pi}_{KK}, \tilde{\Pi}_{TT}, \tilde{\Pi}_{KT}\), and \(\tilde{\Pi}_{FK}\).
The proof of Lemma 4.12 is similar.
The proof of Lemma 4.13 is similar, except for the case of rewrite rules.
The set \(M_{(\eta (\Pi_{KK} C D), \psi)}\) is the set \(F(\{e\} \times M_{C, \psi}, (M_{D, \psi} e))\), except if \((M_{D, \psi} e) = \{e\}\) in which case \(M_{(\eta (\Pi_{KK} C D), \psi)} = \{e\}\). The set \(M_{\Pi_{x: (\eta C)} (\eta (D z)), \psi}\) is this same set. Thus

\[
M_{(\eta (\Pi_{KK} C D), \psi)} = M_{\Pi_{x: (\eta C)} (\eta (D z)), \psi}
\]

We have

\[
M_{(\varepsilon (\Pi_{TT} C D))} = \{e\} = M_{\Pi_{x: (\varepsilon C)} (\varepsilon (D z))}
\]

We have

\[
M_{(\varepsilon (\Pi_{KT} C D))} = \{e\} = M_{\Pi_{x: (\varepsilon C)} (\varepsilon (D z))}
\]

The proof of Lemma 4.14 must be adapted to check the case of the symbols \(\tilde{\Pi}_{KK}, \tilde{\Pi}_{TT}, \tilde{\Pi}_{KT}\), and \(\tilde{\Pi}_{FK}\).
The proof of Lemma 4.15 is similar.
The proof of Lemma 4.16 is similar, except for the case of rewrite rules.

\[
\left[ (\eta (\tilde{\Pi}_{KK} C D)) \right]_{\psi, \phi} = \tilde{\Pi}(\tilde{C}_{\psi, \phi}, \{(D)_{\psi, \phi} (\varepsilon, s) \mid s \in M_{C, \psi}\}) = \Pi_{y : (\eta C) (\eta (D y))}
\]

\[
\left[ (\varepsilon (\tilde{\Pi}_{TT} C D)) \right]_{\psi, \phi} = \tilde{\Pi}(\tilde{C}_{\psi, \phi}, \{(D)_{\psi, \phi} (\varepsilon, e) \mid e \in M_{C, \psi}\}) = \Pi_{y : (\varepsilon C) (\varepsilon (D y))}
\]

\[
\left[ (\varepsilon (\tilde{\Pi}_{KT} C D)) \right]_{\psi, \phi} = \tilde{\Pi}(\tilde{C}_{\psi, \phi}, \{(D)_{\psi, \phi} (\varepsilon, e) \mid e \in M_{C, \psi}\}) = \Pi_{y : (\varepsilon C) (\varepsilon (D y))}
\]

5 Termination of proof reduction

We now prove that proof reduction terminates in the \(\lambda\)-calculus modulo any super-consistent theory such as Simple type theory without or with a parametric quantifier or the Calculus of constructions.

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, the rewrite rule \(c \rightarrow c\) does not terminate. But, the congruence defined by this rewrite rule is the identity and proofs modulo this theory are just proofs in pure Predicate logic. Thus, proof reduction in Deduction modulo this theory terminates. Hence in this theory, the \(\beta\)-reduction terminates, but the \(\beta R\)-reduction does not terminate, as the \(R\)-reduction alone does not terminate.

In this paper, we restrict to prove the termination of \(\beta\)-reduction, not \(\beta R\)-reduction. In some cases the termination of the \(\beta R\)-reduction is a simple corollary of the termination of the \(\beta\)-reduction (see Section 5.3). In some others, it is not.

The main notion used in this proof is that of reducibility candidate introduced by Girard [12]. Our inductive definition, however, follows that of Parigot [18].

17
5.1 The candidates

Definition 5.1 (Operations on set of terms) The set ̂\top 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 ̂\( \Pi(C,S) \) is defined as the set of strongly terminating terms \( t \) such that if \( t \rightarrow_\beta^* \lambda x : A \ t' \) then for all \( t'' \in C \), and for all \( D \in S \), \( (t''/x)t' \in D \).

The main property of the operation ̂\( \Pi \) is expressed by the following Lemma.

Lemma 5.1 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 \( t_1 \in ̂\Pi(C,S), t_2 \in C, \) and \( (t_1 \ t_2) \rightarrow_\beta^* u, n_1 \) and \( n_2 \) be natural numbers such that \( n_1 \) is the maximum length of a reduction sequence issued from \( t_1 \), and \( n_2 \) 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 ̂\( \Pi(C,S), u \in D \). Otherwise, the reduction takes place in \( t_1 \) or in \( t_2 \), and we apply the induction hypothesis.

Definition 5.2 (Candidates) Candidates are inductively defined by the three rules

- the set ̂\top of all strongly terminating terms is a candidate,
- if \( C \) is a candidate and \( S \) is a set of candidates, then ̂\( \Pi(C,S) \) is a candidate,
- if \( S \) is a non empty set of candidates, then ̂\( \bigcap S \) is a candidate.

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

The algebra \( (\mathcal{C}, \leq, ̂\top, ^+, \Pi, ̂\Pi) \), where \( \leq \) is the trivial relation such that \( C \leq C' \) always, and ̂\( \lambda \) is any function from \( C \times C \) to \( C \), for instance the constant function equal to ̂\top, is a full ̂\( \Pi \)-algebra.

It is ordered by the subset relation and complete for this order.

Lemma 5.2 (Termination) If \( C \) is a candidate, then all the elements of \( C \) strongly terminate.

Proof. By induction on the construction of \( C \).

Lemma 5.3 (Variables) If \( C \) is a candidate and \( x \) is a variable, then \( x \in C \).

Proof. By induction on the construction of \( C \).

Lemma 5.4 (Closure by reduction) If \( C \) is a candidate, \( t \in C \), and \( t \rightarrow_\beta^* t' \), then \( t' \in C \).

Proof. By induction on the construction of \( C \).

If \( C = ̂\top \), then as \( t \) is an element of \( C \), it strongly terminates, thus \( t' \) strongly terminates, and \( t' \in C \).

If \( C = ̂\Pi(D,S) \), then as \( t \) is an element of \( C \), it strongly terminates, thus \( t' \) strongly terminates.

If moreover \( t' \rightarrow_\beta^* \lambda x : A \ t_1 \), then \( t \rightarrow_\beta^* \lambda x : A \ t_1 \), and for all \( u \in D \), and for all \( U \in S \), \( (u/x)t_1 \in U \). 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 5.5 (Applications) Let \( C \) be a candidate and \( S \) be a set of candidates, \( t_1 \) and \( t_2 \) such that \( t_1 \in ̂\Pi(C,S) \) and \( t_2 \in C \), and \( D \) be an element of \( S \). Then \( (t_1 \ t_2) \in D \).

Proof. As \( t_1 \in ̂\Pi(C,S) \) 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 5.1, 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 \( D \).
Proof. By induction on the structure of the term \((t_1, t_2)\). Let \(\phi \in C\), \(\Gamma \vdash \phi : A\). Then let \((t_1, t_2) = \phi\), \((t_1, t_2) \rightarrow^* \alpha : A\). The reduction sequence is finite, and if \((t_1, t_2) \rightarrow^* u_2 \rightarrow^* \alpha : A\). We have \(u_2 \in D\) and \(u_2 \rightarrow^* \alpha : A\). Thus, for all \(w\) in \(C\) and \(F\) in \(S\), \((w/x)u_2 \in F\). Therefore, \((t_1, t_2) \in \Pi(C,S) = D\).

5.2 Termination

Consider a super-consistent theory \(\Sigma, R\). We want to prove that \(\beta\)-reduction terminates in the \(\lambda\Pi\)-calculus modulo this theory. As this theory is super-consistent, it has a model \(M\) valued in the \(\Pi\)-algebra \(\langle C, \Sigma, \Gamma \\vdash \phi \rangle \). Consider this model.

If \(t\) has type \(B\) in some context \(\Gamma\), then \(B\) has type \(Type\) in \(\Gamma\), \(B\) has type \(Kind\) in \(\Gamma\), or \(B = Kind\). Thus, \([B]_\Sigma\) is an element of \(M_{Type} = C\), \([B]_\Sigma\) is an element of \(M_{Kind} = C\), or \([B]_\Sigma = \top\). In all these cases \([B]_\Sigma\) is a candidate.

Lemma 5.6 Let \(\Gamma\) be a context, \(\overline{\phi} = \phi_1, ..., \phi_n\) be a sequence of functions such that \(\phi_i\) maps \(x : A\) of \(\Gamma\) to an element of \(\Pi_{A, \phi_1, ..., \phi_{i-1}}\), \(\sigma\) be a substitution mapping every \(x : A\) of \(\Gamma\) to an element of \([A]_\Sigma\) and \(t\) a term of type \(B\) in \(\Gamma\). Then \(\sigma t \in [B]_\Sigma\).

Proof. By induction on the structure of the term \(t\).

- If \(t = Type\), then \(B = Kind\), \([B]_\Sigma = \top\) and \(\sigma t = Type \in [B]_\Sigma\).

- If \(t = x\) is a variable, then by definition of \(\sigma\), \(\sigma t \in [B]_\Sigma\).

- If \(t = \Pi x : C\ D\), then \(B = Type\) or \(B = Kind\), and \([B]_\Sigma = \top\), \(\Gamma \vdash C : Type\) and \(\Gamma, x : C \vdash D : Type\) or \(\Gamma, x : C \vdash D : Kind\), by induction hypothesis \(\sigma C \in [Type]_\Sigma = \top\), that is \(\sigma C\) strongly terminates and \(\sigma D \in [Type]_\Sigma = \top\) or \(\sigma D \in [Kind]_\Sigma = \top\), that is \(\sigma D\) strongly terminates. Thus, \(\sigma (\Pi x : C\ D) = \Pi x : C\ \sigma C\ \sigma D\) strongly terminates also and it is an element of \(\top = [B]_\Sigma\).

- If \(t = \lambda x : C\ u\) where \(u\) has type \(D\). Then \(B = \Pi x : C\ D\) and \([B]_\Sigma = \Pi x : C\ D\) = \(\Pi ([C]_\Sigma, \{D|_{\phi_{x=x}}| r \in D_C^n \times ... \times D_{C,\phi_x,...,\phi_{n-1}}\})\) is the set of terms \(s\) such that \(s\) strongly terminates and if \(s\) reduces to \(E\ s_1\) then for all \(s'\) in \([C]_\Sigma\) and all \(r\) in \(D_1^n \times ... \times D_{C,\phi_x,...,\phi_{n-1}}\), \((s'/x)s_1\) is an element of \([D]_{\phi_{x=x}}\).

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 \(\sigma C\) and \(\sigma u\). By induction hypothesis, the term \(\sigma C\) is an element of \([Type]_\Sigma = \top\) and the term \(\sigma u\) is an element of \([D]_\Sigma\), thus the reduction sequence is finite. Furthermore, every reduction of \(\sigma t\) has the form \(\lambda x : C'\ v\) where \(C'\) is a reduct of \(\sigma C\) and \(v\) is a reduct of \(\sigma u\). Let \(w\) be any term of \([C]_\Sigma\) and \(r\) be any element of \(D_1^n \times ... \times D_{C,\phi_x,...,\phi_{n-1}}\), the term \((w/x)\sigma u\) can be obtained by reduction from \(((w/x)\sigma)u\). By induction hypothesis, the term \(((w/x)\sigma)u\) is an element of \([D]_{\phi_{x=x}}\). Hence, by Lemma 5.4 the term \((w/x)\sigma u\) is an element of \([D]_{\phi_{x=x}}\). Therefore, the term \(\sigma t\) is an element of \([B]_\Sigma\).

- 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 \(\sigma u_1 \in [\Pi x : C\ D]_\Sigma = \Pi ([C]_\Sigma, \{D|_{\phi_{x=x}}| r \in D_1^n \times ... \times D_{C,\phi_x,...,\phi_{n-1}}\})\) and \(\sigma u_2 \in [C]_\Sigma\). By Lemma 5.5, \((\sigma u_1\ \sigma u_2) \in [D]_{\phi_{x=x}} = D_1^n \times ... \times D_{u_2,\phi_1,...,\phi_{n-1}} = [(u_2/x)D]_\Sigma = [B]_\Sigma\).
Theorem 5.1 Let $t$ be a term well-typed in a context $\Gamma$. Then $t$ strongly terminates.

Proof. Let $B$ be the type of $t$ in $\Gamma$, let $\vec{\phi} = \phi_1, \ldots, \phi_n$ be a sequence of functions such that $\phi_i$ maps $x : A$ of $\Gamma$ to an element of $D_{A,\phi_1,\ldots,\phi_{i-1}}$. $\sigma$ be the substitution mapping every $x : A$ of $\Gamma$ to itself. Note that, by Lemma 5.3, this variable is an element of $[A]_{\vec{\phi}}$. Then $t = \sigma t \in [B]_{\vec{\phi}}$. Hence it strongly terminates.

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

We finally prove the termination of the $\beta\mathcal{R}$-reduction for Simple type theory without or with a parametric quantifier and for the Calculus of constructions. The rules $\mathcal{R}$ of Simple type theory are

\[
\varepsilon (\Rightarrow x y) \rightarrow (\varepsilon x) \rightarrow (\varepsilon y)
\]

\[
\varepsilon (\forall_A x) \rightarrow \Pi z : A (\varepsilon (x z))
\]

This set $\mathcal{R}$ of rewrite rules terminates, as each reduction step reduces the number of symbols $\Rightarrow$ and $\forall_A$ in the term. Then, $\mathcal{R}$-reduction can create $\beta$-redices, but only $\beta$-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 $\beta$-reducing it first, then $\mathcal{R}$-reducing it, then $\beta$-reducing the trivial $\beta$-redices created by the $\mathcal{R}$-reduction.

A similar argument applies to Simple type theory with a parametric quantifier and to the Calculus of constructions.

Acknowledgements

The author wants to thank Frédéric Blanqui for very helpful remarks on a previous version of this paper.

References


6 Appendix: Super-consistency of the Calculus of constructions

Let $B = (\mathcal{B}, \bar{\mathcal{T}}, \bar{\Lambda}, \mathcal{P}^+ (B), \bar{I})$ be a full II-algebra. Let $\{e\}$ be an arbitrary one-element set. Let $U$ and $V$ defined as in Section 4.3.

Note that $U$ does not need to be closed by dependent function space. This can be compared with the fact that all terms that can be typed in the Calculus of constructions can be typed in the system $F\omega$.

**Definition 6.1** The interpretation function $N$ is defined as follows

- $N_{\text{type}} = N_{\text{Kind}} = V$,
- $N_{\Pi x:C} D$ is the set $F\langle N_C, N_D \rangle$, except if $N_D = \{e\}$, in which case $N_{\Pi x:C} D = \{e\}$,
- $N_{\text{type}} = U$,
- $N_\alpha = N_{\Pi \alpha} = N_{\Pi \tau} = N_{\Pi \kappa} = N_\eta = N_\varepsilon = \{e\}$,
- $N_\varepsilon = \{e\}$,
- $N_{\lambda x:C} t = N_t$,
- $N_{(t u)} = N_t$.

We first prove the two following lemmas.

**Lemma 6.1** If the term $t$ is an object, then

$$N_t = \{e\}$$

**Proof.** By induction on the structure of the term $t$. The term $t$ is neither $\text{Kind}$, $\text{Type}$, nor $\text{type}$. It is not a product. If it has the form $\lambda x : C t'$, then $t'$ is an object. If it has the form $(t' t'' \varepsilon)$, then $t'$ is an object.

**Lemma 6.2** If $u$ is an object, then

$$N_{(u/z)} t = N_t$$

**Proof.** By induction on the structure of the term $t$. If $t = x$ then, by Lemma 6.1

$$N_{(u/z)} x = N_u = \{e\} = N_t$$

If $t$ is $\text{Kind}$, $\text{Type}$, a constant, or a variable different from $x$, then $x$ does not occur in $t$. If it is a product, an abstraction, or an application, we use the induction hypothesis.

**Lemma 6.3** (Validity of the congruence) If $t \equiv_{\beta R} u$ then

$$N_t = N_u$$

**Proof.** If $t = (\lambda x : C t' \ v')$, then $u'$ is an object, then by Lemma 6.2

$$N_{(\lambda x : C t') \ v'} = N_{t'} = N_{(u'/z)}$$

Then, as for all $v$, $N_{(\eta v)} = N_\eta = \{e\}$, for all $w$, $N_{(w \ v)} = N_\varepsilon = \{e\}$, and if $N_D = \{e\}$, then $N_{\Pi x:C} D = \{e\}$, we have

$$N_{(\eta (\Pi_k \ C \ D))} = \{e\} = N_{\Pi x (\eta C) \ (\eta \ (D \ z))}$$
$$N_{(\varepsilon (\Pi \tau \ C \ D))} = \{e\} = N_{\Pi x (\varepsilon C) \ (\varepsilon \ (D \ z))}$$
$$N_{(\varepsilon (\Pi \kappa \ C \ D))} = \{e\} = N_{\Pi x (\varepsilon C) \ (\varepsilon \ (D \ z))}$$
$$N_{(\eta (\Pi \kappa \ C \ D))} = \{e\} = N_{\Pi x (\eta C) \ (\eta \ (D \ z))}$$

We prove, by induction on $t$, that if $t \rightarrow_{\beta R} u$ then $N_t = N_u$ and we conclude with a simple induction on the structure of a reduction of $t$ and $u$ to a common term.
Definition 6.2 The interpretation function $\mathcal{M}$ is defined as follows

- $\mathcal{M}_{\text{Kind}, \psi} = \mathcal{M}_{\text{Type}, \psi} = B$,
- $\mathcal{M}_{\Pi x: C D, \psi}$ is the set of functions $f$ mapping $\langle c', c \rangle$ in $N_C \times \mathcal{M}_{C, \psi}$ to an element of $\mathcal{M}_{D, (\psi, x = c')}$, except if for all $c'$ in $N_C$, $\mathcal{M}_{D, (\psi, x = c')} = \{e\}$, in which case $\mathcal{M}_{\Pi x: C D, \psi} = \{e\}$,
- $\mathcal{M}_{\text{Type}, \psi} = B$,
- $\mathcal{M}_{\eta, \psi}$ is the function of $F(U, V)$ mapping $S$ to $S$,
- $\mathcal{M}_{\epsilon, \psi}$ is the function of $F(\{e\}, V)$, mapping $e$ to $\{e\}$,
- $\mathcal{M}_{o, \psi} = B$,
- $\mathcal{M}_{\dot{\Pi} \text{KK}}$ is the function mapping $S$ in $U$ and $h$ in $F(\{e\}, U)$ to the set $F(\{e\} \times S, (h e))$, except if $(h e) = \{e\}$ in which case it maps $S$ and $h$ to $\{e\}$,
- $\mathcal{M}_{\dot{\Pi} \text{TT}} = e$,
- $\mathcal{M}_{\dot{\Pi} \text{KT}} = e$,
- $\mathcal{M}_{\dot{\Pi} \text{TK}}$ is the function mapping $e$ and $h$ in $F(\{e\}, U)$ to the set $F(\{e\} \times \{e\}, (h e))$, except if $(h e) = \{e\}$ in which case it maps $e$ and $h$ to $\{e\}$,
- $\mathcal{M}_{\lambda x: C t, \psi}$ is the function mapping $c$ in $N_C$ to $\mathcal{M}_{t, (\psi, x = c)}$, except if for all $c$ in $N_C$, $\mathcal{M}_{t, (\psi, x = c)} = e$ 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 $\mathcal{M}_{(t u), \psi} = e$.

Lemma 6.4 If $\Gamma \vdash C : \text{Type}$, then $N_C \in \mathcal{V}$

Proof. By induction on the structure of the term $C$. As this term has type $\text{Type}$, it is neither $\text{Kind}$ nor $\text{Type}$.

Lemma 6.5 (Well-typedness) If $\Gamma \vdash t : B$ and $\psi$ is a function mapping the variables $x : A$ of $\Gamma$ to elements of $N_A$, then $\mathcal{M}_{t, \psi} \in N_B$

Proof. We check each case of the definition of $\mathcal{M}$.

Lemma 6.6 (Substitution) For all $t$, $u$ and $\psi$ $\mathcal{M}_{(u/x)t, \psi} = \mathcal{M}_{t, (\psi, x = u)}$

Proof. By induction on the structure of the term $t$.

Lemma 6.7 (Validity of the congruence) If $t \equiv_{\beta R} u$ then $\mathcal{M}_{t, \psi} = \mathcal{M}_{u, \psi}$
Proof. If \( t = ((\lambda x : C \ t') \ u') \), then if for all \( c \in N_C \) \( M_{t',(\psi,x=c)} = e \) then
\[
M_{((\lambda x:C \ t') \ u'),\psi} = e = M_{t',(\psi,x=M_{t',\psi})} = M_{(u'/x)\psi}
\]
Otherwise
\[
M_{((\lambda x:C \ t') \ u'),\psi} = M_{t,(\psi,x=M_{t',\psi})} = M_{(u/x)\psi}
\]
The set \( M_{(\eta (\Pi_{KK} C D)),\psi} \) is the set \( \mathcal{F}(\{e\} \times M_{C,\psi},(M_{D,\psi} e)) \), except if \( (M_{D,\psi} e) = \{e\} \) in which case \( M_{(\eta (\Pi_{KK} C D)),\psi} = \{e\} \). The set \( M_{\Pi x(\eta C) (\eta (D x)),\psi} \) is this same set. Thus
\[
M_{(\eta (\Pi_{KK} C D)),\psi} = M_{\Pi x(\eta C) (\eta (D x)),\psi}
\]
We have
\[
M_{(\varepsilon (\Pi_{TT} C D))} = \{e\} = M_{\Pi x(\varepsilon C) (\varepsilon (D x))}
\]
and
\[
M_{(\varepsilon (\Pi_{KT} C D))} = \{e\} = M_{\Pi x(\varepsilon C) (\varepsilon (D x))}
\]
The set \( M_{(\eta (\Pi_{TT} C D)),\psi} \) is the set \( \mathcal{F}(\{e\} \times \{e\},(M_{D,\psi} e)) \), except if \( (M_{D,\psi} e) = \{e\} \) in which case \( M_{(\eta (\Pi_{TT} C D)),\psi} = \{e\} \). The set \( M_{\Pi x(\varepsilon C) (\eta (D x)),\psi} \) is this same set. Thus
\[
M_{(\eta (\Pi_{TT} C D)),\psi} = M_{\Pi x(\varepsilon C) (\eta (D x)),\psi}
\]
We prove, by induction on \( t \), that if \( t \rightarrow^1_{\beta\eta} u \) then \( M_{t,\psi} = M_{u,\psi} \) and we conclude with a simple induction on the structure of a reduction of \( t \) and \( u \) to a common term.

Definition 6.3 The interpretation function \([_\_]\) is defined as follows

- \([\text{Kind}]_{\psi,\phi} = [\text{Type}]_{\psi,\phi} = \top\),
- \([\Pi x : C \ D]_{\psi,\phi} = \Pi([C]_{\psi,\phi}, \{(D)_{(\psi,x=c), (\phi,x=c)} | c \in N_C, c \in M_{C,\psi}\}),\]
- \([\text{type}]_{\psi,\phi} = \top\),
- \([\varnothing]_{\psi,\phi} = \top\),
- \([\Pi_{KK}]_{\psi,\phi} \) is the function mapping \( \langle S, a \rangle \) in \( U \times B \), \( \langle f, g \rangle \) in \( \mathcal{F}(\{e\}, U) \times \mathcal{F}(\{e\} \times S, B) \) to \( \Pi(a, \{(g (e, s), s) | s \in S\})\),
- \([\Pi_{TT}]_{\psi,\phi} \) is the function mapping \( \langle e, a \rangle \) to \( \{e\} \times B \), and \( \langle e, g \rangle \) in \( \{e\} \times \mathcal{F}(\{e\} \times \{e\}, B) \) to \( \Pi(a, \{(g (e, g), g (e, e))\}),\]
- \([\Pi_{KT}]_{\psi,\phi} \) is the function mapping \( \langle S, a \rangle \) in \( U \times B \), and \( \langle e, g \rangle \) in \( \{e\} \times \mathcal{F}(\{e\} \times S, B) \) to \( \Pi(a, \{(g (e, s), s) | s \in S\})\),
- \([\eta]_{\psi,\phi} \) is the function from \( U \times B \) to \( B \), mapping \( \langle S, a \rangle \) to \( a\),
- \([\varepsilon]_{\psi,\phi} \) is the function from \( \{e\} \times B \) to \( B \), mapping \( \langle e, a \rangle \) to \( a\),
- \([x]_{\psi,\phi} = \phi x\),
- \([\lambda x : C \ t]_{\psi,\phi} \) is the function mapping \( \langle c', e \rangle \) in \( N_C \times M_{C,\psi} \) to \( \Pi(\langle c', e \rangle, \langle c, e \rangle)\), except if for all \( \langle c', e \rangle \) in \( N_C \times M_{C,\psi} \), \( \Pi(\langle c', e \rangle, \langle c, e \rangle) = e \), in which case \( \Pi(\langle c', e \rangle, \langle c, e \rangle) = e\),
- \([t u]_{\psi,\phi} = [t]_{\psi,\phi} \langle M_{u,\psi}, [u]_{\psi,\phi}\rangle, \) except if \( [t]_{\psi,\phi} = e \), in which case \( [t u]_{\psi,\phi} = e\).
Lemma 6.8 (Well-typedness) If $\Gamma \vdash t : B$, $\psi$ is a function mapping variables $x : A$ of $\Gamma$ to elements of $N_A$, and $\phi$ is a function mapping variables $x : A$ of $\Gamma$ to elements of $M_{A, \psi}$, then $$[t]_{\psi, \phi} \in M_{B, \psi}$$

Proof. We check each case of the definition of $[\cdot]$.

Lemma 6.9 (Substitution) For all $t$, $u$, $\psi$, and $\phi$
$$[(u/x)t]_{\psi, \phi} = [t]_{(\psi, x = u, \phi, x = [u]_{\psi, \phi})}$$

Proof. By induction on the structure of the term $t$.

Lemma 6.10 (Validity of the congruence) If $t \equiv_{\beta R} u$ then
$$[t]_{\psi, \phi} = [u]_{\psi, \phi}$$

Proof. If $t = ((\lambda x : C t') u')$, then if for all $c'$ in $N_C$ and $c$ in $M_{C, \psi}$, we have $[t']_{(\psi, x = c'), (\phi, x = c)} = e$ then
$$[(\lambda x : C t') u']_{\psi, \phi} = [t']_{(\psi, x = u', \phi, x = [u']_{\psi, \phi})} = [(u'/x)t']_{\psi, \phi}$$

Otherwise
$$[(\lambda x : C t') u']_{\psi, \phi} = [t']_{(\psi, x = u', \phi, x = [u']_{\psi, \phi})} = [(u'/x)t']_{\psi, \phi}$$

We have
$$[(\eta (\Pi_{KK} C D))]_{\psi, \phi} = \tilde{\Pi}((\psi, \phi), (\{[D]_{\psi, \phi} (\epsilon, s) | s \in M_{C, \psi}\})) = \Pi y : (\eta C) (\eta (D y))_{\psi, \phi}$$

$$[(\epsilon (\Pi_{TT} C D))]_{\psi, \phi} = \tilde{\Pi}((\psi, \phi), (\{[D]_{\psi, \phi} (\epsilon, s) | s \in M_{C, \psi}\})) = \Pi y : (\epsilon C) (\epsilon (D y))_{\psi, \phi}$$

$$[(\epsilon (\Pi_{KT} C D))]_{\psi, \phi} = \tilde{\Pi}((\psi, \phi), (\{[D]_{\psi, \phi} (\epsilon, s) | s \in M_{C, \psi}\})) = \Pi y : (\epsilon C) (\epsilon (D y))_{\psi, \phi}$$

We prove, by induction on $t$, that if $t \rightarrow_{\beta R} u$ then $[t]_{\psi, \phi} = [u]_{\psi, \phi}$ and we conclude with a simple induction on the structure of a reduction of $t$ and $u$ to a common term.