A revised completeness result for the simply typed $\lambda\mu$-calculus using realizability semantics

Karim Nour, Mohamad Ziadeh

A revised completeness result for the simply typed $\lambda\mu$-calculus using realizability semantics

Karim NOUR & **Mohamad ZIADEH**
LAMA - Équipe LIMD
Université Savoie Mont Blanc
73376 Le Bourget du Lac, France
email : karim.nour@univ-smb.fr

Abstract

In this paper, we correct some errors in [21]. We define a new realizability semantics for the simply typed $\lambda\mu$-calculus. We show that if a term is typable, then it inhabits the interpretation of its type. We also prove a completeness result of our realizability semantics using a particular term model.

1 Introduction

Since it was realised that the Curry-Howard isomorphism can be extended to the case of classical logic (cf. [16] and [7]), several calculi have appeared aiming to give an encoding of proofs formulated either in classical natural deduction or in classical sequent calculus. One of them was the $\lambda\mu$-calculus presented by Parigot in [22], which stands very close in nature to the λ-calculus itself. It uses new kinds of variables, the μ-variables, not active at the moment, but to which the current continuation can be passed over. Eliminating cuts with these new formulas leads to the introduction of a new reduction rule : the so-called μ-rule. The result is a calculus, the $\lambda\mu$-calculus [23], which is in relation with classical natural deduction. In addition, more simplification rules, for example the ρ- and θ-rules, are defined by Parigot [23, 24]). Parigot showed that the $\lambda\mu$-calculus is strongly normalizing in [25], he gave a proof of the result with the help of the Tait-Girard reducibility method [6, 30]. An arithmetical proof of the same result was presented by David and Nour in [2].

The idea of the realizability semantics is to associate to each type a set of terms which realize this type. Under this semantics, an atomic type is interpreted as a set of λ-terms saturated by a certain relation. Then, an arrow type receives the intuitive interpretation of a functional space. For example, a term which realizes the type $\mathbb{N} \to \mathbb{N}$ is a function from $\mathbb{N}$ to $\mathbb{N}$. Realizability semantics has been a powerful method for establishing the strong normalization of type systems à la Tait and Girard [30, 6]. The realizability of a type system enables one to also show the soundness of the system in the sense that the interpretation of a type contains all the terms that have this type. Soundness has been an important method for characterizing the computational behaviour of typed terms through their types as has been illuminative in the work of Krivine.

It is also interesting to find the class of types for which the converse of soundness holds i.e., to find the types A for which the realizability interpretation contains exactly, in a certain sense, the terms typable by A. This property is called completeness and has not yet been studied for every type system.

Hindley [9, 10, 11] was the first to study the completeness of a simple type system and he showed that all the types of that system have the completeness property. Then, he generalised his completeness proof for an intersection type system [8]. [15] has established completeness for a class of types in Girard-Reynolds's system

F known as the strictly positive types. [4, 5] generalised the result of [15] for the larger class which includes all the positive types and also for the second order functional arithmetic. [1] established by a different method using Kripke models the completeness for the simply typed λ-calculus. Finally [12] established the soundness and completeness for a strict non linear intersection type system with a universal type.

In [19], Nour and Saber adapted Parigot's method and established a short semantical proof of the strong normalization of classical natural deduction with disjunction as primitive. In general all the known semantical proofs of strong normalization use a variant of the reducibility candidates based on a correctness result, which has been important also for characterizing computational behavior of some typed terms, as it was done in J.-L. Krivine's works [14]. This inspired K. Nour and K. Saber to define a general semantics for classical natural deduction in [20] and gave such characterizations. In [21], Nour and Saber proposed a realizability semantics for the simply typed $\lambda\mu$-calculus and proved a completeness theorem. This semantics is inspired by the strong normalization proof of Parigot's $\lambda\mu$-calculus, which consists in rewriting each reducibility candidate as a double orthogonal. In [29], van Bakel, Barbanera and de'Liguoro prove the completeness of the $\lambda\mu$-calculus with intersection types using filter models.

The semantics proposed in [21] has several defaults. First of all, there is a mistake in the proof of the correctness lemma (the case of the typing rule ($\bot$) in the proof of Lemma 3.3) which is not correctable. This comes essentially from the permission to have μ-variables in the sequence of terms by adopting Saurin's interpretation. This mistake makes the semantics less interesting even though the proof of completeness remains correct. Besides, although the statement of Lemma 4.2 of [21] is correct, the proposed proof is false. The correction of these mistakes, mainly the first, needs the introduction of another realizability semantics which is completely different from the one proposed in [21].

In the present work we provide another realizability semantics for the simply typed $\lambda\mu$-calculus, by changing the concept of saturation. For this, we add an important and an indispensable modification to the notion of μ-saturation using more bottom sets. The saturation conditions give a very satisfactory correctness result. The completeness model gives also more intuition about the models that we consider.

This paper is organized as follows. Section 2 is an introduction to the simply typed $\lambda\mu$-calculus. We also define the semantics and prove its correctness. Section 3 is devoted to the completeness result.

2 The simply typed $\lambda\mu$-calculus

2.1 The syntax of the system

In this paper, we use the $\lambda\mu$-calculus à la de Groote which is more expressive than Parigot's original version. In [3], de Groote has proposed a new version of the $\lambda\mu$-calculus by modifying its syntax. Namely, in the construction of terms the distinction between named and unnamed terms has disappeared and the term forming rules has became more flexible: a μ-operator can be followed now by any kind of term (in the untyped version), not necessarily by a term beginning with a μ -variable.

Definition 2.1 (Terms and reductions)

1. *Let $\mathcal{V}$ and $\mathcal{A}$ be two infinite sets of disjoint alphabets for distinguishing λ-variables and μ-variables. The $\lambda\mu$-terms are given by the following grammar:*

$$\mathcal{T} := \mathcal{V} \mid \lambda\mathcal{V}.\mathcal{T} \mid (\mathcal{T}\ \mathcal{T}) \mid \mu\mathcal{A}.\mathcal{T} \mid (\mathcal{A}\ \mathcal{T})$$

2. *The complexity of a term is defined by* $c(x) = 0$, $c(\lambda x.t) = c(\mu\alpha.t) = c((\alpha\ t)) = c(t) + 1$ *and* $c((u\ v)) = c(u) + c(v) + 1$.

3. *The basic reduction rules are* $\triangleright_\beta$ *and* $\triangleright_\mu$.

   - $(\lambda x.u\ v) \triangleright_\beta u[x := v]$
   - $(\mu\alpha.u\ v) \triangleright_\mu \mu\alpha.u[\alpha :=^* v]$
     where $u[\alpha :=^* v]$ *is obtained from* u *by replacing inductively each subterm in the form* $(\alpha\ w)$ *in* u *by* $(\alpha\ (w\ v))$.

4. *Let* $\triangleright$ *stand for the compatible closure of the union of the rules given above, and, as usual,* $\triangleright^*$ *denotes the reflexive transitive closure of* $\triangleright$, *and* $\simeq$ *the equivalence relation induced by* $\triangleright^*$. *We also denote by* $t \triangleright^n t'$ *that* $t \triangleright^* t'$ *and the length of this reduction (number of reduced redexes) is* n. *A term is said to be normal if it does not contain a redex. A term* t *is called strongly normalizable, if there is no infinite reduction sequence starting from* t.

We find in the current literature of the $\lambda\mu$-calculus other simplification rules such as: $\triangleright_\theta$, $\triangleright_\rho$, $\triangleright_\nu$, $\triangleright_\eta$, $\triangleright_{\mu'}$, ... These rules allow to get less normal forms (see [17, 23, 25, 26]). In this paper, we do not need these rules for our completeness result.

Definition 2.2 (Types and typing rules)

1. *Types are formulas of the propositional logic built from the infinite set of propositional variables* $\mathcal{P} = \{X, Y, Z, \dots\}$ *and a constant of type* $\bot$, *using the connective* $\to$.

2. *The complexity of a type is defined by* $c(\bot) = c(X) = 0$ *and* $c(A \to B) = c(A) + c(B) + 1$.

3. *Let* $A_1, A_2, \dots, A_n, A$ *be types, we denote the type* $A_1 \to (\cdots \to (A_n \to A)\dots)$ *by* $A_1, \dots, A_n \to A$.

4. *Proofs are presented in natural deduction system with several conclusions, such that formulas in the left-hand-side of* $\vdash$ *are indexed by* λ*-variables and those in right-hand-side of* $\vdash$ *are indexed by* μ*-variables, except one which is indexed by a term. Let* t *be a* $\lambda\mu$*-term,* A *a type,* $\Gamma = \{x_i : A_i\}_{1\le i\le n}$ *and* $\Delta = \{\alpha_j : B_j\}_{1\le j\le m}$ *contexts, using the following rules, we will define "*t *typed with type* A *in the contexts* Γ *and* Δ*" and we denote it* $\Gamma \vdash t : A\ ;\Delta$.

$$\frac{}{\Gamma \vdash x_i : A_i\ ;\ \Delta}ax \qquad 1 \le i \le n$$

$$\frac{\Gamma, x : A \vdash t : B; \Delta}{\Gamma \vdash \lambda x.t : A \to B; \Delta}\ \to_i \qquad \frac{\Gamma \vdash u : A \to B; \Delta \quad \Gamma \vdash v : A; \Delta}{\Gamma \vdash (u\ v) : B; \Delta}\ \to_e$$

$$\frac{\Gamma \vdash t : \bot; \Delta, \alpha : A}{\Gamma \vdash \mu\alpha.t : A; \Delta}\ \mu \qquad \frac{\Gamma \vdash t : A; \Delta, \alpha : A}{\Gamma \vdash (\alpha\ t) : \bot; \Delta, \alpha : A}\ \bot$$

We denote this typed system by S_μ.

We have the following results (for more details, see [2, 25, 26]).

Theorem 2.1 (Confluence result) *If $t \triangleright^* t_1$ and $t \triangleright^* t_2$, then there exists t_3 such that $t_1 \triangleright^* t_3$ and $t_2 \triangleright^* t_3$.*

Theorem 2.2 (Subject reduction) *If $\Gamma \vdash t : A; \Delta$ and $t \triangleright^* t'$ then $\Gamma \vdash t' : A; \Delta$.*

Theorem 2.3 (Strong normalization) *If $\Gamma \vdash t : A; \Delta$, then t is strongly normalizable.*

We need some specific definitions and notations.

Definition 2.3

1. *Let t be a term. The term $\vec{\lambda\mu}.t$ denotes the term t preceded by a sequence of λ and μ abstractions.*
2. *Let t be a term and $\bar{v}$ a finite sequence of terms (the empty sequence is denoted by $\emptyset$). The term $(t\,\bar{v})$ is defined by $(t\,\emptyset) = t$ and $(t\,u\bar{u}) = ((t\,u)\,\bar{u})$.*
3. *Let us recall that a term t either has a head redex (i.e. $t = \vec{\lambda\mu}.(u\,\bar{v})$ where u is a redex called the head redex), or is in head normal form (i.e. $t = \vec{\lambda\mu}.(x\,\bar{v})$ or $t = \vec{\lambda\mu}.(\alpha\,u\,\bar{v})$ where x and α are variables called the head variable).*
4. *The leftmost reduction (denoted by $\triangleright_l$) consits in reducing the redex nearest to the left of the term. We can also see it as an iteration of the head reduction: once we find the head normal form, we reduce the arguments of the head variable.*

Lemma 2.1 *The leftmost reduction of a normalizing term terminates.*

Proof See [26]. □

The previous lemma allows one to define a notion of length for normalizing terms.

Definition 2.4 *Let t be a normalizing term. We define $l(t)$ as the number of leftmost reductions needed to find the normal form of t.*

We need to define the concept of simultaneous substitution to be able to present the correctness lemma.

Definition 2.5 *Let $t, u_1, \ldots, u_n$ be terms, $\bar{v}_1, \ldots, \bar{v}_m$ finite sequences of terms, and σ the simultaneous substitution $[(x_i := u_i)_{1\leq i\leq n}; (\alpha_j :=^* \bar{v}_j)_{1\leq j\leq m}]$ which is not an object of the syntax. Then $t\sigma$ is obtained from the term t by replacing each x_i by u_i and replacing inductively each subterm of the form $(\alpha_j\,u)$ in t by $(\alpha_j(u\,\bar{v}_j))$.*

Lemma 2.2 *If $t \triangleright^* t'$ and σ a simultaneous substitution, then $t\sigma \triangleright^* t'\sigma$.*

Proof It suffies to check the property for one step of reduction. Then we proceed by induction on $c(t)$. □

2.2 The semantics of the system

In this section, we define the realizability semantics and we prove its correctness lemma. We begin by the definition of the saturation of sets of terms and then the operation $\rightsquigarrow$ between these sets which will serve to interpret the arrow on types.

Definition 2.6

1. *We say that a set of terms $\mathcal{S}$ is saturated if for all terms u and v, if $v \triangleright^* u$ and $u \in \mathcal{S}$, then $v \in \mathcal{S}$.*

2. *Consider two sets of terms $\mathcal{K}$ and $\mathcal{L}$, we define a new set of terms*

$$\mathcal{K} \rightsquigarrow \mathcal{L} = \{t \in \mathcal{T} \ / \ \forall\, u \in \mathcal{K} \ ; \ (t\ u) \in \mathcal{L}\}.$$

3. *We denote $\mathcal{T}^{<\omega}$ as the set of finite sequences of terms. Let $\mathcal{L}$ be a set of terms and $\mathcal{X} \subseteq \mathcal{T}^{<\omega}$, then we define a new set of terms*

$$\mathcal{X} \rightsquigarrow \mathcal{L} = \{t \in \mathcal{T} \ / \ \forall\, \bar{u} \in \mathcal{X} \ ; \ (t\ \bar{u}) \in \mathcal{L}\}.$$

Lemma 2.3 *If $\mathcal{L}$ is a saturated set and $\mathcal{X} \subseteq \mathcal{T}^{<\omega}$, then $\mathcal{X} \rightsquigarrow \mathcal{L}$ is also a saturated one.*

Proof Let u and v be terms such that $v \rhd^* u$ and $u \in \mathcal{X} \rightsquigarrow \mathcal{L}$. Then $\forall\, \bar{w} \in \mathcal{X}$, $(u\ \bar{w}) \in \mathcal{L}$ and $(v\ \bar{w}) \rhd^* (u\ \bar{w})$. Since $\mathcal{L}$ is a saturated set, we have $\forall\, \bar{w} \in \mathcal{X}$, $(v\ \bar{w}) \in \mathcal{L}$, thus $v \in \mathcal{X} \rightsquigarrow \mathcal{L}$. □

Now, we are going to define the realizability model for the system S_μ. For that, we need many bottom sets $(\perp\!\!\!\perp_i)_{i\in I}$ including a particular one denoted by $\perp\!\!\!\perp_0$. We also need several sets of μ-variables $(\mathcal{C}_i)_{i\in I}$ which allow to pass from bottoms to $\perp\!\!\!\perp_0$ and vice versa. We also allow that the models have particular sets $(R_j)_{j\in J}$ satisfying some properties. In order to obtain the completeness theorem, it is possible to take from the beginning $I = \mathbb{N}$ without considering the sets $(R_j)_{j\in J}$. However, the flexibility to have $I \subseteq \mathbb{N}$ and some sets $(R_j)_{j\in J}$ will allow to have more models. This will also allow the use of the correctness lemma in order to study the computational behaviour of some typed terms (cf. the example given at the end of this section, Theorem 2.4).

Definition 2.7 1. *A model $\mathcal{M}$ of S_μ is defined by giving three subsets $\langle(\mathcal{C}_i)_{i\in I}, (\perp\!\!\!\perp_i)_{i\in I}, (R_j)_{j\in J}\rangle$ where :*

- *I, J are subsets of $\mathbb{N}$ such that $0 \in I$,*
- *$(\mathcal{C}_i)_{i\in I}$ a sequence of disjoint infinite subsets of μ-variables,*
- *$(\perp\!\!\!\perp_i)_{i\in I}$ and $(R_j)_{j\in J}$ sequences of non-empty saturated subsets of terms*

such that

- *$\forall\, i \in I$, if $\alpha_i \in \mathcal{C}_i$ and $u \in \perp\!\!\!\perp_0$ then $\mu\alpha_i.u \in \perp\!\!\!\perp_i$ (i.e. if, for some $\alpha_i \in \mathcal{C}_i$, $u[\alpha := \alpha_i] \in \perp\!\!\!\perp_0$, then $\mu\alpha.u \in \perp\!\!\!\perp_i$),*
- *$\forall\, i \in I$, if $\alpha_i \in \mathcal{C}_i$ and $u_i \in \perp\!\!\!\perp_i$, then $(\alpha_i\, u_i) \in \perp\!\!\!\perp_0$,*
- *$\forall\, j \in J$, $\exists\, i \in I$, $\exists\, \mathcal{X}_j \subseteq \mathcal{T}^{<\omega}$, such that $R_j = \mathcal{X}_j \rightsquigarrow \perp\!\!\!\perp_i$.*

2. *If $\mathcal{M} = \langle(\mathcal{C}_i)_{i\in I}, (\perp\!\!\!\perp_i)_{i\in I}, (R_j)_{j\in J}\rangle$ is a model of S_μ, we denote by $|\mathcal{M}|$ the smallest set containing the sets $\perp\!\!\!\perp_i$ and R_j and closed under the constructor $\rightsquigarrow$.*

We will see further that we did not need to get the subsets $(R_j)_{j\in J}$ to have the correcteness lemma. We allow a model to have such sets to enrich the concept of model and give the possibility of interpreting the types with more options.

Now we are going to prove that every element of a realizability model can be written as the orthogonal of a set of sequence terms. This property is essential to interpret the μ-variables and announce the generalized correctness lemma (Lemma 2.6). The difficulty here with respect to the semantics proposed in [19, 20, 21] is the presence of several bottoms. As for the orthogonal of an element of a model, we will choose the convenient bottom which has the smallest index (Lemma 2.5).

Lemma 2.4 *Let $\mathcal{M} = \langle(\mathcal{C}_i)_{i\in I}, (\perp\!\!\!\perp_i)_{i\in I}, (R_j)_{j\in J}\rangle$ be a model and $\mathcal{G} \in |\mathcal{M}|$.*
There exists a set $\mathcal{X}_\mathcal{G} \subseteq \mathcal{T}^{<\omega}$ and $i \in I$ such that $\mathcal{G} = \mathcal{X}_\mathcal{G} \rightsquigarrow \perp\!\!\!\perp_i$.

Proof By induction on $\mathcal{G}$.

- If $\mathcal{G} = \perp\!\!\!\perp_i$, take $\mathcal{X}_{\mathcal{G}} = \{\phi\}$.
- If $\mathcal{G} = R_j$, then, by definition of R_j, $\mathcal{G} = \mathcal{X}_j \rightsquigarrow \perp\!\!\!\perp_i$ and we take $\mathcal{X}_{\mathcal{G}} = \mathcal{X}_j$.
- If $\mathcal{G} = \mathcal{G}_1 \rightsquigarrow \mathcal{G}_2$, then, by induction hypothesis, $\mathcal{G}_2 = \mathcal{X}_{\mathcal{G}_2} \rightsquigarrow \perp\!\!\!\perp_i$ where $\mathcal{X}_{\mathcal{G}_2} \subseteq \mathcal{T}^{<\omega}$, and take $\mathcal{X}_{\mathcal{G}} = \{u\bar{v} \ / \ u \in \mathcal{G}_1 \text{ and } \bar{v} \in \mathcal{X}_{\mathcal{G}_2}\}$.

□

Definition 2.8 *If $\mathcal{M} = \langle (\mathcal{C}_i)_{i\in I}, (\perp\!\!\!\perp_i)_{i\in I}, (R_j)_{j\in J}\rangle$ is a model and $\mathcal{G} \in |\mathcal{M}|$, let*

- *$w(\mathcal{G})$ the smallest integer i such that $\mathcal{G} = \mathcal{X} \rightsquigarrow \perp\!\!\!\perp_i$ for some $\mathcal{X} \subseteq \mathcal{T}^{<\omega}$,*
- *$\mathcal{G}^{\perp} = \bigcup\{\mathcal{X} \subseteq \mathcal{T}^{<\omega} \ / \ \mathcal{G} = \mathcal{X} \rightsquigarrow \perp\!\!\!\perp_{w(\mathcal{G})}\}$.*

Lemma 2.5 *Let $\mathcal{M} = \langle (\mathcal{C}_i)_{i\in I}, (\perp\!\!\!\perp_i)_{i\in I}, (R_j)_{j\in J}\rangle$ be a model and $\mathcal{G} \in |\mathcal{M}|$, then $\mathcal{G} = \mathcal{G}^{\perp} \rightsquigarrow \perp\!\!\!\perp_{w(\mathcal{G})}$.*

Proof

$\subseteq$: Let $t \in \mathcal{G}$. If $\bar{u} \in \mathcal{G}^{\perp}$, then $\bar{u} \in \mathcal{X} \subseteq \mathcal{T}^{<\omega}$ and $\mathcal{G} = \mathcal{X} \rightsquigarrow \perp\!\!\!\perp_{w(\mathcal{G})}$, thus $(t\ \bar{u}) \in \perp\!\!\!\perp_{w(\mathcal{G})}$. Therefore $t \in \mathcal{G}^{\perp} \rightsquigarrow \perp\!\!\!\perp_{w(\mathcal{G})}$.

$\supseteq$: By Lemma 2.4, we have $\mathcal{G} = \mathcal{X} \rightsquigarrow \perp\!\!\!\perp_{w(\mathcal{G})}$ for some $\mathcal{X} \subseteq \mathcal{T}^{<\omega}$, then $\mathcal{X} \subseteq \mathcal{G}^{\perp}$. Let $t \in \mathcal{G}^{\perp} \rightsquigarrow \perp\!\!\!\perp_{w(\mathcal{G})}$. We have $\forall\, \bar{u} \in \mathcal{X}$, $\bar{u} \in \mathcal{G}^{\perp}$, then $(t\ \bar{u}) \in \perp\!\!\!\perp_{w(\mathcal{G})}$. Therefore $t \in \mathcal{G}$.

□

We can now interpret the types in a model and also give the definition of the general interpretation of a type.

Definition 2.9 *1. Let $\mathcal{M} = \langle (\mathcal{C}_i)_{i\in I}, (\perp\!\!\!\perp_i)_{i\in I}, (R_j)_{j\in J}\rangle$ be a model.*

(a) An $\mathcal{M}$-interpretation $\mathcal{I}$ is a function $X \mapsto \mathcal{I}(X)$ from the set of propositional variables $\mathcal{P}$ to $|\mathcal{M}|$ which we extend for any formula as follows: $\mathcal{I}(\perp) = \perp\!\!\!\perp_0$ and $\mathcal{I}(A \to B) = \mathcal{I}(A) \rightsquigarrow \mathcal{I}(B)$.

(b) For any type A, we denote $|A|_{\mathcal{M}} = \bigcap\{\mathcal{I}(A) \ / \ \mathcal{I}$ an $\mathcal{M}$-interpretation$\}$ the interpretation of A in the model $\mathcal{M}$.

2. For any type A, we denote $|A| = \bigcap\{|A|_{\mathcal{M}} \ / \ \mathcal{M}$ a model$\}$ the general interpretation of A .

The correctness lemma is sort of a validation of the notion of models that we considered. It states that a term of a type A is within the general interpretation of A. As for the semantics defined in [21], the proof of the correctness lemma is false and the mistake is difficult to find. It is the case of the typing rule ($\perp$) which is problematic. The mistake comes from the permission to have μ-variables in the sequence of terms by adopting Saurin's interpretation.

Lemma 2.6 (General correctness lemma)
Let $\mathcal{M} = \langle (\mathcal{C}_i)_{i\in I}, (\perp\!\!\!\perp_i)_{i\in I}, (R_j)_{j\in J}\rangle$ be a model, $\mathcal{I}$ an $\mathcal{M}$-interpretation,
$\Gamma = \{x_k : A_k\}_{1\leq k\leq n}$, $\Delta = \{\alpha_r : B_r\}_{1\leq r\leq m}$ such that $\alpha_r \in \mathcal{C}_{w(\mathcal{I}(B_r))}$ $(1 \leq r \leq m)$,
$u_k \in \mathcal{I}(A_k)$ $(1 \leq k \leq n)$, $\bar{v_r} \in (\mathcal{I}(B_r))^{\perp}$ $(1 \leq r \leq m)$) and
$\sigma = [(x_k := u_k)_{1\leq k\leq n}; (\alpha_r :=^* \bar{v}_r)_{1\leq r\leq m}]$. If $\Gamma \vdash u : A \ ; \Delta$, then $u\sigma \in \mathcal{I}(A)$.

Proof By induction on the derivation, we consider the last rule used.

ax: In this case $u = x_k$ and $A = A_k$, then $u\sigma = u_k \in \mathcal{I}(A)$.

$\to_i$: In this case $u = \lambda x.v$ and $A = B \to C$ such that $\Gamma, x : B \vdash v : C\ ;\Delta$. Let $w \in \mathcal{I}(B)$ and $\delta = \sigma + [x := w]$, by induction hypothesis, $v\delta \in \mathcal{I}(C)$. Since $(\lambda x.v\sigma\ w) \triangleright^* v\delta$, then $(\lambda x.v\sigma\ w) \in \mathcal{I}(C)$. Therefore $\lambda x.v\sigma \in \mathcal{I}(B) \rightsquigarrow \mathcal{I}(C)$ and finally $u\sigma \in \mathcal{I}(A)$.

$\to_e$: In this case $u = (u_1\, u_2)$, $\Gamma \vdash u_1 : B \to A\ ;\Delta$ and $\Gamma \vdash u_2 : B\ ;\Delta$. By induction hypothesis, $u_1\sigma \in \mathcal{I}(B) \rightsquigarrow \mathcal{I}(A)$ and $u_2\sigma \in \mathcal{I}(B)$, therefore $(u_1\sigma\, u_2\sigma) \in \mathcal{I}(A)$, i.e. $u\sigma \in \mathcal{I}(A)$.

μ: In this case $u = \mu\alpha.v$ and $\Gamma \vdash v :\bot\ ; \alpha : A, \Delta$. We can assume that α is a new variable which belongs to $\mathcal{C}_{w(\mathcal{I}(A))}$ (there is always such a variable because the sets $\mathcal{C}_i$ are infinite). Let $\bar{v} \in (\mathcal{I}(A))^\perp$ and $\delta = \sigma + [\alpha :=^* \bar{v}]$. By induction hypothesis, $v\delta \in \perp\!\!\!\perp_0$, then $\mu\alpha.v\delta \in \perp\!\!\!\perp_{w(\mathcal{I}(A))}$. Since $(\mu\alpha.v\sigma\ \bar{v}) \triangleright^* \mu\alpha.v\delta$, then, $(\mu\alpha.v\sigma\ \bar{v}) \in \perp\!\!\!\perp_{w(\mathcal{I}(A))}$. We deduce that for all $\bar{v} \in (\mathcal{I}(A))^\perp$, $(\mu\alpha.v\sigma\ \bar{v}) \in \perp\!\!\!\perp_{w(\mathcal{I}(A))}$, therefore $\mu\alpha.v\sigma \in \mathcal{I}(A)$, i.e. $u\sigma \in \mathcal{I}(A)$.

$\bot$: In this case $u = (\alpha_r\ v)$, $A =\bot$ and $\Gamma \vdash v : B_r\ ;\Delta$. By induction hypothesis, $v\sigma \in \mathcal{I}(B_r)$, hence $(v\sigma\ \bar{v_r}) \in \perp\!\!\!\perp_{w(\mathcal{I}(B_r))}$, therefore $(\alpha_r\ (v\sigma\ \bar{v_r})) \in \perp\!\!\!\perp_0$, i.e. $u\sigma \in \mathcal{I}(A)$.

□

We can now state and prove the correctness lemma.

Corollary 2.1 (Correctness lemma) *Let A be a type and t a closed term. If $\vdash t : A$, then, $t \in |A|$.*

Proof Let $\mathcal{M}$ be a model and $\mathcal{I}$ an $\mathcal{M}$-interpretation. Since $\vdash t : A$, then, by the general correctness lemma, $t \in \mathcal{I}(A)$. This is true for any model $\mathcal{M}$ and for any $\mathcal{M}$-interpretation $\mathcal{I}$, therefore $t \in |A|$. □

According to the cases $\to_i$ and μ of the proof of Lemma 2.6, we observe that the saturation conditions can be weakened as follows. We do not need the saturation by anti-reduction but only the saturation by weak-head anti-reduction.

A set of terms $\mathcal{S}$ is saturated if :

- *for all terms $u, v, \overline{w}$, if $(u[x := v]\ \overline{w}) \in \mathcal{S}$, then $(\lambda x.u\ v\overline{w}) \in \mathcal{S}$.*
- *for all terms $u, \overline{v}$, if $\mu\alpha.u[\alpha :=^* \overline{v}] \in \mathcal{S}$, then $(\mu\alpha.u\ \overline{v}) \in \mathcal{S}$.*

We take an example from [21] to show that it is sometimes useful not to have in a model an infinite number of bottoms as well as the relevance of the sets $(R_j)_{j\in J}$. Let $e = \lambda x.\mu\alpha.x$, then we have $\vdash e :\bot\to X$ and for all finite sequence of terms $t\,\bar{u}$, $(e\ t\,\bar{u}) \triangleright^* \mu\alpha.t$. We will prove this general result.

Theorem 2.4 *Let E be a closed term of type $\bot\to X$, then, for each finite sequence of distinct λ-variables $x\,\bar{y}$, $(E\ x\,\bar{y}) \triangleright^* \underline{\mu}.x$ where $\underline{\mu}.x$ denote the variable x preceded by μ-abstractions and μ-applications.*

Proof Let $x\,\bar{y}$ be a finite sequence of distinct λ-variables, $\mathcal{C}_0 = \mathcal{A}$, $\perp\!\!\!\perp_0 = \{t \in \mathcal{T}\ /\ t \triangleright^* \underline{\mu}.x\}$, $\mathcal{R} = \{\bar{y}\} \rightsquigarrow \perp\!\!\!\perp_0$, $\mathcal{M} = \langle \mathcal{C}_0, \perp\!\!\!\perp_0, \mathcal{R}\rangle$ and $\mathcal{I}$ the interpretation such that $\mathcal{I}(X) = \overline{\mathcal{R}}$. By Corollary 2.1, $E \in \mathcal{I}(\bot\to X) = \perp\!\!\!\perp_0 \rightsquigarrow (\{\bar{y}\} \rightsquigarrow \perp\!\!\!\perp_0)$. Since $x \in \perp\!\!\!\perp_0$ and $\bar{y} \in \{\bar{y}\}$, we have $(E\ x\,\bar{y}) \in \perp\!\!\!\perp_0$, and finally $(E\ x\,\bar{y}) \triangleright^* \underline{\mu}.x$. □

The model that we considered in the previous proof only contains $\perp\!\!\!\perp_0$ and the set $\mathcal{R}$ is necessary to find the computational behavior of the term E.

3 The completeness result

Before presenting our completeness result, we need some definitions and some technical results.

3.1 Some results

Lemma 3.1 is very intuitive. It states that, for a variable y, if the term $(t\,y)$ is normalizable, then t is also normalizable. The proof of this result in λ-calculus is very simple because a reduction of $(t\,y)$ will automatically give a reduction of t (the variable y interacts only when t reduces to a λ-abstraction $\lambda x.t'$ and we get in this case $t'[x := y)$). The situation in $\lambda\mu$-calculus is completly different. If the term t is reduced to a μ-abstraction $\mu\alpha.t'$ and we reduce the term $(\mu\alpha.t'\,y)$, we get $\mu\alpha.t'[\alpha :=^* y]$, the variable y can be found in several sub-terms of t' and may also create other redex. Hence the need for a detailed and comprehensive proof of Lemma 3.1. Note also that the proof of this result in [21] is not correct.

Lemma 3.1 *Let t be a term and y a variable. If $(t\,y)$ is normalizable, then t is also normalizable.*

Proof By induction on the pair (l, c) (for the lexicographical order) where $l = l((t\,y))$ and $c = c(t)$.

1. If t does not begin with λ or μ, then $(t\,y) \rhd_l (t'\,y)$ where $t \rhd_l t'$. By induction hypothesis (l decreases), t' is normalizable, then t is also normalizable.

2. If t begins with a μ, then $t = \mu\alpha.\vec{\lambda\mu}.(u\,\bar{u})$.

   - If $u = (\lambda x.v\,w)$, then
     $(t\,y) \rhd_l \mu\alpha.\vec{\lambda\mu}.(\lambda x.v[\alpha :=^* y]\,w[\alpha =^* y]\,\bar{u}[\alpha :=^* y]) \rhd_l$
     $\mu\alpha.\vec{\lambda\mu}.(v[\alpha :=^* y][x := w[\alpha :=^* y]]\,\bar{u}[\alpha :=^* y]) =$
     $\mu\alpha.\vec{\lambda\mu}.(v[x := w][\alpha :=^* y]\,\bar{u}[\alpha :=^* y]) = t''$.
     On the other hand, $t \rhd_l \mu\alpha.\vec{\lambda\mu}.(v[x := w]\ \bar{u}) = t'$ and $(t'\,y) \rhd_l t''$, then $(t'\,y)$ is normalizable, therefore, by induction hypothesis (l decreases), t' is normalizable, thus t is also normalizable.
   - If $u = (\mu\beta.v\,w)$, then
     $(t\,y) \rhd_l \mu\alpha.\vec{\lambda\mu}.(\mu\beta.v[\alpha :=^* y]\,w[x :=^* y]\,\bar{u}[\alpha :=^* y]) \rhd_l$
     $\mu\alpha.\vec{\lambda\mu}.(\mu\beta.v[\alpha :=^* y][\beta :=^* w[\alpha :=^* y]]\,\bar{u}[\alpha :=^* y]) =$
     $\mu\alpha.\vec{\lambda\mu}.(\mu\beta.v[\beta :=^* w[\alpha :=^* y]]\,\bar{u}[\alpha :=^* y)] = t''$.
     On the other hand, $t \rhd_l \mu\alpha.\vec{\lambda\mu}.(\mu\beta.v[\beta :=^* w]\,\bar{u}) = t'$ and $(t'\,y) \rhd_l t''$, then $(t'\,y)$ is normalizable, therefore, by induction hypothesis (l decreases), t' is normalizable, thus t is also normalizable.
   - If $u = x$, then
     $(t\,y) \rhd_l \mu\alpha.\vec{\lambda\mu}.(x\,u_1[\alpha :=^* y] \ldots u_n[\alpha :=^* y])$.
     The term $u_i[\alpha :=^* y]$ is normalizable and $(\mu\alpha.u_i\,y) \rhd_l \mu\alpha.u_i[\alpha :=^* y]$, then $(\mu\alpha.u_i\,y)$ is normalizable and, by induction hypothesis (l does not decrease but c decreases), $\mu\alpha.u_i$ is normalizable, thus u_i is also normalizable. We deduce that t is also normalizable.

- If $u = (\beta\, u')$, then
  $(t\, y) \rhd_l \mu\alpha.\vec{\lambda\mu}.((\beta\, u'[\alpha :=^* y])\ u_1[\alpha :=^* y] \dots u_n[\alpha :=^* y])$.
  As in the previous case, we prove that the terms $u', u_1, \dots, u_n$ are normalizable, then t is also normalizable.
- If $u = (\alpha\, u')$, then
  $(t\, y) \rhd_l \mu\alpha.\vec{\lambda\mu}.((\alpha\, (u'[\alpha :=^* y]\, y))\ u_1[\alpha :=^* y] \dots u_n[\alpha :=^* y])$.
  As in the previous case, we prove that the terms $(u'\, y)$,$u_1, \dots, u_n$ are normalizable, then, by induction hypothesis (l does not decrease but c decreases), $u', u_1, \dots, u_n$ are normalizable, then t is also normalizable.

3. If t begins with λ, we do the same proof

□

In the technical results that we prove in this section, a particular kind of redex appears (the argument of λ-abstraction or μ-abstraction is a variable). It is therefore helpful to understand what is happening by reducing these redexes. Lemma 3.2 will not be used in other proofs but will explain what happens at this special reduction.

Definition 3.1 *Let y be a variable. A y-redex is a β-redex or a μ-redex having y as an argument i.e. a term of the form $(\lambda x.t\, y)$ or $(\mu\alpha.t\, y)$.*

Lemma 3.2 *Let t be a normal term and y a variable such that $y \notin FV(t)$.*
If $(t\, y) \rhd^+ t'$, then each redex in t' is a y-redex preceded by a μ-variable.

Proof By induction the number n of steps to reduce $(t\, y)$ to t'.

1. If $n = 1$, then t is of the form $\lambda x.u$ or $\mu\alpha.u$.
   - If $t = \lambda x.u$, then $t' = u[x := y]$ which is normal.
   - If $t = \mu\alpha.u$, then $t' = \mu\alpha.u[\alpha :=^* y]$. Since u is normal, every new redex of $u[\alpha :=^* y]$ is of the form $(v\, y)$ preceded by α.
2. If $(t\, y) \rhd^{n+1} t'$, then $(t\, y) \rhd^n t'' \rhd t'$. By induction, each redex in t'' is a y-redex preceded by a μ-variable. We examine the reduction $t'' \rhd t'$.
   - If the reduced y-redex of t'' is of the form $(\lambda x.u\, y)$, then the redexes of t' are the same in t'' except the redex that we contracted.
   - If the reduced y-redex of t'' is of the form $(\mu\beta.u\, y)$, then y has evolved from being an argument to this redex, to become an argument for each subterm, of this redex, preceded by β ; so a new y-redex preceded by a μ-variable has been created and the initial redex could not get a new argument as it is locked by β.

□

Lemmas 3.3 and 3.4 will allow us to rebuild the typing in the proof of the completeness theorem. These results are quite technical and to prove them we have to generalize the needed results. For example, in Lemma 3.3, we need two substitutions σ and δ in order to prove Lemma 3.4 and also in Lemma 3.4, we need a simultaneous substitution because the length of such a substitution may increase (case (b)-1).

Lemma 3.3 *Let t be a normal term, y a λ-variables such that $y \notin FV(t)$, $\sigma = [(\beta_j :=^* y)_{1 \le j \le m}]$, $\delta = [(\alpha_i :=^* y)_{1 \le i \le n}]$ and $u = t\sigma$.*
If $\Gamma, y : A \vdash u\delta : C; (\alpha_i : B_i)_{1 \le i \le n}, \Delta$, then $\Gamma, y : A \vdash u : C; (\alpha_i : A \to B_i)_{1 \le i \le n}, \Delta$.

Proof By induction on the normal term t. As t is normal and $t\sigma\delta$ is typable, t is restricted to the following forms : $\lambda x.t'$, $\mu\alpha.t'$, $(x\, t_1 \dots t_k)$ and $(\gamma\, t')$.

1. If $t = \lambda x.t'$, then $u = \lambda x.u'$ where $u' = t'\sigma$, $C = D \to E$ and $\Gamma, y : A, x : D \vdash u'\delta : E; (\alpha_i : B_i)_i, \Delta$. By induction hypothesis, we have $\Gamma, y : A, x : D \vdash u' : E; (\alpha_i : A \to B_i)_i, \Delta$, then $\Gamma, y : A, x : D \vdash u' : E; (\alpha_i : A \to B_i)_i, \Delta$, thus $\Gamma, y : A \vdash \lambda x.u' : D \to E; (\alpha_i : A \to B_i)_i, \Delta$ and finally $\Gamma, y : A \vdash u : C; (\alpha_i : A \to B_i)_i, \Delta$.

2. If $t = \mu\alpha.t'$, then $u = \mu\alpha.u'$ where $u' = t'\sigma$ and $\Gamma, y : A \vdash u'\delta : \bot; \alpha : C, (\alpha_i : B_i)_i, \Delta$. By induction hypothesis, we have $\Gamma, y : A \vdash u' : \bot; \alpha : A \to C, (\alpha_i : A \to B_i)_i, \Delta$, thus $\Gamma, y : A, \vdash \mu\alpha.u' : A \to C; (\alpha_i : A \to B_i)_i, \Delta$, and finally $\Gamma, y : A \vdash u : C; (\alpha_i : A \to B_i)_i, \Delta$.

3. If $t = (x\, t_1 \dots t_k)$, then $u = (x\, u_1 \dots u_k)$ where, for $1 \le r \le k$, $u_r = t_r\sigma$, x has the type $C_1, \dots, C_k \to C$ and, for $1 \le r \le k$, $\Gamma, y : A \vdash u_r\delta : C_r; (\alpha_i : B_i)_i, \Delta$. By induction hypothesis, we have, for $1 \le r \le k$, $\Gamma, y : A \vdash u_r : C_r; (\alpha_i : A \to B_i)_i, \Delta$, then $\Gamma, y : A \vdash (x\, u_1 \dots u_r) : C; (\alpha_i : A \to B_i)_i,\ \Delta$ and finally $\Gamma, y : A \vdash u : C; (\alpha_i : A \to B_i)_i, \Delta$.

4. If $t = (\gamma\, t')$ where $\gamma \neq \alpha_i$ and $\gamma \neq \beta_j$, then $u = (\gamma\, u')$ where $u' = t'\sigma$, $C = \bot$, γ has the type D and $\Gamma, y : A \vdash u'\delta : D; (\alpha_i : B_i)_i, \Delta$. By induction hypothesis, we have $\Gamma, y : A \vdash u' : D; (\alpha_i : A \to B_i)_i, \Delta$, then $\Gamma, y : A \vdash (\gamma\, u') : \bot; (\alpha_i : A \to B_i)_i, \Delta$ and finally $\Gamma, y : A \vdash u : C; (\alpha_i : A \to B_i)_i, \Delta$.

5. If $t = (\gamma\, t')$ where $\gamma = \beta_k$ $(1 \le k \le m)$, then $u = (\gamma\, (u'\, y))$ where $u' = t'\sigma$, $C = \bot$, γ has the type D and $\Gamma, y : A \vdash (u'\delta\, y) : D; (\alpha_i : B_i)_i, \Delta$, thus $\Gamma, y : A \vdash u'\delta : A \to D; (\alpha_i : B_i)_i, \Delta$. By induction hypothesis, we have $\Gamma, y : A \vdash u' : A \to D; (\alpha_i : A \to B_i)_i, \Delta$, then $\Gamma, y : A \vdash (\gamma\, (u'\, y)) : \bot; (\alpha_i : A \to B_i)_i, \Delta$ and finally $\Gamma, y : A \vdash u : C; (\alpha_i : A \to B_i)_i, \Delta$.

6. If $t = (\gamma\, t')$ where $\gamma = \alpha_k$ $(1 \le k \le n)$, then $u = (\gamma\, u')$ where $u' = t'\sigma$, $u\delta = (\gamma\, (u'\delta\, y)$, $C = \bot$ and $\Gamma, y : A \vdash (u'\delta\, y) : B_k; (\alpha_i : B_i)_i, \Delta$, then $\Gamma, y : A \vdash u'\delta : A \to B_k; (\alpha_i : B_i)_i, \Delta$. By induction hypothesis, we have $\Gamma, y : A \vdash u' : A \to B_k; (\alpha_i : A \to B_i)_i, \Delta$, then $\Gamma, y : A \vdash (\gamma\, u') : \bot; (\alpha_i : A \to B_i)_i, \Delta$ and finally $\Gamma, y : A \vdash u : C; (\alpha_i : A \to B_i)_i, \Delta$.

□

Lemma 3.4 *Let t be a normal term, y a λ-variable such that $y \notin FV(t)$ and $\sigma = [(\alpha_i :=^* y)_{1 \le i \le n}]$.*

1. *If $(t\sigma\, y)$ is normalizable, t' its normal form and $\Gamma, y : A \vdash t' : C; \Delta$, then $\Gamma, y : A \vdash t\sigma : A \to C; \Delta$.*

2. *If $t\sigma$ is normalizable, t'' its normal form and $\Gamma, y : A \vdash t'' : C; \Delta$, then $\Gamma, y : A \vdash t\sigma : C; \Delta$.*

Proof By simultaneous induction on $c(t)$.

1. As t is normal and t' is typable, t is restricted to the following forms $\lambda x.u$, $\mu\beta.u$ and $(x\, u_1 \dots u_m)$.

   (a) If $t = \lambda x.u$, then $(t\sigma\, y) \triangleright u[x := y]\sigma \triangleright^* t'$. By induction hypothesis (2), we have $\Gamma, y : A \vdash u[x := y]\sigma : C; \Delta$, then $\Gamma, y : A, y' : A \vdash u[x := y']\sigma : C; \Delta$ where y' is a new λ-variable, thus $\Gamma, y : A \vdash \lambda y'.u[x := y']\sigma : A \to C; \Delta$ and finally $\Gamma, y : A \vdash t\sigma : A \to C; \Delta$.

(b) If $t = \mu\beta.u$, then $(t\sigma\, y) \triangleright \mu\beta.u[\sigma, \beta :=^* y] \triangleright^* t'$, therefore, $t' = \mu\beta.u'$, u' is the normal form of $u[\sigma, \beta :=^* y]$ and $\Gamma, y : A \vdash u' :\bot; \beta : C, \Delta$. By induction hypothesis (2), we have $\Gamma, y : A \vdash u\sigma[\beta :=^* y] :\bot; \beta : C, \Delta$, then, by Lemma 3.3, $\Gamma, y : A \vdash u\sigma :\bot; \beta : A \to C, \Delta$, thus $\Gamma, y : A \vdash \mu\beta.u\sigma : A \to C; \Delta$, and finally $\Gamma, y : A \vdash t\sigma : A \to C; \Delta$.

(c) If $t = (x\, u_1 \dots u_m)$, then $t' = (x\, u'_1 \dots u'_m y)$ where, for $1 \leq j \leq m$, u'_j is the normal form of $u_j\sigma$, x has the type $C_1, \dots, C_n, A \to C$ and, for $1 \leq j \leq m$, $\Gamma, y : A \vdash u'_j : C_j; \Delta$. By induction hypothesis (2), we have $\Gamma, y : A \vdash u_j\sigma : C_j; \Delta$, then $\Gamma, y : A \vdash (x\, u_1\sigma \dots u_n\sigma) : A \to C; \Delta$ and finally $\Gamma, y : A \vdash t\sigma : A \to C; \Delta$.

2. As t is normal and t'' is typable, t is restricted to the following forms $\lambda x.u$, $\mu\beta.u$, $(x\, u_1 \dots u_m)$ and $(\beta\, u)$.

(a) If $t = \lambda x.u$, then $t'' = \lambda x.u''$ where u'' is the normal form of $u\sigma$, $C = D \to E$ and $\Gamma, y : A, x : D \vdash u'' : E; \Delta$. By induction hypothesis (2), we have $\Gamma, y : A, x : D \vdash u\sigma : E; \Delta$, then $\Gamma, y : A \vdash \lambda x.u\sigma : D \to E; \Delta$ and finally $\Gamma, y : A \vdash t\sigma : C; \Delta$.

(b) If $t = \mu\beta.u$, then $t'' = \mu\beta.u''$ where u'' is the normal form of $u\sigma$ and $\Gamma, y : A \vdash u'' :\bot; \beta : C, \Delta$. By induction hypothesis (2), we have $\Gamma, y : A \vdash u\sigma :\bot; \beta : C, \Delta$, then $\Gamma, y : A \vdash \mu\beta.u\sigma : C; \Delta$ and finally $\Gamma, y : A \vdash t\sigma : C; \Delta$.

(c) If $t = (x\, u_1 \dots u_m)$, then $t'' = (x\, u''_1 \dots u''_m)$ where, for $1 \leq j \leq m$, u''_j is the normal form of $u_j\sigma$, x has the type $C_1, \dots, C_m \to C$ and, for $1 \leq j \leq m$, $\Gamma, y : A \vdash u''_j : C_j; \Delta$. By induction hypothesis (2), we have, for $1 \leq j \leq m$, $\Gamma, y : A \vdash u_j\sigma : C_j; \Delta$, then $\Gamma, y : A \vdash (x\, u_1\sigma \dots u_m\sigma) : C; \Delta$ and finally $\Gamma, y : A \vdash t\sigma : C; \Delta$.

(d) If $t = (\beta\, u)$ where $\beta \neq \alpha_j$, then $t'' = (\beta\, u'')$ where u'' is the normal form of $u\sigma$, $C =\bot$, β has the type D and $\Gamma, y : A \vdash u'' : D; \Delta$. By induction hypothesis (2), we have $\Gamma, y : A \vdash u\sigma : D; \Delta$, then $\Gamma, y : A \vdash (\beta\, u\sigma) :\bot; \Delta$ and finally $\Gamma, y : A \vdash t\sigma : C; \Delta$.

(e) If $t = (\alpha_j\, u)$ $(1 \leq j \leq n)$, then $t'' = (\alpha_j\, u'')$ where u'' is the normal form of $(u_j\sigma\, y)$, $C =\bot$ and $\Gamma, y : A \vdash u'' : B; \Delta$. By induction hypothesis (1), we have $\Gamma, y : A \vdash u\sigma : A \to B; \Delta$, then $\Gamma, y : A \vdash (\alpha_j\, (u\sigma\, y)) :\bot; \Delta$ and finally $\Gamma, y : A \vdash t\sigma : C; \Delta$.

□

3.2 Completeness model

We will now prove that if t is in the general interpretation of a type A, then t has the type A. For this, we will construct a particular term model $\mathbb{M}$ in which, we will get the equivalence between "having the type A" and "being within the type A" (see Lemma 3.6). The construction of this model looks like the constructions of the completeness models of the papers [8, 9, 10, 11, 15, 4, 5, 12, 21]. We start with enumerating infinite sets of variables (which will be parameters of this model), then enumerating sets of types and finally fixing two infinite contexts (by associating enumerated types to enumerated variables) in which the terms will be typed. This will allow to define the bottoms and the variable sets associated to the models. Note that in this completeness model we don't need the sets $(R_j)_{j \in J}$.

Definition 3.2 *1. Let $\mathbb{V}_1 = \{x_i \ / \ i \in \mathbb{N}\}$ (resp., $\mathbb{V}_2 = \{\alpha_i \ / \ i \in \mathbb{N}\}$ be an enumeration of an infinite set of λ-variables (resp., μ-variables). We put $\mathbb{V} = \mathbb{V}_1 \bigcup \mathbb{V}_2$.*

2. *Let* $\mathbb{T}_1 = \{A_i \ / \ i \in \mathbb{N}\}$ *and* $\mathbb{T}_2 = \{B_i \ / \ i \in \mathbb{N}\}$ *be enumerations of all types where each type comes infinitely many times.*

3. *We define* $\mathbb{G} = \{x_i : A_i \ / \ i \in \mathbb{N}\}$ *and* $\mathbb{D} = \{\alpha_i : B_i \ / \ i \in \mathbb{N}\}$.

4. *Let* u *be a term, such that* $FV(u) \subseteq \mathbb{V}$, *the contexts* $\mathbb{G}_u$ *(resp.* $\mathbb{D}_u$*) are defined as the restrictions of* $\mathbb{G}$ *(resp.* $\mathbb{D}$*) at the declarations containing the variables of* $FV(u)$.

5. *The notation* $\mathbb{G} \vdash u : C;\ \mathbb{D}$ *means that* $\mathbb{G}_u \vdash u : C;\ \mathbb{D}_u$, *we denote* $\mathbb{G} \vdash^* u : C;\ \mathbb{D}$ *iff there exists a term* u', *such that* $u \rhd^* u'$ *and* $\mathbb{G} \vdash u' : C;\ \mathbb{D}$.

6. *Let* $\mathbb{P} = \{X_i \ / \ i \in \mathbb{N}\}$ *be an enumeration of* $\{\bot\} \cup \mathcal{P}$. *We assume that* $X_0 = \bot$.

7. *For each* $i \in \mathbb{N}$, *let* $\perp\!\!\!\perp_i = \{t \ / \ \mathbb{G} \vdash^* t : X_i;\ \mathbb{D}\}$ *and* $\mathbb{C}_i = \{\alpha \in \mathbb{V}_2 \ / \ (\alpha : X_i) \in \mathbb{D}\}$.

Lemma 3.5 $\mathbb{M} = \langle (\mathbb{C}_i)_{i\in\mathbb{N}}, (\perp\!\!\!\perp_i)_{i\in\mathbb{N}}, \emptyset\rangle$ *is a model for* S_μ.

Proof It is easy to show that $(\perp\!\!\!\perp_i)_{i\in\mathbb{N}}$ is a sequence of saturated subsets of terms.

- $\forall\, i \in \mathbb{N}$, if $\alpha \in \mathbb{C}_i$ and $u \in \perp\!\!\!\perp_0$, then $(\alpha : X_i) \in \mathbb{D}$ and $\mathbb{G} \vdash^* u : \bot;\ \mathbb{D}$, therefore $\mathbb{G} \vdash^* \mu\alpha.u : X_i;\ \mathbb{D}$, thus $\mu\alpha.u \in \perp\!\!\!\perp_i$.
- $\forall\, i \in \mathbb{N}$, if $\alpha \in \mathbb{C}_i$ and $u \in \perp\!\!\!\perp_i$, then $(\alpha : X_i) \in \mathbb{D}$ and $\mathbb{G} \vdash^* u : X_i;\ \mathbb{D}$, therefore $\mathbb{G} \vdash^* (\alpha\, u) : \bot;\ \mathbb{D}$, thus $(\alpha\, u) \in \perp\!\!\!\perp_0$.

□

Observe that the model $\mathbb{M}$ is parameterized by the two infinite sets of variables and the enumerations. We need just these infinite sets of variables and not all the variables. This is an important remark since it will serve us in the proof of compleness theorem (Theorem 3.1).

Definition 3.3 *We define the* $\mathbb{M}$*-interpretation* $\mathbb{I}$ *as follows* $\forall\, i \in \mathbb{N}$, $\mathbb{I}(X_i) = \perp\!\!\!\perp_i$.

The following lemma is the generalized version of the completeness theorem. It states the equivalence between "a term t is of type A in the fixed contexts" and "a term t belongs to the interpretation of the type A in the model $\mathbb{M}$". The proof is done by simultaneous induction and uses the technical lemmas from the beginning of this section.

Lemma 3.6 *Let* A *be a type and* t *a term.*

1. *If* $\mathbb{G} \vdash^* t : A\ ;\mathbb{D}$, *then* $t \in \mathbb{I}(A)$.
2. *If* $t \in \mathbb{I}(A)$, *then* $\mathbb{G} \vdash^* t : A\ ;\mathbb{D}$.

Proof By a simultaneous induction on $c(A)$.

1. (a) If $A = X$ or $\bot$, the result is immediate from the definition of $\mathbb{I}$.
   (b) If $A = B \to C$, then $t \rhd^* t'$ such that: $\mathbb{G} \vdash t' : B \to C\ ;\mathbb{D}$. Let $u \in \mathbb{I}(B)$. By induction hypothesis (2), we have $\mathbb{G} \vdash^* u : B\ ;\mathbb{D}$, this implies that $u \rhd^* u'$ and $\mathbb{G} \vdash u' : B\ ;\mathbb{D}$. Hence $\mathbb{G} \vdash (t'\ u') : C\ ;\mathbb{D}$, so, by the fact that $(t\ u) \rhd^* (t'\ u')$, we have $\mathbb{G} \vdash^* (t\ u) : C\ ;\mathbb{D}$, then, by induction hypothesis (1), $(t\ u) \in \mathbb{I}(C)$. Therefore $t \in \mathbb{I}(B \to C)$.
2. (a) If $A = X$ or $\bot$, the result is immediate from the definition of $\mathbb{I}$.

(b) If $A = B \to C$ and $t \in \mathbb{I}(B) \rightsquigarrow \mathbb{I}(C)$, let y be a λ-variable such $y \notin FV(t)$ and $(y : B) \in \mathbb{G}$. We have $y : B \vdash y : B$, hence, by induction hypothesis (1), $y \in \mathbb{I}(B)$, then, $(t\ y) \in \mathbb{I}(C)$. By induction hypothesis (2), $\mathbb{G} \vdash^* (t\ y) : C\ ; \mathbb{D}$, then $(t\ y) \triangleright^* u$ such that $\mathbb{G} \vdash u : C\ ; \mathbb{D}$ and, by the Lemma 3.1, t is a normalizable term. Let t' (resp. u') be the normal form of t (resp. of u). So $(t'\ y) \triangleright^* u'$, by Lemma 3.4, $\mathbb{G} \vdash t' : B \to C\ ; \mathbb{D}$, then $\mathbb{G} \vdash^* t : B \to C\ ; \mathbb{D}$,

□

Note that the item 1 of Lemma 3.6 can not be deduced from the correctness lemma. This comes from the presence of contexts $\mathbb{G}$ and $\mathbb{D}$ to type a reduced of term t.

We can now state and prove the completeness theorem.

Theorem 3.1 (Completeness theorem) *Let A be a type and t a term.*
We have $t \in |A|$ iff there exists a closed term t' such that $t \triangleright^* t'$ and $\vdash t' : A$.

Proof

$\Leftarrow$: By Corollary 2.1, $t' \in |A|$, then, $t \in |A|$ because $|A|$ is saturated.

$\Rightarrow$: We consider an infinite set of λ and μ variables Ω which contains none of the free variables of t, then from this set we build the completeness model as described in Definition 3.2. If $t \in |A|$, then $t \in \mathbb{I}(A)$, hence by (2) of Lemma 3.6 and by the fact that $FV(t') \subseteq FV(t)$, we have $t \triangleright^* t'$ and $\vdash t' : A$.

□

Here are some direct and unexpected consequences of the completeness theorem.

Corollary 3.1 *Let A be a type and t a term.*

1. *If $t \in |A|$, then t is normalizable.*
2. *If $t \in |A|$, then there exists a closed term t' such that $t \simeq t'$.*
3. *The set $|A|$ is closed under equivalence.*

Proof (1) and (2) are direct consequences of Theorems 2.3 and 3.1. (3) can be deduced from Theorem 3.1 and Lemma 2.6. □

4 Future work

Throughout this work we have seen that the propositional types of the system S_μ are complete for a realizability semantics. Two questions will be interesting to study.

1. The models that we have considered in this paper are sufficient to get correctness and completeness results: the saturation conditions that we have imposed in these models allow to have these two results. It will be interesting to understand more this kind of models: for example, build models with more bottoms to study the computational behaviour of some typed terms.
2. What about the types of the second order typed $\lambda\mu$-calculus? We know that, for the system $\mathcal{F}$, the $\forall^+$-types (types with positive quantifiers) are complete for a realizability semantics (see [4, 15]). But for the classical system $\mathcal{F}$, we cannot easily generalize this result. We think we need to add more restrictions on the positions of $\forall$ in the $\forall^+$-types to obtain the smallest class of types that we suppose can be proved to be complete.

Acknowledgements. We wish to thank P. Battyányi and N. Bernard for the numerous corrections and suggestions.

References

[1] T. Coquand. *Completeness theorem and λ-calculus.* The 7th International Conference, TLCA 2005, Nara, Japan, April 21-23, 2005, Lecture Notes in Computer Science, pp. 1-9, volume 3461/2005.

[2] R. David and K. Nour. *A short proof of the strong normalization of the simply typed $\lambda\mu$-calculus.* Schedae Informaticae vol 12, pp. 27-33, 2003.

[3] Ph. de Groote. *An environment machine for the lambda-mu-calculus.* Mathematical Structure in Computer Science, vol 8, pp. 637-669, 1998.

[4] S. Farkh and K. Nour. *Un résultat de complétude pour les types $\forall^+$ du système $\mathcal{F}$.* CRAS. Paris 326, Série I, pp. 275-279, 1998.

[5] S. Farkh and K. Nour. *Résultats de complètude pour des classes de types du système $\mathcal{AF}2$.* Informatique Théorique et Application, vol 31, num 6, pp. 513-537, 1998.

[6] J.-Y. Girard, Y. Lafont and P. Taylor. Proofs and types. *Cambridge University Press*, 1986.

[7] T. Griffin. *A formulae-as-types notion of control.* Proc. POPL, 1990.

[8] J.R. Hindley, *The simple semantics for Coppo-Dezani-Sallé types.* In International symposium on programming (Turin, 1982). Lecture Notes and Computer Science, vol 137, Springer, Berlin, pp. 212-226, 1982.

[9] J. R. Hindley. *The completeness theorem for typing λ-terms.* Theoretical Computer Science, vol 22, pp. 1-17, 1983.

[10] J. R. Hindley. *Curry's type-rules are complete with respect to the F-semantics too.* Theoretical Computer Science, vol 22, pp. 127-133, 1983.

[11] J.R. Hindley. *Basic Simple Type Theory.* Cambridge University Press, 1997.

[12] F. Kamareddine and K. Nour. *A completeness result for a realizability semantics for an intersection type system.* Annals of Pure and Applied Logic, vol 146, pp. 180-198, 2007

[13] J.-L. Krivine. *Lambda calcul, types et modèles.* Masson, Paris, 1990.

[14] J.-L. Krivine. *Opérateurs de mise en mémoire et traduction de Gödel.* Archive for Mathematical Logic, vol 30, pp. 241-267, 1990.

[15] R. Labib-Sami. *Typer avec (ou sans) types auxiliaires.* Manuscrit, 1986.

[16] C. R. Murthy. *An evaluation semantics for classical proofs.* Proceedings of the sixth annual IEEE symposium, pp. 96-107, 1991.

[17] K. Nour. *La valeur d'un entier classique en lambda-mu-calcul.* Archive for Mathematical Logic 36, pp. 461-473, 1997.

[18] K. Nour. *Mixed Logic and Storage Operators.* Archive for Mathematical Logic, vol 39, pp. 261-280, 2000.

[19] K. Nour and K. Saber. *A semantical proof of strong normalization theorem for full propositional classical natural deduction.* Archive for Mathematical Logic, vol 45, pp. 357-364, 2005.

[20] K. Nour and K. Saber. *A Semantics of Realizability for the Classical Propositional Natural Deduction.* Electronic Notes in Theoretical Computer Science, vol 140, pp. 31-39, 2005.

[21] K. Nour and K. Saber. *A completeness result for the simply typed lambda-mu calculus.* Annals of Pure and Applied Logic, vol 161, pp. 109-118, 2009.

[22] M. Parigot. *Free Deduction: An Analysis of "Computations" in Classical Logic.* Lecture Notes in Computer Science (592), pp. 361-380, Springer-Verlag, Berlin, 1990.

[23] M. Parigot. *$\lambda\mu$-calculus: An algorithm interpretation of classical natural deduction.* Lecture Notes in Artificial Intelligence, vol 624, pp. 190-201. Springer Verlag, 1992.

[24] M. Parigot. *Classical proofs as programs.* Lecture Notes in Computer Science (713), pp. 263-276, Springer Verlag, Berlin, 1993.

[25] M. Parigot. *Proofs of strong normalization for second order classical natural deduction.* Journal of Symbolic Logic, vol 62 (4), pp. 1461-1479, 1997.

[26] W. Py. *Confluence en $\lambda\mu$-calcul.* PhD thesis, University of Chambéry, 1998.

[27] K. Saber. *Étude d'un λ-calcul issu d'une logique classique.* PhD Thesis, University of Chambéry, 2007.

[28] A. Saurin. *Separation and the $\lambda\mu$-calculus.* Proceedings of the Twentieth Annual IEEE Symp. on Logic in Computer Science, LICS 2005, IEEE Computer Society Press, pp. 356-365, 2005.

[29] S. van Bakel, F. Barbanera and U. de'Liguoro. *A Filter Model for the $\lambda\mu$-Calculus.* Proceedings of 10th International Conference on Typed Lambda Calculi and Applications (TLCA'11), Novi Sad, Serbia, June 1-3, 2011. Volume 6690 of Lecture Notes in Computer Science, pages 213-228, Springer-Verlag, 2011.

[30] W. W. Tait. *A realizability interpretation of the theory of species.* In : R. Parikh (Ed.), Logic Colloquium Boston 1971/72, vol. 435 of Lecture Notes in Mathematics, Springer Verlag, pp. 240-251, 1975.