A completeness result for the simply typed $\lambda\mu$-calculus

Karim NOUR & **Khelifa SABER**
LAMA - Équipe LIMD
Université de Chambéry
73376 Le Bourget du Lac
e-mail : {knour,khsab}@univ-savoie.fr

Abstract

In this paper, we define a semantic of realizability for the simply typed $\lambda\mu$-calculus. An adequation lemma is proved, it allows to give characterizations of the computational behavior of some closed typed terms through their types. We dealt also with the problem of completeness and prove a completeness result.

1 Introduction

What came to be called the Curry-Howard correspondence has proven to be a robust technique to study proofs of intuitionistic logic, since it exhibits the structural bond between this logic and the λ-calculus. T. Griffin's works [7] in 1990 allowed to extend this correspondence to the classical logic, which had several consequences. On basis of this new contribution, the $\lambda\mu$-calculus was introduced by M. Parigot [19] and [20]. The $\lambda\mu$-calculus is a natural extension of the λ-calculus which exactly captures the algorithmic content of proofs written in the second order classical natural deduction system. The typed $\lambda\mu$-calculus enjoys all the good properties: the subject reduction, the strong normalization and confluence theorems.

The strong normalization theorem of the second order classical natural deduction [20] is based on a result known as the correctness lemma, which stipulates that each term is in the interpretation of its type. This is also based on the notion of the semantics of realizability. The idea of this semantics consists in associating to each type a set of terms which *realizes* it, this method has been very effective for establishing the strong normalization of type system "à la Tait and Girard". J.- Y. Girard used it to give a proof of the strong normalization of its system $\mathcal{F}$, method known also as the reducibility candidates, later M. Parigot extended this method to the classical case and provided a proof of the strong normalization of the typed $\lambda\mu$-calculus. In a previous work [16], we 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 proof of strong normalization use a variant of the reducibility candidates based on a correctness lemma, which has been important also for characterizing the operational behavior of some typed terms and this only through their types, as it was done in J.-L. Krivine's works [12]. This inspired us also to define a general semantics for the classical natural deduction in [15] and gave such characterizations.

The question that we can ask now is: "does the correctness lemma have a converse?". By this we mean: "can we find a class of types for which the converse of the correctness lemma (completeness result) holds?". J.R. Hindley was the first to study the completeness of simple type system property [8], [9] and [10]. R. Labib-sami has established in [14] completeness for a class of types in Girard's system $\mathcal{F}$ known as strictely positive types, and this for a semantics based on the sets stable under the $\beta\eta$-equivalence. S. Farkh and K. Nour revisited this result, and generalized it, in fact they proved a refined result by indicating that weak-head-expansion is sufficient [4]. In [5], they established an other completeness result for a class of types in Krivine's system $\mathcal{AF}2$. Recently, F. Kamareddine and K. Nour improved the result of Hindley, to a system with an intersection type. Independently, T. Coquand established in [1] by methods using Kripke's models, the completeness for the simply typed λ-calculus.

In the present work we dealt with this problem and we prove the completeness for the simply typed $\lambda\mu$-calculus. The semantics that we define here is not completely different from that of [16] and [15], nevertheless we add a slight but an indispensable modification to the notion of the μ-saturation. In fact, to show that each element $\mathcal{R}$ of the model can be written in the form $\mathcal{R}^{\perp} \to \mathcal{S}$ (where $\mathcal{S}$ replaced the set $\mathcal{N}$ of strongly normalizable term and satisfied the saturation property), and under the constraint of the definition of these $\mathcal{R}$ imposed by the completeness side, we are compelled to bring this subtle modification.

Moreover the strong normalization, the semantics of the adequation lemma allows to give short proofs of theorems describing the computational behavior of closed typed terms through their types. Nevertheless, the proofs suppose well known the behaviors, therefore the models are exactly built to satisfy the required properties. This is not the case of the syntactical proofs, where we guess the behavior through the type, which is rather constructive, but of an other share these proofs are more complicated than the semantical ones. In what follows, we give at each time, both of semantics and syntactical proofs.

This paper is organized as follows. Section 2 is an introduction to the simply typed $\lambda\mu$-calculus. In section 3, we define the semantics and prove a correctness lemma. In Section 4, we give characterizations of some closed typed terms. Finally, Section 5 is devoted to the completeness result.

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

Definition 2.1 1. *Let $\mathcal{X}$ and $\mathcal{A}$ be two infinite sets of disjoint alphabets for distinguiching λ-variables and μ-variables. The $\lambda\mu$-terms are given by the following grammar:*

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

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

3. *As usual we denote by $\neg A$ the formula $A \to \perp$. Let $A_1, A_2, ..., A_n, A$ be types, we denote the type $A_1 \to (A_2 \to (... \to (A_n \to A)...))$ by $A_1, A_2, ..., A_n \to A$.*

4. *Proofs are presented in natural deduction system with two conclusions, such that formulas in the left of $\vdash$ are indexed by λ-variables and those in right of $\vdash$ are indexed by μ-variables, except one which is indexed by a term.*

5. *Let t be a $\lambda\mu$-term, A a type, $\Gamma = \{x_i : A_i\}_{1\leq i\leq n}$ and $\Delta = \{a_j : B_j\}_{1\leq j\leq m}$, using the following rules, we will define "t typed with type A in the contexts Γ and Δ" and we denote it $\Gamma \vdash t : A\ ; \Delta$.*

$$\frac{}{\Gamma \vdash x_i : A_i \; ; \; \Delta} ax \quad \textit{for } 1 \leq i \leq n.$$

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

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

We denote this typed system by S_μ*.*

6. *The basic reduction rules are* β *and* μ *reductions.*
   - $(\lambda x.u\ v) \triangleright_\beta u[x := v]$
   - $(\mu a.u\ v) \triangleright_\mu \mu a.u[a :=^* v]$
     where $u[a :=^* v]$ *is obtained from* u *by replacing inductively each subterm in the form* $(a\ w)$ *in* u *by* $(a\ (w\ v))$*.*

7. *We denote* $t \triangleright t'$ *if* t *is reduced to* t' *by one of the rules given above. As usual* $\triangleright^*$ *denotes the reflexive transitive closure of* $\triangleright$*, and* $\simeq$ *the equivalence relation induced by* $\triangleright^*$*.*

We have the following results (for more lecture, see [20]).

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.*

Definition 2.2

1. *Let* t *be a term and* $\bar{v}$ *a finite sequence of terms (the empty sequence is denoted by* $\emptyset$*), then, the term* $t\bar{v}$ *is defined by* $(t\ \emptyset) = t$ *and* $(t\ u\bar{u}) = ((t\ u)\ \bar{u})$*.*

2. *Let* $t, u_1, ..., u_n$ *be terms and* $\bar{v}_1, ..., \bar{v}_m$ *finite sequences of terms, then*

   $t[(x_i := u_i)_{1 \leq i \leq n}; (a_j :=^* \bar{v}_j)_{1 \leq j \leq m}]$ *is obtained from the term* t *by replacing inductively each* x_i *by* u_i *and each subterm in the form* $(a_j\ u)$ *in* t *by* $(a_j\ (u\, \bar{v}_j))$*.*

Lemma 2.1 *Let* $\sigma = [(x_i := u_i)_{1 \leq i \leq n}; (a_j :=^* \bar{v}_j)_{1 \leq j \leq m}]$ *and* t, t' *two terms such that* $t \triangleright^* t'$*, then,* $t\sigma \triangleright^* t'\sigma$*.*

Proof. By induction on t. ■

3 The semantics of S_μ

Definition 3.1

1. *Let* $\mathcal{S}$ *be a set of terms, we say that* $\mathcal{S}$ *is a saturated set iff for each terms* u *and* v*, if* $u \in \mathcal{S}$ *and* $v \triangleright^* u$*, then,* $v \in \mathcal{S}$*.*

2. *Let us take a saturated set of terms* $\mathcal{S}$ *and a set* $\mathcal{C}$ *of an infinite classical variables (*μ*-variables). We said that* $\mathcal{S}$ *is* $\mathcal{C}$*-saturated iff for each* $t \in \mathcal{S}$ *and for each* $a \in \mathcal{C}$*,* $\mu a.t \in \mathcal{S}$ *and* $(a\ t) \in \mathcal{S}$*.*

3. *Consider two sets of terms $\mathcal{K}$ and $\mathcal{L}$, we define a new set of terms:* $\mathcal{K} \rightsquigarrow \mathcal{L} = \{t \ / \ (t \ u) \in \mathcal{L}$, *for each* $u \in \mathcal{K}\}$. *It is clear that when* $\mathcal{L}$ *is a saturated set, then* $\mathcal{K} \rightsquigarrow \mathcal{L}$ *is also a saturated one.*

4. *We denote* $\mathcal{T} \cup \mathcal{A}$ *by* $\mathcal{T}'$ *and* ${\mathcal{T}'}^{<\omega}$ *the set of finite sequences of* $\mathcal{T}'$. *Let* t *be a term and* $\pi \in {\mathcal{T}'}^{<\omega}$, *then the term* $(t \ \pi)$ *is defined by* $(t \ \emptyset) = t$, $(t \ \pi) = ((t \ u) \ \pi')$ *if* $\pi = u\pi'$ *and* $(t \ \pi) = ((a \ t) \ \pi')$ *if* $\pi = a\pi'$.

5. *Let* $\mathcal{S}$ *be a set of terms and* $\mathcal{X} \subseteq {\mathcal{T}'}^{<\omega}$, *then we define* $\mathcal{X} \rightsquigarrow \mathcal{S} = \{t \ / \ (t \ \pi) \in \mathcal{S}$, *for each* $\pi \in \mathcal{X}\}$.

Remark 3.1 *The fact that the application* $(a\,t)$ *is denoted by* $(t\,a)$ *is not something new, it is already present in Saurin's work [23]. Except that for us, it is a simple notation in order to uniformize the definition of the application. But for Saurin, it is crucial to obtain the theorem of separation in the* $\lambda\mu$*-calculus.*

Definition 3.2 *Let* $\mathcal{S}$ *be a* $\mathcal{C}$*-saturated set and* $\{\mathcal{R}_i\}_{i\in I}$ *subsets of terms such that* $\mathcal{R}_i = \mathcal{X}_{\mathcal{R}_i} \rightsquigarrow \mathcal{S}$ *for some* $\mathcal{X}_{\mathcal{R}_i} \subseteq {\mathcal{T}'}^{<\omega}$. *A model* $\mathcal{M}=\langle \mathcal{C}, \mathcal{S}, \{\mathcal{R}_i\}_{i\in I}\rangle$ *is the smallest set of subsets of terms containing* $\mathcal{S}$ *and* $\mathcal{R}_i$, *and closed under the constructor* $\rightsquigarrow$.

Lemma 3.1 *Let* $\mathcal{M} = \langle \mathcal{C}, \mathcal{S}, \{\mathcal{R}_i\}_{i\in I}\rangle$ *be a model and* $\mathcal{G} \in \mathcal{M}$. *There exists a set* $\mathcal{X}_\mathcal{G} \subseteq {\mathcal{T}'}^{<\omega}$ *such that* $\mathcal{G} = \mathcal{X}_\mathcal{G} \rightsquigarrow \mathcal{S}$.

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

- If $\mathcal{G} = \mathcal{S}$, take $\mathcal{X}_\mathcal{G} = \{\phi\}$.
- If $\mathcal{G} = \mathcal{R}_i$, take $\mathcal{X}_\mathcal{G} = \mathcal{X}_{\mathcal{R}_i}$.
- If $\mathcal{G} = \mathcal{G}_1 \rightsquigarrow \mathcal{G}_2$, then, by the induction hypothesis, $\mathcal{G}_2 = \mathcal{X}_{\mathcal{G}_2} \rightsquigarrow \mathcal{S}$ where $\mathcal{X}_{\mathcal{G}_2} \subseteq {\mathcal{T}'}^{<\omega}$, and take $\mathcal{X}_\mathcal{G} = \{u\bar{v} \ / \ u \in \mathcal{G}_1$ and $\bar{v} \in \mathcal{X}_{\mathcal{G}_2}\}$.

■

Definition 3.3 *Let* $\mathcal{M} = \langle \mathcal{C}, \mathcal{S}, \{\mathcal{R}_i\}_{i\in I}\rangle$ *be a model and* $\mathcal{G} \in \mathcal{M}$. *We define the set* $\mathcal{G}^\perp = \cup\{\mathcal{X}_\mathcal{G} \ / \ \mathcal{G} = \mathcal{X}_\mathcal{G} \rightsquigarrow \mathcal{S}\}$.

Lemma 3.2 *Let* $\mathcal{M} = \langle \mathcal{C}, \mathcal{S}, \{\mathcal{R}_i\}_{i\in I}\rangle$ *be a model and* $\mathcal{G} \in \mathcal{M}$. *We have* $\mathcal{G} = \mathcal{G}^\perp \rightsquigarrow \mathcal{S}$.

Proof. This comes from the fact that: if for every $j \in J$, $\mathcal{G} = \mathcal{X}_{\mathcal{G}_j} \rightsquigarrow \mathcal{S}$, then, $\mathcal{G} = (\cup_{j\in J}\mathcal{X}_{\mathcal{G}_j}) \rightsquigarrow \mathcal{S}$. ■

Definition 3.4

1. *Let* $\mathcal{M} = \langle \mathcal{C}, \mathcal{S}, \{\mathcal{R}_i\}_{i\in I}\rangle$ *be a model. An* $\mathcal{M}$*-interpretation* $\mathcal{I}$ *is an application* $X \mapsto \mathcal{I}(X)$ *from the set of propositional variables* $\mathcal{P}$ *in* $\mathcal{M}$ *which we extend for any formula as follows:*
   - $\mathcal{I}(\perp) = \mathcal{S}$
   - $\mathcal{I}(A \to B) = \mathcal{I}(A) \rightsquigarrow \mathcal{I}(B)$.
2. *For any type* A, *we denote* $|A|_\mathcal{M} = \bigcap\{\mathcal{I}(A) \ / \ \mathcal{I}$ *an* $\mathcal{M}$*-interpretation*$\}$.
3. *For any type* A, $|A| = \bigcap\{|A|_\mathcal{M} \ / \ \mathcal{M}$ *a model*$\}$.
4. *Let* u, v *be two terms. The expression* $v \approx_\mathcal{C} u$ *means that* v *is obtained from* u *by replacing the free classical variables of* u *by some others in* $\mathcal{C}$, *i.e. if we denote* u *by* $u[a_1, ..., a_n]$ *where the* a_i *are the free classical variables of* u, *then* v *will be* $u[a_1 := b_1, ..., a_n := b_n]$ *where* $b_i \neq b_j$ *for* $(i \neq j)$ *and* $b_i \in \mathcal{C}$ *for each* $1 \leq i \leq n$.

Lemma 3.3 (Adequation lemma) *Let* $\Gamma = \{x_i : A_i\}_{1\leq i\leq n}$, $\Delta = \{a_j : B_j\}_{1\leq j\leq m}$, $\mathcal{M} = \langle \mathcal{C}, \mathcal{S}, \{\mathcal{R}_i\}_{i\in I}\rangle$ *a model,* $\mathcal{I}$ *an* $\mathcal{M}$*-interpretation,* $u_i \in \mathcal{I}(A_i)$, $\bar{v_j} \in (\mathcal{I}(B_j))^{\perp}$, $\sigma = [(x_i := u_i)_{1\leq i\leq n}; (a_j :=^* \bar{v}_j)_{1\leq j\leq m}]$*, and* u, v *two terms such that* $v \approx_{\mathcal{C}} u$*. If* $\Gamma \vdash u : A\ ;\Delta$*, then,* $v\sigma \in \mathcal{I}(A)$*.*

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

- ax: In this case $u = x_i = v$ and $A = A_i$, then $v\sigma = u_i \in \mathcal{I}(A)$.
- $\to_i$: In this case $u = \lambda x.u_1$ and $A = B \to C$ such that $\Gamma, x : B \vdash u_1 : C\ ;\Delta$. Then $v = \lambda x.v_1$ and $v_1 \approx_{\mathcal{C}} u_1$. By the induction hypothesis, $v_1\sigma[x := w] \in \mathcal{I}(C)$ for each $w \in \mathcal{I}(B)$, hence $(\lambda x.v_1\sigma\ w) \in \mathcal{I}(C)$, therefore $\lambda x.v_1\sigma \in \mathcal{I}(B) \rightsquigarrow \mathcal{I}(C)$. Finally $v\sigma \in \mathcal{I}(A)$.
- $\to_e$: In this case $u = (u_1\ u_2)$, $\Gamma \vdash u_1 : B \to A\ ;\Delta$ and $\Gamma \vdash u_2 : B\ ;\Delta$. We have also $v = (v_1\ v_2)$ where $v_1 \approx_{\mathcal{C}} u_1$ and $v_2 \approx_{\mathcal{C}} v_1$. By the induction hypothesis, $v_1\sigma \in \mathcal{I}(B) \rightsquigarrow \mathcal{I}(A)$ and $v_2\sigma \in \mathcal{I}(B)$, therefore $(v_1\sigma\ v_2\sigma) \in \mathcal{I}(A)$, this implies that $v\sigma \in \mathcal{I}(A)$.
- μ: In this case $u = \mu a.u_1$, then $v = \mu b.v_1$ where $v_1 \approx_{\mathcal{C}} u_1$ and b is a new variable which belongs to $\mathcal{C}$ and not free in u_1 (there is always such variable because $\mathcal{C}$ is infinite). Let $\bar{v} \in (\mathcal{I}(A))^{\perp}$. By the induction hypothesis, $v_1\sigma[b :=^* \bar{v}] \in \mathcal{S}$, and, by the definition of $\mathcal{S}$, we have, $\mu b.v_1\sigma[b :=^* \bar{v}] \in \mathcal{S}$, since $(\mu b.v_1\sigma\ \bar{v}) \triangleright^* \mu b.v_1\sigma[b :=^* \bar{v}]$, then, $\mu b.v_1\sigma \in \mathcal{I}(A)$, i.e, $v\sigma \in \mathcal{I}(A)$.
- $\perp$: In this case $u = (a\ u_1)$, then, $v = (b\ v_1)$ where $v_1 \approx_{\mathcal{C}} u_1$ such that the free variable a was repalced by b in u_1 and $b \notin Fv(u_1)$ is new variable which belongs to $\mathcal{C}$. By the induction hypothesis, $v_1\sigma[b :=^* \bar{v}] \in \mathcal{I}(A)$ where $\bar{v} \in (\mathcal{I}(A))^{\perp}$, hence $(v_1\sigma[b :=^* \bar{v}]\ \bar{v}) \in \mathcal{S}$. Therefore, by the definition of $\mathcal{S}$, $(b\ (v_1\sigma[b :=^* \bar{v}]\ \bar{v})) \in \mathcal{S}$, and finally $v\sigma \in \mathcal{S}$.

■

Corollary 3.1 *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 adequation lemma, $t \in \mathcal{I}(A)$. This is true for any $\mathcal{M}$ model and for any $\mathcal{M}$-interpretation I, therefore $t \in |A|$. ■

4 Characterization of some typed terms

We start this section by adding to our system new propositional constants to obtain a new parametrized typed system. This will be useful for the proof of the lemma 4.3, which allows us to provide the syntactical proofs concerning the characterization of some typed terms. This part is inspired by Nour's works [17] and [18].

4.1 The system ${S_\mu}^{\bar{O}}$

Definition 4.1 *Let* $\bar{O} = O_1, ..., O_n$ *be a sequence of new propositional constants.*

1. *We said that* $\bar{O}$ *is different from* $\perp$ *iff each* O_i *is different from* $\perp$*.*
2. *A type* A *is said to be ending by* $\bar{O}$ *iff* A *is obtained by the following rules:*
   - *Each* O_i *ends by* $\bar{O}$*.*
   - *If* B *ends by* $\bar{O}$*, then,* $A \to B$ *ends by* $\bar{O}$*.*

3. *The typed system ${S_\mu}^{\bar{O}}$ is the system S_μ at which we add the following conditions:*

   - *The rules ax is replaced by*

   $$\frac{}{\Gamma \vdash_{\bar{O}} x_i : A_i \ ; \Delta} ax$$

   where Δ does not contain declarations of the form $a : C$ such that C ends by $\bar{O}$.

   - *The rules $\rightarrow_e$ is replaced by*

   $$\frac{\Gamma \vdash_{\bar{O}} u : A \rightarrow B; \Delta \quad \Gamma \vdash_{\bar{O}} v : A; \Delta}{\Gamma \vdash_{\bar{O}} (u\ v) : B; \Delta} \rightarrow_e$$

   where B is not ending by $\bar{O}$.

Remark 4.1 *It is obvious that ${S_\mu}^{\bar{O}}$ can be seen as a subsystem of S_μ where the syntax of formulas is extended by the new constants $\bar{O}$, therefore in the remainder of this work we consider that, any typed term in the system ${S_\mu}^{\bar{O}}$ is strongly normalizable.*

Lemma 4.1 *If $\Gamma \vdash t : A \ ; \Delta$ then $\Gamma \vdash_{\bar{O}} t : A[X := F] \ ; \Delta$ where F does not end by $\bar{O}$.*

Proof. By induction on the derivation. ■

The following lemma stipulates that the new system ${S_\mu}^{\bar{O}}$ is closed under reduction (subject reduction).

Lemma 4.2 *If $\Gamma \vdash_{\bar{O}} t : A \ ; \Delta$ and $t \triangleright^* t'$, then $\Gamma \vdash_{\bar{O}} t' : A \ ; \Delta$*

Proof. By induction on the length of the reduction $t \triangleright^* t'$. It suffices to check this result for $t \triangleright_\beta t'$ and $t \triangleright_\mu t'$. We process by induction on t. ■

Lemma 4.3 *Let $\Gamma = \{x_i : A_i\}_{1 \leq i \leq n}$, $\Delta = \{a_j : B_j\}_{1 \leq j \leq m}$ $\bar{O} = O_1, ..., O_k$ different from $\perp$ and $1 \leq l \leq k$. If $\Gamma \vdash_{\bar{O}} t : O_l \ ; \Delta$, then, $t = x_j$ for certain $1 \leq j \leq n$ and $A_j = O_l$.*

Proof. By induction on the derivation.

ax: Then, $\Gamma \vdash x_j : A_j;\ \Delta$, hence $t = x_j$ and $O_l = A_j$

$\rightarrow_i$: A contradiction because this implies that O_l is not atomic.

$\rightarrow_e$: This implies that $t = (u\ v)$, then, $\Gamma \vdash u : A \rightarrow O_l; \Delta$, therefore this gives a contradiction with the restriction on the rule $\rightarrow_e$ since O_l ends by $\bar{O}$.

μ: Then, $t = \mu a.t_1$ and $\Gamma \vdash t_1 : \perp;\ \Delta', a : O_l$, where $\Delta = \Delta' \cup \{a : O_l\}$, therefore this gives a contradiction with the fact that Δ does not contain declarations in the form $a_j : O_j$.

$\perp$: A contradiction because O_l is different from $\perp$.

■

We give now some applications of the adequation lemma.

Definition 4.2 *Let t be a term. We denote M_t the smallest set containing t such that: if $u \in M_t$ and $a \in \mathcal{A}$, then $\mu a.u \in M_t$ and $(a\ u) \in M_t$. Each element of M_t is denoted $\underline{\mu}.t$. For example, the term $\mu a.\mu b.(a\ (b\ (\mu c.(a\ \mu d.t))))$ is denoted by $\underline{\mu}.t$.*

4.2 Terms of type $\perp\to X$

Example 4.1 *Let $T_1 = \lambda z.\mu a.z$ and $T_2 = \lambda z.\mu b.(b\,\mu a.z)$, we have $\vdash T_i :\perp\to X$.*

Let then, x be λ-variable and $\bar{y}$ a finite sequence of λ-variables, we have:

- $(T_1\ x)\ \bar{y} \rhd^* \mu a.x$
- $(T_2\ x)\ \bar{y} \rhd^* \mu b.(b\ \mu a.x)$

The operational behavior of closed terms with the type $\perp\to X$ is given in the following theorem.

Theorem 4.1 *Let T be a closed term of type $\perp\to X$, then, for each λ-variable x and for each finite sequence of λ-variables $\bar{y}$, $(T\,x)\ \bar{y} \rhd^* \underline{\mu}.x$*

Proof.

Semantical proof:
Let x be a λ-variable and $\bar{y}$ a finite sequence of λ-variables. Let $\mathcal{C} = \mathcal{A}$ and take $\mathcal{S} = \{t\,/\,t \rhd^* \underline{\mu}.x\}$ and $\mathcal{R} = \{\bar{y}\} \rightsquigarrow \mathcal{S}$. It is clear that $\mathcal{S}$ is $\mathcal{C}$-saturated set and $x \in \mathcal{S}$. So let $\mathcal{M} =< \mathcal{C}, \mathcal{S}, \mathcal{R} >$ and take $\mathcal{I}$ the interpretation which at X associates $\mathcal{I}(X) = \mathcal{R}$. By the adequation lemma, $T \in \mathcal{I}(\perp\to X)$, then, $T \in \mathcal{S} \rightsquigarrow \mathcal{R}$, i.e. $T \in \mathcal{S} \rightsquigarrow (\{\bar{y}\} \rightsquigarrow \mathcal{S})$, therefore $(T\ x) \in \{\bar{y}\} \rightsquigarrow \mathcal{S}$, and $(T\ x)\ \bar{y} \in \mathcal{S}$. Finally $(T\,x)\ \bar{y} \rhd^* \underline{\mu}.x$.

Syntactical proof:
We can give also a syntactical proof of this result. Let $\bar{O} = O_1, ..., O_n$ be a sequence of new constants different from $\perp$, $A = O_1, ..., O_n \to \perp$ and $\bar{y} = y_1...y_n$ a sequence of λ-variables. By the lemma 4.1, $\vdash_{\bar{O}} T :\perp\to A$,, then, $x :\perp, (y_i : O_i)_{1\leq i\leq n} \vdash_{\bar{O}} (T\ x)\bar{y} :\perp$, hence $(T\ x)\bar{y} \rhd^* \tau$. It suffices to prove that, if τ is a normal term and $x : \perp, (y_i : O_i)_{1\leq i\leq n} \vdash_{\bar{O}} \tau : \perp\ ; (b_j : \perp)_{1\leq j\leq m}$, then $\tau = \underline{\mu}.x$. This can be proved easily by induction on τ. ■

Corollary 4.1 *Let T be a closed term of type $(\perp\to X)$, then, for each term u and for each $\bar{v} \in \mathcal{T}^{<\omega}$, $(T\,u)\ \bar{v} \rhd^* \underline{\mu}.u$*

Proof. Immediately from the previous theorem and the lemma 2.1. ■

Remark 4.2 *Let $\vdash T : \perp \to X$, the term $(T\ u)$ modelizes an instruction like* `exit`(u) (`exit` *is to be understood as in the C programming language). In the reduction of a term, if the sub-term $(T\ u)$ appears in head position (the term has the form $((T\ u)\ \bar{v})$), then after some reductions, we obtain u as result.*

4.3 Terms of type $(\neg X \to X) \to X$

Example 4.2 *Let the terms $T_1 = \lambda z.\mu a.a(z\ \lambda y.(a\ y))$ and $T_2 = \lambda z.\mu a.(a\ (z\ (\lambda s.a(z\ \lambda y.(a\,s)))))$, we have $\vdash T_i : (\neg X \to X) \to X$.*

Let x, z_1, z_2 be λ-variables and $\bar{y}$ a finite sequence of λ-variables, we have:

- $(T_1\ x)\ \bar{y} \rhd^* \mu a.a\ ((x\ \theta_1)\ \bar{y})$ *and* $(\theta_1\ z_1) \rhd^* a(z_1\ \bar{y})$.
- $(T_2\ x)\ \bar{y} \rhd^* \mu a.((a((x\ \theta_1)\ \bar{y}))\ \bar{y})$, $(\theta_1\ z_1) \rhd^* (a\ ((x\ \theta_2)\ \bar{y}))$, *and* $(\theta_2\ z_2) \rhd^* (a\ (z_1\ \bar{y}))$.

The following theorem describes the computational behavior of closed terms with type $(\neg X \to X) \to X$.

Theorem 4.2 *Let T be a closed term of type $(\neg X \to X) \to X$, then, for each λ-variable x, for each finite sequence of λ-variables $\bar{y}$ and for each sequence of λ-variables $(z_i)_{i\in\mathbb{N}^*}$ such that: x, y_j are differents from any z_i. There exist $m \in \mathbb{N}^*$ and terms $\theta_1, ..., \theta_m$, such that we have:*

- $(T\ x)\bar{y} \rhd^* \underline{\mu}.(x\ \theta_1)\ \bar{y}$
- $(\theta_k\ z_k) \rhd^* \underline{\mu}.(x\ \theta_{k+1})\ \bar{y}$ *for all* $1 \leq k \leq m-1$
- $(\theta_m\ z_m) \rhd^* \underline{\mu}.(z_l\ \bar{y})$ *for a certain* $1 \leq l \leq m$

Proof.

Semantical proof:

Let x be a λ-variable, $\bar{y}$ a finite sequence of λ-variables and $(z_i)_{i\in\mathbb{N}^*}$ a sequence of λ-variables as in the theorem. Take $\mathcal{S} = \{t\ /\ \forall\, r \geq 0,\ \exists\, m \geq 0,\ \exists\, \theta_1, ..., \theta_m,\ \exists\, j$: $t \rhd^* \underline{\mu}.((x\ \theta_1)\ \bar{y})$, $(\theta_k\ z_{k+r}) \rhd^* \underline{\mu}.((x\ \theta_{k+1})\ \bar{y})$ for every $1 \leq k \leq m-1$ and $(\theta_m\ z_{m+r}) \rhd^* \underline{\mu}.(z_j\ \bar{y})\}$ and $\mathcal{R} = \{\bar{y}\} \leadsto \mathcal{S}$.

It is important to clarify the case $m = 0$ in the definition of $\mathcal{S}$, this corresponds exaclty to $\exists\, j$: $t \rhd^* \underline{\mu}.(z_j\ \bar{y})$, thus they do not exist terms θ_i.

It is clear that $\mathcal{S}$ is a μ-saturated set. Let $\mathcal{M} = \langle \mathcal{A}, \mathcal{S}, \mathcal{R} \rangle$ and an $\mathcal{M}$-interpretation I such that $I(X) = \mathcal{R}$. By the theorem 3.1, $T \in [(\mathcal{R} \leadsto \mathcal{S}) \leadsto \mathcal{R}] \leadsto (\{\bar{y}\} \leadsto \mathcal{S})$. Let us check that $x \in (\mathcal{R} \leadsto \mathcal{S}) \leadsto \mathcal{R}$. For this, we take $\theta \in (\mathcal{R} \leadsto \mathcal{S})$ and we prove that $(x\ \theta) \in \mathcal{R}$, i.e. $((x\ \theta)\ \bar{y}) \in \mathcal{S}$. By the definition of $\mathcal{S}$, $(z_r\ \bar{y}) \in \mathcal{S}$ for each $r \geq 0$, hence $z_r \in \mathcal{R}$. Therefore $(\theta\ z_r) \in \mathcal{S}$, so we have:

$\forall r'$, $\exists m \geq 1$, $\exists \theta_1, ..., \theta_m$, $\exists j$:

- $(\theta\ z_r) \rhd^* \underline{\mu}.((x\ \theta_1)\ \bar{y})$
- $(\theta_k\ z_{k+r'}) \rhd^* \underline{\mu}.((x\ \theta_{k+1})\ \bar{y})$ for every $1 \leq k \leq m-1$
- $(\theta_m\ z_{m+r'}) \rhd^* \underline{\mu}.(z_j\ \bar{y})$.

(when $m = 0$, this gives: $\exists j : (\theta\ z_r) \rhd^* \underline{\mu}.(z_j\ \bar{y})$, then $((x\ \theta)\ \bar{y}) \rhd^* \underline{\mu}.((x\ \theta'_1)\ \bar{y})$ and $(\theta'_1\ z_r) \rhd^* \underline{\mu}.(z_j\ \bar{y})$ with $m' = 1$ and $\theta'_1 = \theta$. Therefore $((x\ \theta)\ \bar{y}) \in \mathcal{S}$).

More general, since this holds for any r', take $r' = r + 1$, then
$\exists m \geq 1$, $\exists \theta_1, ..., \theta_m$, $\exists j$:

- $(\theta\ z_r) \rhd^* \underline{\mu}.((x\ \theta_1)\ \bar{y})$
- $(\theta_k\ z_{k+1+r}) \rhd^* \underline{\mu}.((x\ \theta_{k+1})\ \bar{y})$ for every $1 \leq k \leq m-1$
- $(\theta_m\ z_{m+1+r}) \rhd^* \underline{\mu}.(z_j\ \bar{y})$.

Therefore take also $m' = m + 1$, and the terms $\theta'_1 = \theta$, $\theta'_2 = \theta_1$, ..., $\theta'_{m+1} = \theta_m$, hence check easily that we have for any fixed r:
$\exists m' \geq 1$, $\exists \theta'_1, ..., \theta'_{m'}$, $\exists j$:

- $((x\ \theta)\ \bar{y}) \rhd^* \underline{\mu}.((x\ \theta'_1)\ \bar{y})$
- $(\theta'_1\ z_r) \rhd^* \underline{\mu}.((x\ \theta'_2)\ \bar{y})$
- $(\theta'_k\ z_{k+r}) \rhd^* \underline{\mu}.((x\ \theta'_{k+1})\ \bar{y})$ for every $1 \leq k \leq m'-1$
- $(\theta'_{m'}\ z_{m'+r}) \rhd^* \underline{\mu}.(z_j\ \bar{y})$.

Thus $((x\ \theta)\ \bar{y}) \in \mathcal{S}$ which implies that $((T\,x)\,\bar{y}) \in \mathcal{S}$. By the fact that T is a closed term, the λ-variable x and the sequence $\bar{y}$ are differents from each z_i, one can ensure that the assertion $[\exists m = 0,\ \exists j : ((T\,x)\,\bar{y}) \triangleright^* \underline{\mu}.(z_j\,\bar{y})]$ can not hold. Then for $r = 0$, $\exists m \geq 1, \exists \theta_1, ..., \theta_m, \exists j$ such that:

- $((T\,x)\,\bar{y}) \triangleright^* \underline{\mu}.((x\,\theta_1)\,\bar{y})$
- $(\theta_k\ z_k) \triangleright^* \underline{\mu}.((x\,\theta_{k+1})\,\bar{y})$ for every $1 \leq k \leq m-1$
- $(\theta_m\ z_m) \triangleright^* \underline{\mu}.(z_j\,\bar{y})$ for a certain $1 \leq j \leq m$.

Syntactical proof:

We give now a syntactical proof of this result. Let $\bar{O} = O_1, ..., O_n$ be new constants different from $\perp$, $A = O_1, ..., O_n \rightarrow \perp$ and $\bar{y} = y_1 ... y_n$ a sequence of variables. By the lemma 4.1 $\vdash_{\bar{O}} T : (\neg A \rightarrow A) \rightarrow A$, then, $x : \neg A \rightarrow A, (y_i : O_i)_{1 \leq i \leq n} \vdash_{\bar{O}} (T\ x)\bar{y} : \perp$. Therefore, $(T\ x)\bar{y} \triangleright^* \tau$, where τ is a normal term and $x : \neg A \rightarrow A, (y_i : O_i)_{1 \leq i \leq n} \vdash_{\bar{O}} \tau : \perp$.

Following the form of τ we have only one case to examine, the others give always contradictions. This case is $\tau = \underline{\mu}.(x\ U_1)\ t_1 ... t_n$ where $U_1, t_1, ..., t_n$ are normal terms, $x : \neg A \rightarrow A, (y_i : O_i)_{1 \leq i \leq n} \vdash_{\bar{O}} U_1 : \neg A\ ; (b_j : \perp)_{1 \leq i \leq m}$ and for all $1 \leq k \leq n$, $x : \neg A \rightarrow A, (y_i : O_i)_{1 \leq i \leq n} \vdash_{\bar{O}} t_k : O_k\ ; (b_j : \perp)_{1 \leq j \leq m}$. We deduce, by lemma 4.3, that, for all $1 \leq k \leq n$, $t_k = y_k$.

We prove, by induction and using the lemma 4.3, that if $x : \neg A \rightarrow A, (y_i : O_i)_{1 \leq i \leq n}, (z_k : A)_{1 \leq k \leq i-1} \vdash_{\bar{O}} U_i : \neg A\ ; (b_j : \perp)_{1 \leq j \leq m}$, then

$$\begin{cases} (U_i\ z_i) \triangleright^* \underline{\mu}.(x\ U_{i+1})\bar{y} \ \textit{and}\ x : \neg A \rightarrow A, (y_i : O_i)_{1 \leq i \leq n}, (z_k : A)_{1 \leq k \leq i} \vdash_{\bar{O}} U_{i+1} : \\ \neg A\ ; (b_j : \perp)_{1 \leq j \leq m} \\ \ \textit{or} \\ \exists j : (1 \leq j \leq i),\ \textit{such that}:\ (U_i\ z_i) \triangleright^* \underline{\mu}.z_j \bar{y} \end{cases}$$

The sequence $(U_i)_{i \geq 1}$ is not infinite, else the term $((T\ \lambda x.\mu a.(x\,z))\bar{y})$ is not normalizable, which is impossible, since
$x : \neg A, z : A, (y_i : O_i)_{1 \leq i \leq n} \vdash_{\bar{O}} ((T\ \lambda x.\mu a.(x\,z))\bar{y}) : \perp$. ■

Corollary 4.2 *Let T be a closed term of type $(\neg X \rightarrow X) \rightarrow X$, then, for each term u, for each sequence $\bar{w} \in \mathcal{T}^{<\omega}$ and for each sequence $(v_i)_{i \in \mathbb{N}^*}$ of terms. There exist $m \in \mathbb{N}$ and terms $\theta_1, ..., \theta_m$ such that we have:*

- $(T\ u)\bar{w} \triangleright^* \underline{\mu}.(u\ \theta_1)\ \bar{w}$
- $(\theta_i\ v_i) \triangleright^* \underline{\mu}.(u\ \theta_{i+1})\ \bar{w}$ *for all* $1 \leq i \leq m-1$
- $(\theta_m\ v_m) \triangleright^* \underline{\mu}.(v_i\ \bar{w})$ *for some* $1 \leq i \leq m$

Proof. Immediately from the previous theorem and the lemma 2.1. ■

Remark 4.3 *Let $\vdash T : (\neg X \rightarrow X) \rightarrow X$, the term T allows to modelizing the* `Call/cc` *instruction in the Scheme functional programming language.*

5 The completeness result

The following part is devoted to the construction of the completeness model.

Definition 5.1 (*and notation*)

1. *Let $\Omega = \{x_i\ /\ i \in \mathbb{N}\} \cup \{a_j\ /\ j \in \mathbb{N}\}$ an enumeration of infinite sets of λ and μ-variables.*

2. *Let $\Omega_1 = \{A_i \;/\; i \in \mathbb{N}\}$ an enumeration of all types where each type comes an infinite times.*

3. *Let $\Omega_2 = \{B_j \;/\; j \in \mathbb{N}\}$ an enumeration of all types where $\perp$ comes an infinite times.*

4. *We define $\mathbb{G} = \{x_i : A_i \;/\; i \in \mathbb{N}\}$ and $\mathbb{D} = \{a_j : B_j \;/\; j \in \mathbb{N}\}$.*

5. *Let u be a term, such that $Fv(u) \subseteq \Omega$, 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 $Fv(u)$.*

6. *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 it exists a term u', such that $u \triangleright^* u'$ and $\mathbb{G} \vdash u' : C;\ \mathbb{D}$.*

7. *Let $\mathbb{C} = \{a_j \;/\; (a_j : \perp) \in \mathbb{D}\}$ and $\mathbb{S} = \{t \;/\; \mathbb{G} \vdash^* t : \perp;\ \mathbb{D}\}$. For each propositional variable X we define a set of terms $\mathbb{R}_X = \{t \;/\; \mathbb{G} \vdash^* t : X;\ \mathbb{D}\}$.*

Lemma 5.1 *1. $\mathbb{S}$ is a $\mathbb{C}$-saturated set.*

2. *The sets $\mathbb{R}_X$ are saturated.*

3. *For each propositional variable X, $\mathbb{R}_X = \{a_j \;/\; a_j : X \in \mathbb{D}\} \rightsquigarrow \mathbb{S}$.*

4. *$\mathbb{M} = \langle \mathbb{C}, \mathbb{S}, (\mathbb{R}_X)_{X \in \mathcal{P}} \rangle$ is a model*

Proof. Easy. ■

Remark 5.1 *Observe that the model $\mathbb{M}$ is parametrized by the infinite sets of variables and the enumerations.*

Definition 5.2 *We define the $\mathbb{M}$-interpretation $\mathbb{I}$ as follows:*

- *$\mathbb{I}(\perp) = \mathbb{S}$.*
- *$\mathbb{I}(X) = \mathbb{R}_X$ for each propositional variable.*

Lemma 5.2 *Let y be a λ-variable, $\sigma = [(x_i := y)_{1 \le i \le n}, (a_i :=^* y)_{1 \le j \le m}]$ and t a term.*

1. *If $(t\sigma\ y)$ is normalizable, then t is normalizable.*

2. *If $t\sigma$ is normalizable, then t is normalizable.*

Proof. By a simultaneous induction on t, we use the standardization theorem of the $\lambda\mu$-calculus [21].

1. We examine the case where $t = \lambda x.u$. Then $(t\sigma\ y) = (\lambda x.u\sigma\ y)$ is normalizable, this implies that $u\sigma[x := y]$ is normalizable, hence by (2), u is normalizable, therefore t is normalizable too.

2. We examine the case where $t = (a\,u)$. Then $t\sigma = (a\ (u\sigma\ y))$ is normalizable, this implies that $(u\sigma\ y)$ is normalizable, hence by (1), u is normalizable, therfore t is normalizable too.

■

Corollary 5.1 *Let t by a term and y a λ-variable. If $(t\ y)$ is normalizable, then, t is normalizable also.*

Proof. Immediately from the previous lemma. ■

Lemma 5.3 *Let t and τ be two normal terms, y a λ-variable such that $y \notin Fv(t)$, $(t\ y) \triangleright^* \tau$, A and B types, and $\Gamma, y : A \vdash \tau : B;\ \Delta$. Then $\Gamma \vdash t : A \to B;\ \Delta$.*

Proof. See the appendix. ■

Lemma 5.4 *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 the type A.

Proof of (1)

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

Proof of (2)

1. If $A = X$ or $\bot$, the result is immediately from the definition of $\mathbb{I}$.
2. Let $A = B \to C$, $t \in \mathbb{I}(B) \leadsto \mathbb{I}(C)$ and 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 the induction hypothesis (1), $y \in \mathbb{I}(B)$, then, $(t\ y) \in \mathbb{I}(C)$. By the induction hypothesis (2), $\mathbb{G} \vdash^* (t\ y) : C\,;\mathbb{D}$, then $(t\ y) \triangleright^* t'$ such that $\mathbb{G} \vdash t' : C\,;\mathbb{D}$ and, by the corollary 5.1, t is a normalizable term. The normal form of t can be either $(x\ u_1)\ u_2 ... u_n$ either $\lambda x.u$ or $\mu a.u$ (the case $(a\ u)$ gives a contradiction for typing reasons).
   (a) If $t \triangleright^* (x\ u_1)\ u_2 ... u_n$ with u_i normal terms, then $\mathbb{G} \vdash (x\ u_1)\ u_2 ... u_n y : C\,;\mathbb{D}$, $x : E_1, E_2, ..., E_n \to (B \to C) \in \mathbb{G}$, $\mathbb{G} \vdash u_i : E_i\,;\mathbb{D}$ and $\mathbb{G} \vdash y : B\,;\mathbb{D}$. Therefore $\mathbb{G} \vdash (x\ u_1)\ u_2 ... u_n : B \to C\,;\mathbb{D}$, and finally $\mathbb{G} \vdash^* t : B \to C\,;\mathbb{D}$.
   (b) If $t \triangleright^* \lambda x.u$ where u is a normal term, then, since $\mathbb{G}$ contains an infinite number of declarations for each type, let y be a λ-variable such that $y : B \in \mathbb{G}$ and $y \notin Fv(u)$. We have $(t\ y) \triangleright^* u[x := y]$ and $\mathbb{G} \vdash u[x := y] : C\,;\mathbb{D}$, hence $\mathbb{G} \vdash \lambda y.u[x := y] : B \to C\,;\mathbb{D}$ and, by the fact that $y \notin Fv(u)$, $\lambda y.u[x := y] = \lambda x.u$. Therefore $\mathbb{G} \vdash \lambda x.u : B \to C\,;\mathbb{D}$, and finally $\mathbb{G} \vdash^* t : B \to C\,;\mathbb{D}$.
   (c) If $t \triangleright^* \mu a.u$ where u is a normal term, then let y be a λ-variable such that $y : B \in \mathbb{G}$ and $y \notin Fv(u)$. We have $(t\ y) \triangleright^* \mu a.u[a :=^* y] \triangleright^* \mu a.u'$ where u' is the normal form of $u[a :=^* y]$, so we have $\mathbb{G}, y : B \vdash \mu a.u' : C\,;\mathbb{D}$. By the lemma 5.3, we obtain $\mathbb{G} \vdash \mu a.u : B \to C\,;\mathbb{D}$, and finally $\mathbb{G} \vdash^* t : B \to C\,;\mathbb{D}$.

■

Theorem 5.1 *Let A be a type and t a term. We have $t \in |A|$ iff $t \triangleright^* t'$ and $\vdash t' : A$.*

Proof. $\Leftarrow$) By the lemma 3.3.
$\Rightarrow$) We can suppose that the sets $\mathbb{G}$ and $\mathbb{D}$ do not contain declarations for the free variables of t. If $t \in |A|$, then $t \in \mathbb{I}(A)$, hence by (1) of the lemma 5.4 and by the fact that $Fv(t') \subseteq Fv(t)$, we have $t \triangleright^* t'$ and $\vdash t' : A$. ■

Corollary 5.2 *Let A be a type and t a term.*

1. *If $t \in |A|$, then t is normalizable and $t \simeq t'$, where t' is a closed term.*
2. *$|A|$ is closed under equivalence (i.e. if $t \in |A|$ and $t \simeq t'$, then, $t' \in |A|$).*

Proof. (1) is a direct consequence of theorem 5.1. (2) can be deduced from theorem 5.1 and the lemma 3.3. ■

6 Future work

Through this work, we have seen that the propositional types of the system S_μ are complete for the semantics defined previously. But what about the types of the second order typed $\lambda\mu$-calclus? We know that, for the system $\mathcal{F}$, the $\forall^+$-types (types with positive quantifiers) are complete for a realizability semantics [14] and [4]. But for the classical system $\mathcal{F}_C$, we cannot generalize this result. We check easily that, if $t = \mu a.(a\,\lambda y_1 \lambda z \mu b.(a\,\lambda y_2 \lambda x.z))$ and $A = \forall Y\{Y \to \forall X(X \to X)\}$, then $t \in |A|$, but t does not have the type A. We need to add more restrictions on the positions of the quantifier $\forall$ in the $\forall^+$-types to obtain a smallest class which we suppose that it can be proved complete. The problem is not the same when we consider the propositional classical natural deduction system with the connectives $\wedge$ and $\vee$, in previouses works [15] and [16], we define an interpretation of $\wedge$ and $\vee$ according to the functional constructors $\curlywedge$ and $\curlyvee$ respectively as follows:

- $\mathcal{K} \curlywedge \mathcal{L} = \{t \in \mathcal{T} \,/\, (t\,\pi_1) \in \mathcal{K}$ and $(t\,\pi_2) \in \mathcal{L}\}$
- $\mathcal{K} \curlyvee \mathcal{L} = \{t \in \mathcal{T} \,/$ for each u, v if (for each $r \in \mathcal{K}$, $s \in \mathcal{L}$: $u[x := r] \in \mathcal{S}$ and $v[y := s] \in \mathcal{S}$), then $(t\,[x.u, y.v]) \in \mathcal{S}\}$

Where $\mathcal{K}$, $\mathcal{L}$ are sets of terms and $\mathcal{S}$ is a particular set of terms.
These interpretations allow to obtain an adequation lemma. We can easily check that The term $\mu a.(a\ \langle \mu b.(a\ \langle \lambda x.x, \mu c.(b\ \lambda y.\lambda z.z)\rangle), \lambda x.x\rangle)$ belongs to the interpretation of the type $A = (X \to X) \wedge (X \to X)$ but it does not have the type A. The treatment of the disjunction is even hard, so we think that to circumventing this difficulties, and if we hope a completeness theorem, we have to modified the semantics.

Acknowledgements: We wish to thank R. Matthes and P. De Groote for helpful discussions.

7 Appendix

This part is devoted to the proof of the lemma 5.3.

Notation 7.1 *Let y be a λ-variable. The expression $u \triangleright_{\beta y} v$ (resp $u \triangleright_{\mu y} v$) means that we reduce in u only a β (resp μ)-redex where y is the argument, i.e. a redex in the form $(\lambda z.u\ y)$ (resp $(\mu b.u\ y)$). We denote by $\triangleright_y$ the union of $\triangleright_{\beta y}$ and $\triangleright_{\mu y}$ and $\triangleright^*_y$ (resp $\triangleright^*_{\beta y}$, $\triangleright^*_{\mu y}$) the transitive and reflexive closure of $\triangleright_y$ (resp $\triangleright_{\beta y}$, $\triangleright_{\mu y}$).*

Lemma 7.1 *Let t be a normal term, $\sigma = [(a_i :=^* y)_{1 \leq i \leq n}]$ and τ the normal form of $t\sigma$, then, $t\sigma \triangleright^*_y \tau$.*

Proof. By induction on the normal term t, the important case is the one where $t = (a_i\ u)$ and u a normal term, the others are direct consequences of the induction hypothesis. Let us examine the different forms of the normal term u, here there are two important subcases $u = \lambda x.v$ and $u = \mu b.v$ with v a normal term (these are the two cases where there is a creation of redexes after the substitution).

1. If $u = \lambda x.v$, then, $u\sigma = \lambda x.v\sigma$ and $t\sigma = (a_i\ (\lambda x.v\sigma\ \ y)) \triangleright_{\beta y} (a_i\ v\sigma[x := y])$. By the induction hypothesis, $v\sigma \triangleright^*_y v'$ where v' is the normal form of $v\sigma$, hence $(a_i\ v\sigma[x := y]) \triangleright^*_y (a_i\ v'[x := y])$ which is the normal form of $t\sigma$.

2. If $u = \mu b.v$, then, $u\sigma = \mu b.v\sigma$ and $t\sigma = (a_i\ (\mu b.v\sigma\ \ y)) \triangleright_{\mu y} (a_i\ \mu b.v\sigma[b :=^* y])$. By the induction hypothesis, $v\sigma[b :=^* y]$ is normalizable only with $\triangleright^*_y$ reductions, therefore $t\sigma$ is also normalizable only by $\triangleright^*_y$ reductions.

■

Lemma 7.2 *Let t be a normal term, τ the normal form of $t[a :=^* y]$ and A, B two types. If $\Gamma, y : A \vdash \tau : B; \Delta$. Then $\Gamma, y : A \vdash t[a :=^* y] : B; \Delta$.*

Proof. By induction on the length of the reduction $t[a :=^* y] \triangleright^*_y \tau$. By the lemma 7.1, it suffices to prove the following lemma. ■

Lemma 7.3 *Let τ be a normal term, t a term and A, B two types. If $t \triangleright_{\beta y} \tau$ (resp $t \triangleright_{\mu y} \tau$) and $\Gamma, y : A \vdash \tau : B\ ; \Delta$ then $\Gamma, y : A \vdash t : B\ ; \Delta$.*

Proof. By induction on t, we examine how $t \triangleright_{\beta y} \tau$ (resp $t \triangleright_{\mu y} \tau$). The proof is similar to the proof of (2) of the lemma 5.4. ■

Lemma 7.4 *Let t be a normal term, y a λ-variable such that $y \notin Fv(t)$, $\sigma = [a :=^* y]$ and A, B, C types. If $\Gamma, y : A \vdash t\sigma : B\ ; \Delta, a : C$, then, $\Gamma \vdash t : B\ ; \Delta, a : A \to C$.*

Proof. By induction on t.

1. $t = (x\, u_1)\, u_2 ... u_n$, then, $t\sigma = (x\, u_1\sigma)\, u_2\sigma ... u_n\sigma$ and $\Gamma, y : A \vdash (x\, u_1\sigma)\, u_2\sigma ... u_n\sigma : B\ ; \Delta, a : C$. Therefore $x : E_1, ..., E_n \to B \in \Gamma$ and $\Gamma, y : A \vdash u_i\sigma : E_i; \Delta, a : C$. By the induction hypothesis, we have $\Gamma \vdash u_i\sigma : E_i; \Delta, a : A \to C$, hence $\Gamma \vdash (x\ u_1)\, u_2 ... u_n : B\ ; \Delta, a : A \to C$.

2. $t = \lambda x.u$, then, $t\sigma = \lambda x.u\sigma$ and $\Gamma, y : A \vdash \lambda x.u\sigma : B; \Delta, a : C$, this implies that $B = F \to G$ and $\Gamma, y : A, x : F \vdash u\sigma : G; \Delta, a : C$. By the induction hypothesis, $\Gamma, x : F \vdash u : G; \Delta, a : A \to C$, then, $\Gamma \vdash \lambda x.u : F \to G; \Delta, a : A \to C$, therefore $\Gamma \vdash \lambda x.u : B; \Delta, a : A \to C$.

3. $t = \mu b.u$, then, $t\sigma = \mu b.u\sigma$ and $\Gamma, y : A \vdash \mu b.u\sigma : B\ ; \Delta, a : C$, this implies that $\Gamma, y : A \vdash u\sigma :\perp\ ; \Delta, a : C, b : B$. By the induction hypothesis, $\Gamma \vdash u :\perp\ ; \Delta, a : A \to C, b : B$, therefore $\Gamma \vdash \mu b.u : B\ ; \Delta, a : A \to C$.

4. $t = (a\ u)$, then $t\sigma = (a\ (u\sigma\ y))$ and $\Gamma, y : A \vdash (a\ (u\sigma\ y)) : \perp\ ; \Delta, a : C$, this implies that $\Gamma, y : A \vdash (u\sigma\ y) : C\ ; \Delta, a : C$ and $\Gamma, y : A \vdash u\sigma : A \to C\ ; \Delta, a : C$. By the induction hypothesis, $\Gamma \vdash u : A \to C\ ; \Delta, a : A \to C$, therefore $\Gamma \vdash (a\ u) : \perp\ ; \Delta, a : A \to C$.

5. $t = (b\ u)$, then, $t\sigma = (b\ u\sigma)$ and $\Gamma, y : A \vdash (b\ u\sigma) : \perp\ ; \Delta, a : C$, this implies that $\Gamma, y : A \vdash u\sigma : G\ ; \Delta, b : G, a : C$. By the induction hypothesis, $\Gamma \vdash u : G\ ; \Delta, b : G, a : A \to C$, therefore $\Gamma \vdash (b\ u) : \perp\ ; \Delta, a : C$.

■

Proof.[of lemma 5.3] By induction on t, the cases where $t = (x\ u_1)\, u_2 ... u_n$ and $t = \lambda x.u$ are similars to those in the proof of (2) of the lemma 5.4. Let us examine the case where $t = \mu a.u$, then $(t\ y) \triangleright^* \mu a.u[a :=^* y] \triangleright^* \mu a.u' = \tau$ where u' is the normal form of $u[a :=^* y]$. We have $\Gamma, y : A \vdash \mu a.u' : B;\ \Delta$, then $\Gamma, y : A \vdash u' : \perp;\ \Delta, a : B$. By the lemma 7.1, $u[a :=^* y] \triangleright_y^* u'$, then, by the lemma 7.2, $\Gamma, y : A \vdash u[a :=^* y] : \perp;\ \Delta, a : B$. Hence by the lemma 7.4, $\Gamma \vdash u : \perp;\ \Delta, a : A \to B$ and finally $\Gamma \vdash \mu a.u : A \to B;\ \Delta$. ■

References

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

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

[3] R. David. *Une preuve simple de résultats classiques en λ-calcul.* Compte Rendu de l'Académie des Sciences. Paris, Tome 320, Série 1, pp. 1401-1406, 1995.

[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. *Types Complets dans une extension du système* $\mathcal{AF}2$. Informatique Théorique et Application, 31-6, pp. 513-537, 1998.

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

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

[8] J. R. Hindley. *The simple semantics for Coppe-Dezani-Sallé types.* Proceeding of the 5th Colloquium on International Symposium on Programming, pp. 212-226, April 06-08, 1982.

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

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

[11] F. Kamareddine and K. Nour. *A completeness result for a realizability semantics for an intersection type system.* Submitted.

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

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

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

[15] 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.

[16] 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.

[17] K. Nour. *Opérateurs de mise en mémoire et types ∀-positifs.* Theoretical Informatics and Applications, vol 30, n 3, pp. 261-293, 1996.

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

[19] 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.

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

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

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

[23] 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.

[24] 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.