



HAL
open science

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

Karim Nour, Khelifa Saber

► **To cite this version:**

Karim Nour, Khelifa Saber. A completeness result for the simply typed $\lambda\mu$ -calculus. *Annals of Pure and Applied Logic*, 2009, 161, pp.109-118. hal-00380646

HAL Id: hal-00380646

<https://hal.science/hal-00380646>

Submitted on 4 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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@univ-savoie.fr ksaber@messel.emse.fr}

Abstract

In this paper, we define a 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. This result serves to give characterizations of the computational behavior of some closed typed terms. We also prove a completeness result of our realizability semantics using a particular term model.

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 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 good properties: the subject reduction, the strong normalization and confluence theorems.

The strong normalization theorem of second order classical natural deduction [20] is based on a lemma known as the correctness result, 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 that *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 his 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 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 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 [12]. This inspired us also to define a general semantics for classical natural deduction in [15] and gave such characterizations.

The question that we now can ask is: “does the correctness result have a converse?”. By this we mean: “can we find a class of types for which the converse of the correctness result (completeness result) holds?”. J.R. Hindley was the first who

study the completeness of simple type systems [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 strictly positive types, and this for a semantics based on sets stable under $\beta\eta$ -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 deal with this problem and prove the completeness for the simply typed $\lambda\mu$ -calculus. The semantics that we define here is not completely different from that of [15] and [16], nevertheless we add a slight but an indispensable modification to the notion of the μ -saturation. 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.

The correctness result allows to describe the computational behavior of closed typed terms. We have two kinds of proofs for such characterizations. Semantical proofs, in which we guess the computational behaviors, models used in such proofs are exactly built to meet the required characterization. Syntactical proofs, where we construct the behavior based on the type, these proofs are shorter 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 its correctness. Section 4 is devoted to the completeness result. Finally, in Section 5 we give characterizations of some closed typed terms.

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

In this work, we use the $\lambda\mu$ -calculus à la De Groote, where the binder μ and the naming construct are split. This allows more expressivity than the Parigot’s original version.

Definition 2.1 *1. Let \mathcal{X} 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:*

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

2. *Types are formulas of the propositional logic built from the infinite set of propositional variables $\mathcal{P} = \{X, Y, Z, \dots\}$ and a constant of type \perp , using the connective \rightarrow .*
3. *As usual we denote by $\neg A$ the formula $A \rightarrow \perp$. Let A_1, A_2, \dots, A_n, A be types, we denote the type $A_1 \rightarrow (A_2 \rightarrow (\dots \rightarrow (A_n \rightarrow A)\dots))$ by $A_1, A_2, \dots, A_n \rightarrow A$.*
4. *Proofs are presented in natural deduction system with two 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.*

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 $\Gamma \vdash t : A ; \Delta$.

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

$$\frac{\Gamma, x : A \vdash t : B ; \Delta}{\Gamma \vdash \lambda x.t : A \rightarrow B ; \Delta} \rightarrow_i \quad \frac{\Gamma \vdash u : A \rightarrow B ; \Delta \quad \Gamma \vdash v : A ; \Delta}{\Gamma \vdash (u v) : B ; \Delta} \rightarrow_e$$

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

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 details, 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, \dots, u_n be terms and $\bar{v}_1, \dots, \bar{v}_m$ 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 $(a_j (u \bar{v}_j))$.

Remark 2.1 *In order to avoid the heavy notation of the substitution $[(x_i := u_i)_{1 \leq i \leq n}; (a_j :=^* \bar{v}_j)_{1 \leq j \leq m}]$, we denote it by σ (which is not an object of the syntax). Then $t[(x_i := u_i)_{1 \leq i \leq n}; (a_j :=^* \bar{v}_j)_{1 \leq j \leq m}]$ is denoted by $t\sigma$.*

Lemma 2.1 *Let t, t' be terms and σ a substitution, if $t \triangleright^* t'$, then, $t\sigma \triangleright^* t'\sigma$.*

Proof. By induction on t . ■

3 The semantics of S_μ

In this part we define the realizability semantics and prove its correctness.

Definition 3.1 1. We say that a set of terms \mathcal{S} is saturated when the conditions: $v \triangleright^* u$ and $u \in \mathcal{S}$ imply $v \in \mathcal{S}$ for all terms u and v .

2. Let us take a saturated set of terms \mathcal{S} and a set \mathcal{C} of an infinite classical variables (μ -variables). We say that \mathcal{S} is \mathcal{C} -saturated when the condition: $t \in \mathcal{S}$ implies $\mu a.t \in \mathcal{S}$ and $(a t) \in \mathcal{S}$ for all term t and all μ -variable $a \in \mathcal{C}$

Remark 3.1 The difference between this semantics and those defined in [15] and [16], is the notion of the \mathcal{C} -saturation which is not necessary for the correctness part, but indispensable for the completeness side. It is obvious that this notion introduces ill-typed terms, thing which seems to go against completeness. Nevertheless, the key point is that \mathcal{C} is a parameter attached to a particular model, therefore when we take the intersection of all models, all these bad terms are removed. This is exactly what is done in the proof of the theorem 4.1.

Definition 3.2 1. Consider two sets of terms \mathcal{K} and \mathcal{L} , we define a new set of terms: $\mathcal{K} \rightsquigarrow \mathcal{L} = \{t / (t u) \in \mathcal{L}, \text{ for each } u \in \mathcal{K}\}$. It is clear that when \mathcal{L} is a saturated set, then $\mathcal{K} \rightsquigarrow \mathcal{L}$ is also saturated one.

2. We denote $\mathcal{T} \cup \mathcal{A}$ by \mathcal{T}' and $\mathcal{T}'^{<\omega}$ the set of finite sequences of elements 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 $\pi = a\pi'$.

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

Remark 3.2 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 separation theorem in the $\lambda\mu$ -calculus.

Definition 3.3 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 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 $\mathcal{X}_{\mathcal{G}} = \{\emptyset\}$.
- 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 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 $\mathcal{X}_{\mathcal{G}} = \{u\bar{v} / u \in \mathcal{G}_1 \text{ and } \bar{v} \in \mathcal{X}_{\mathcal{G}_2}\}$.

■

Definition 3.4 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. Immediate. ■

Definition 3.5 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:

- $\mathcal{I}(\perp) = \mathcal{S}$
- $\mathcal{I}(A \rightarrow B) = \mathcal{I}(A) \rightsquigarrow \mathcal{I}(B)$.

2. For any type A , we denote $|A|_{\mathcal{M}} = \bigcap\{\mathcal{I}(A) / \mathcal{I} \text{ an } \mathcal{M}\text{-interpretation}\}$.

3. For any type A , $|A| = \bigcap\{|A|_{\mathcal{M}} / \mathcal{M} \text{ a model}\}$.

The notion of \mathcal{C} -saturation is indispensable for completeness but, as we said in the remark 3.1, it provides ill-terms. The presence of such terms has some drawbacks on the correctness side, hence we introduce in the following definition a parameterized relation $\hookrightarrow_{\mathcal{C}}$.

Definition 3.6 Let u, v be two terms. The expression $u \hookrightarrow_{\mathcal{C}} v$ 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 $u[a_1, \dots, a_n]$ where the a_i are the free classical variables of u , then v will be $u[a_1 := b_1, \dots, a_n := b_n]$ where $b_i \neq b_j$ for $(i \neq j)$ and $b_i \in \mathcal{C}$ for each $1 \leq i \leq n$ (it is obvious that $\hookrightarrow_{\mathcal{C}}$ is parameterized by \mathcal{C}).

Lemma 3.3 (Correctness) 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 $u \hookrightarrow_{\mathcal{C}} v$. 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 $v\sigma = u_i \in \mathcal{I}(A)$.

\rightarrow_i : In this case $u = \lambda x.u_1$ and $A = B \rightarrow C$ such that $\Gamma, x : B \vdash u_1 : C ; \Delta$. Then $v = \lambda x.v_1$ and $u_1 \hookrightarrow_{\mathcal{C}} v_1$. Let $w \in \mathcal{I}(B)$ and $\delta = \sigma + [x := w]$, by induction hypothesis, $v_1\delta \in \mathcal{I}(C)$, 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 $v\sigma \in \mathcal{I}(A)$.

\rightarrow_c : In this case $u = (u_1 u_2)$, $\Gamma \vdash u_1 : B \rightarrow A ; \Delta$ and $\Gamma \vdash u_2 : B ; \Delta$. We also have $v = (v_1 v_2)$ where $u_1 \hookrightarrow_{\mathcal{C}} v_1$ and $u_2 \hookrightarrow_{\mathcal{C}} v_2$. By 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 $u_1 \hookrightarrow_{\mathcal{C}} v_1$ 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 $\bar{v} \in (\mathcal{I}(A))^\perp$ and $\delta = \sigma + [b := \bar{v}]$. By induction hypothesis, $v_1\delta \in \mathcal{S}$, and by the definition of \mathcal{S} , we have, $\mu b.v_1\delta \in \mathcal{S}$. Since $(\mu b.v_1\sigma \bar{v}) \triangleright^* \mu b.v_1\delta$, 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 $u_1 \hookrightarrow_{\mathcal{C}} v_1$ such that the free variable a was replaced by b in u_1 and $b \notin Fv(u_1)$ is new variable which belongs to \mathcal{C} . Let $\delta = \sigma + [b := \bar{v}]$ where $\bar{v} \in (\mathcal{I}(A))^\perp$, by induction hypothesis, $v_1\delta \in \mathcal{I}(A)$, hence $(v_1\delta \bar{v}) \in \mathcal{S}$. Therefore, by the definition of \mathcal{S} , $(b (v_1\delta \bar{v})) \in \mathcal{S}$, 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 lemma 3.3, $t \in \mathcal{I}(A)$. This is true for any model \mathcal{M} and for any \mathcal{M} -interpretation \mathcal{I} , therefore $t \in |A|$. ■

4 The completeness result

Roughly speaking, completeness of the semantics amounts to saying that if t is in the interpretation of a type A , then t has the type A . In order to prove the completeness result, we construct in the following part a particular term model.

Definition 4.1 (*and notation*)

1. Let $\Omega = \{x_i / i \in \mathbb{N}\} \cup \{a_j / j \in \mathbb{N}\}$ be an enumeration of infinite sets of λ and μ -variables.
2. Let $\Omega_1 = \{A_i / i \in \mathbb{N}\}$ be an enumeration of all types where each type comes infinitely many times.
3. Let $\Omega_2 = \{B_j / j \in \mathbb{N}\}$ be an enumeration of all types where the type \perp comes infinitely many 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 there 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 $\mathbb{S} = \{t / \mathbb{G} \vdash^* t : \perp; \mathbb{D}\}$.
8. For each propositional variable X , we define a set of terms $\mathbb{R}_X = \{t / \mathbb{G} \vdash^* t : X; \mathbb{D}\}$.

Lemma 4.1 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 4.1 *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 the theorem 4.1.*

Definition 4.2 *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 4.2 *Let y be a λ -variable, $\sigma = [(x_i := y)_{1 \leq i \leq n}, (a_i :=^* y)_{1 \leq j \leq m}]$ a substitution 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, therefore t is normalizable too. ■

Corollary 4.1 *Let t be a term and y a λ -variable. If $(t y)$ is normalizable, then, t is also normalizable.*

Proof. Immediate from the previous lemma. ■

Lemma 4.3 *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 $\Gamma \vdash t : A \rightarrow B; \Delta$.*

Proof. See the appendix. ■

Lemma 4.4 *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 \perp , the result is immediate from the definition of \mathbb{I} .
2. Let $A = B \rightarrow C$ and $\mathbb{G} \vdash^* t : A; \mathbb{D}$, then $t \triangleright^* t'$ such that: $\mathbb{G} \vdash t' : B \rightarrow C; \mathbb{D}$. Let $u \in \mathbb{I}(B)$. By induction hypothesis (2), we have $\mathbb{G} \vdash^* u : B; \mathbb{D}$, this implies that $u \triangleright^* u'$ 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 induction hypothesis (1), $(t u) \in \mathbb{I}(C)$. Therefore $t \in \mathbb{I}(B \rightarrow C)$.

Proof of (2)

1. If $A = X$ or \perp , the result is immediate from the definition of \mathbb{I} .
2. Let $A = B \rightarrow C$, $t \in \mathbb{I}(B) \rightsquigarrow \mathbb{I}(C)$ 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 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 $(t y) \triangleright^* t'$ such that $\mathbb{G} \vdash t' : C; \mathbb{D}$ and, by the corollary 4.1, t is a normalizable term. The normal form of t can be either $(x u_1) u_2 \dots u_n$ either $\lambda x.u$ or $\mu a.u$ (the case $(a u)$ gives a contradiction for typing reasons).

- (a) If $t \triangleright^* (x u_1) u_2 \dots u_n$ with u_i normal terms, then $\mathbb{G} \vdash (x u_1) u_2 \dots u_n y : C ; \mathbb{D}, x : E_1, E_2, \dots, E_n \rightarrow (B \rightarrow C) \in \mathbb{G}, \mathbb{G} \vdash u_i : E_i ; \mathbb{D}$ and $\mathbb{G} \vdash y : B ; \mathbb{D}$. Therefore $\mathbb{G} \vdash (x u_1) u_2 \dots u_n : B \rightarrow C ; \mathbb{D}$, finally $\mathbb{G} \vdash^* t : B \rightarrow C ; \mathbb{D}$.
- (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 $\mathbb{G} \vdash \lambda y.u[x := y] : B \rightarrow C ; \mathbb{D}$ and, by the fact that $y \notin Fv(u)$, $\lambda y.u[x := y] = \lambda x.u$. Therefore $\mathbb{G} \vdash \lambda x.u : B \rightarrow C ; \mathbb{D}$, finally $\mathbb{G} \vdash^* t : B \rightarrow C ; \mathbb{D}$.
- (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 4.3, we obtain $\mathbb{G} \vdash \mu a.u : B \rightarrow C ; \mathbb{D}$, finally $\mathbb{G} \vdash^* t : B \rightarrow C ; \mathbb{D}$. ■

Theorem 4.1 *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 the lemma 3.3.

\Rightarrow) We consider an infinite set of λ and μ variables Ω such that it contains none of the free variables of t , then from this set we build the completeness model as described in the definition 4.1. If $t \in |A|$, then $t \in \mathbb{I}(A)$, hence by (1) of the lemma 4.4 and by the fact that $Fv(t') \subseteq Fv(t)$, we have $t \triangleright^* t'$ and $\vdash t' : A$. ■

Corollary 4.2 *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. *$|A|$ is closed under equivalence.*

Proof. (1) and (2) are direct consequences of theorem 4.1. (3) can be deduced from the theorem 4.1 and the lemma 3.3. ■

5 Characterization of some typed terms

We begin by adding to our system new propositional constants to obtain a new parameterized typed system. In such systems we can characterize the syntactical form of a term having some type, this will be useful for the proof of the lemma 5.3. This part is inspired by Nour's works [17] and [18].

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

Definition 5.1 *Let $\bar{O} = O_1, \dots, O_n$ be a sequence of fresh propositional constants.*

1. *A type A is said an \bar{O} -type iff A is obtained by the following rules:*
 - *Each O_i is an \bar{O} -type.*
 - *If B is an \bar{O} -type, then, $A \rightarrow B$ is an \bar{O} -type.*
2. *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 is an \bar{O} -type.

- 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 an \bar{O} -type.

Remark 5.1 It is obvious that $S_\mu^{\bar{O}}$ can be seen as the system S_μ where the syntax of formulas is extended by the new constants \bar{O} and some restrictions are imposed on the typing rules. Therefore in the remainder of this work we consider that, any typed term in the system $S_\mu^{\bar{O}}$ is strongly normalizable.

Lemma 5.1 If $\Gamma \vdash t : A ; \Delta$, X a propositional variable and F is not an \bar{O} -type, then $\Gamma \vdash_{\bar{O}} t : A[X := F] ; \Delta$.

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 5.2 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 proceed by induction on t . ■

Lemma 5.3 Let $\Gamma = \{x_i : A_i\}_{1 \leq i \leq n}$, $\Delta = \{a_j : B_j\}_{1 \leq j \leq m}$, $\bar{O} = O_1, \dots, O_k$ and $1 \leq l \leq k$. If $\Gamma \vdash_{\bar{O}} t : O_l ; \Delta$, then, $t = x_j$ for some $1 \leq j \leq n$ 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 is an \bar{O} -type.

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 of the form $a_j : O_j$.

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

Now we give some applications of the lemma 3.3. We will see that the operational behavior of a typed term depends in “certain sense” only of its type.

Definition 5.2 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 $\underline{\mu}.t$.

5.2 Terms of type $\perp \rightarrow X$

Example 5.1 Let $e_1 = \lambda x. \mu a. x$ and $e_2 = \lambda x. \mu b. (b \mu a. x)$, we have:
 $\vdash e_i : \perp \rightarrow X$.

Given a λ -variable x , and a finite sequence of λ -variables \bar{y} , we have:

- $(e_1 x) \bar{y} \triangleright^* \mu a. x$
- $(e_2 x) \bar{y} \triangleright^* \mu b. (b \mu a. x)$

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

Theorem 5.1 Let e be a closed term of type $\perp \rightarrow X$, then, for each λ -variable x and for each finite sequence of λ -variables \bar{y} , $(e x) \bar{y} \triangleright^* \underline{\mu}.x$

Proof.

Semantical proof:

Let x be a λ -variable and \bar{y} a finite sequence of λ -variables. Let $\mathcal{C} = \mathcal{A}$, take $\mathcal{S} = \{t / t \triangleright^* \underline{\mu}.x\}$ 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 $\mathcal{M} = \langle \mathcal{C}, \mathcal{S}, \mathcal{R} \rangle$ and take \mathcal{I} the interpretation which at X associates $\mathcal{I}(X) = \mathcal{R}$. By the lemma 3.3, $e \in \mathcal{I}(\perp \rightarrow X)$, then, $e \in \mathcal{S} \rightsquigarrow \mathcal{R}$, i.e., $e \in \mathcal{S} \rightsquigarrow (\{\bar{y}\} \rightsquigarrow \mathcal{S})$, therefore $(e x) \in \{\bar{y}\} \rightsquigarrow \mathcal{S}$, and $(e x) \bar{y} \in \mathcal{S}$. Finally $(e x) \bar{y} \triangleright^* \underline{\mu}.x$.

Syntactical proof:

We can also give a syntactical proof of this result. Let $\bar{O} = O_1, \dots, O_n$ be a sequence of new constants, $A = O_1, \dots, O_n \rightarrow \perp$ and $\bar{y} = y_1 \dots y_n$ a sequence of λ -variables. By the lemma 5.1, $\vdash_{\bar{O}} e : \perp \rightarrow A$, then, $x : \perp, (y_i : O_i)_{1 \leq i \leq n} \vdash_{\bar{O}} (e x) \bar{y} : \perp$, hence $(e x) \bar{y} \triangleright^* \tau$. 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 5.1 Let e be a closed term of type $(\perp \rightarrow X)$, then, for each term u and for each $\bar{v} \in \mathcal{T}^{<\omega}$, $(e u) \bar{v} \triangleright^* \underline{\mu}.u$

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

Remark 5.2 Let $\vdash e : \perp \rightarrow X$, the term $(e u)$ modelizes an instruction like `exit(u)` (`exit` is to be understood as in the \mathbb{C} programming language). In the reduction of a term, if the subterm $(e u)$ appears in head position (the term has the form $((e u) \bar{v})$), then, after some reductions, the sequence \bar{v} is deleted, and we obtain $\underline{\mu}.u$ as result.

5.3 Terms of type $(\neg X \rightarrow X) \rightarrow X$

Example 5.2 Let the terms $E_1 = \lambda x. \mu a. (a (x \lambda z. (a z)))$
and $E_2 = \lambda x. \mu a. (a (x (\lambda z_1. (a (x \lambda z_2. (a z_1))))))$,
we have: $\vdash E_i : (\neg X \rightarrow X) \rightarrow X$.

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

- $(E_1 x) \bar{y} \triangleright^* \mu a. (a ((x \theta_1) \bar{y}))$ and $(\theta_1 z_1) \triangleright^* (a (z_1 \bar{y}))$, where $\theta_1 = \lambda z. (a (z \bar{y}))$.
- $(E_2 x) \bar{y} \triangleright^* \mu a. (a ((x \theta_1) \bar{y}))$, $(\theta_1 z_1) \triangleright^* (a ((x \theta_2) \bar{y}))$, and $(\theta_2 z_2) \triangleright^* (a (z_1 \bar{y}))$, where $\theta_1 = \lambda z_1. (a ((x \lambda z_2. (a (z_1 \bar{y}))) \bar{y}))$ and $\theta_2 = \lambda z_2. (a (z_1 \bar{y}))$.

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

Theorem 5.2 *Let E be a closed term of type $(\neg X \rightarrow X) \rightarrow X$, 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 different from any z_i . There exist $m \in \mathbb{N}^*$ and terms $\theta_1, \dots, \theta_m$, such that we have:*

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

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 above. Take $\mathcal{S} = \{t \mid \forall r \geq 0: \text{Either } [\exists m \geq 1, \exists \theta_1, \dots, \theta_m, \exists j: t \triangleright^* \underline{\mu}.((x \theta_1) \bar{y}), (\theta_k z_{k+r}) \triangleright^* \underline{\mu}.((x \theta_{k+1}) \bar{y}) \text{ for every } 1 \leq k \leq m - 1 \text{ and } (\theta_m z_{m+r}) \triangleright^* \underline{\mu}.(z_j \bar{y})], \text{ or } [\exists j: t \triangleright^* \underline{\mu}.(z_j \bar{y})]\}$, take also $\mathcal{R} = \{\bar{y}\} \rightsquigarrow \mathcal{S}$.

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 corollary 3.1, $E \in [(\mathcal{R} \rightsquigarrow \mathcal{S}) \rightsquigarrow \mathcal{R}] \rightsquigarrow (\{\bar{y}\} \rightsquigarrow \mathcal{S})$. Let us check that $x \in (\mathcal{R} \rightsquigarrow \mathcal{S}) \rightsquigarrow \mathcal{R}$. For this, we take $\theta \in (\mathcal{R} \rightsquigarrow \mathcal{S})$ 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 $\forall r' \geq 0$:

1. Either $\exists m \geq 1, \exists \theta_1, \dots, \theta_m, \exists j :$

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

More generally, since this holds for any r' , take $r' = r + 1$, then,

$\exists m \geq 1, \exists \theta_1, \dots, \theta_m, \exists j :$

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

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

$\exists m' \geq 1, \exists \theta'_1, \dots, \theta'_{m'}, \exists j :$

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

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

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

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

Syntactical proof:

Now we give a syntactical proof of this result. Let $\bar{O} = O_1, \dots, O_n$ be new constants, $A = O_1, \dots, O_n \rightarrow \perp$ and $\bar{y} = y_1 \dots y_n$ a sequence of variables. By the lemma 5.1 $\vdash_{\bar{O}} E : (\neg A \rightarrow A) \rightarrow A$, then, $x : \neg A \rightarrow A, (y_i : O_i)_{1 \leq i \leq n} \vdash_{\bar{O}} (E x) \bar{y} : \perp$. 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 $\tau = \underline{\mu}(x U_1) t_1 \dots t_n$ where U_1, t_1, \dots, t_n are normal terms, $x : \neg A \rightarrow A, (y_i : O_i)_{1 \leq i \leq n} \vdash_{\bar{O}} U_1 : \neg A ; (b_j : \perp)_{1 \leq j \leq m}$ 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 the lemma 5.3, that, for all $1 \leq k \leq n$, $t_k = y_k$.

We prove, by induction and using the lemma 5.3, 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

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

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

Corollary 5.2 *Let E 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 $\theta_1, \dots, \theta_m$ such that we have:*

- $(E u) \bar{w} \triangleright^* \underline{\mu}(u \theta_1) \bar{w}$
- $(\theta_i v_i) \triangleright^* \underline{\mu}(u \theta_{i+1}) \bar{w}$ 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. Immediate from the previous theorem and the lemma 2.1. ■

Remark 5.3 *In the C programming language, there exist “escape” instructions which allow to manage errors without stopping the program. These are `setjmp` and `longjmp`. If we reduce $(E_1 \lambda y. h) \bar{w}$, we obtain $\mu a.(a(h[y := \theta_1] \bar{w}))$. When θ is executed with some value v , the environment is restored and we get $(a.(v \bar{w}))$. In other words, in the term $(E_1 \lambda y. h)$, E_1 plays the role of the `setjmp` instruction and occurrences of the variables y in h are the `longjmp` instruction. The corollary 5.2 says that every term of type $(\neg X \rightarrow X) \rightarrow X$ has the same operational behavior of E_1 but often in several steps (the sequence of θ_i).*

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.

1. 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] and [14]). But for the classical system \mathcal{F} , 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 $A = \forall Y\{Y \rightarrow \forall X(X \rightarrow X)\}$, then $t \in |A|$, but t does not have the type A . This is due to the presence of \forall in right-hand-side of \rightarrow , hence, we need to add more restrictions on the positions of \forall in the \forall^+ -types to obtain a smallest class of type that we suppose can be proved complete.
2. The problem is not the same when we consider the propositional classical natural deduction system with the connectives \wedge and \vee . In previous works [15] and [16], we define interpretations of \wedge and \vee according to the functional constructors λ and Υ respectively as follows:

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

These interpretations allow to obtain a correctness result. 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 $A = (X \rightarrow X) \wedge (X \rightarrow X)$ but it does not have the type A . The treatment of the disjunction is even a delicate matter, so we think that to circumventing this difficulties, and if we hope a completeness theorem, some deep modifications should be brought to our 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 4.3.

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 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 creation of redexes after 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 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 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 4.4. ■

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, $\Gamma \vdash t : B; \Delta, a : A \rightarrow C$.*

Proof. By induction on t .

1. $t = (x u_1) u_2 \dots u_n$, then, $t\sigma = (x u_1 \sigma) u_2 \sigma \dots u_n \sigma$ and $\Gamma, y : A \vdash (x u_1 \sigma) u_2 \sigma \dots u_n \sigma : B; \Delta, a : C$. Therefore $x : E_1, \dots, E_n \rightarrow B \in \Gamma$ and $\Gamma, y : A \vdash u_i \sigma : E_i; \Delta, a : C$. By induction hypothesis, we have $\Gamma \vdash u_i \sigma : E_i; \Delta, a : A \rightarrow C$, hence $\Gamma \vdash (x u_1) u_2 \dots u_n : B; \Delta, a : A \rightarrow C$.
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 $B = F \rightarrow G$ and $\Gamma, y : A, x : F \vdash u\sigma : G; \Delta, a : C$. By induction hypothesis, $\Gamma, x : F \vdash u : G; \Delta, a : A \rightarrow C$, then, $\Gamma \vdash \lambda x. u : F \rightarrow G; \Delta, a : A \rightarrow C$, therefore $\Gamma \vdash \lambda x. u : B; \Delta, a : A \rightarrow C$.
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 induction hypothesis, $\Gamma \vdash u : \perp; \Delta, a : A \rightarrow C, b : B$, therefore $\Gamma \vdash \mu b. u : B; \Delta, a : A \rightarrow C$.
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 $\Gamma, y : A \vdash u\sigma : A \rightarrow C; \Delta, a : C$. By induction hypothesis, $\Gamma \vdash u : A \rightarrow C; \Delta, a : A \rightarrow C$, therefore $\Gamma \vdash (a u) : \perp; \Delta, a : A \rightarrow C$.
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 induction hypothesis, $\Gamma \vdash u : G; \Delta, b : G, a : A \rightarrow C$, therefore $\Gamma \vdash (b u) : \perp; \Delta, a : C$.

Proof.[of lemma 4.3] By induction on t , the cases where $t = (x u_1) u_2 \dots u_n$ and $t = \lambda x. u$ are similar to those in the proof of (2) of the lemma 4.4. 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, $\Gamma \vdash u : \perp; \Delta, a : A \rightarrow B$ finally $\Gamma \vdash \mu a. u : A \rightarrow B; \Delta$. ■

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 $\lambda\mu$ -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*. Annals of Pure and Applied Logic, vol 146, pp. 180-198, 2007
- [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 \forall -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.