



HAL
open science

Arithmetical proofs of strong normalization results for the symmetric $\lambda\mu$ -calculus

René David, Karim Nour

► **To cite this version:**

René David, Karim Nour. Arithmetical proofs of strong normalization results for the symmetric $\lambda\mu$ -calculus. *Typed Lambda Calculi and Applications*, Apr 2005, Nara, Japan. pp.162-178. hal-00382299

HAL Id: hal-00382299

<https://hal.science/hal-00382299>

Submitted on 7 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.

Arithmetical proofs of strong normalization results for the symmetric $\lambda\mu$ -calculus

René David and Karim Nour
Laboratoire de Mathématiques
Université de Savoie
73376 Le Bourget du Lac. France
e-mail: {david,nour}@univ-savoie.fr

Abstract. The symmetric $\lambda\mu$ -calculus is the $\lambda\mu$ -calculus introduced by Parigot in which the reduction rule μ' , which is the symmetric of μ , is added. We give *arithmetical* proofs of some strong normalization results for this calculus. We show (this is a new result) that the $\mu\mu'$ -reduction is strongly normalizing for the un-typed calculus. We also show the strong normalization of the $\beta\mu\mu'$ -reduction for the typed calculus: this was already known but the previous proofs use candidates of reducibility where the interpretation of a type was defined as the fix point of some increasing operator and thus, were highly non arithmetical.

1 Introduction

Since it has been understood that the Curry-Howard isomorphism relating proofs and programs can be extended to classical logic, various systems have been introduced: the λ_c -calculus (Krivine [12]), the λ_{exn} -calculus (de Groote [6]), the $\lambda\mu$ -calculus (Parigot [18]), the λ^{Sym} -calculus (Barbanera & Berardi [1]), the λ_Δ -calculus (Rehof & Sorensen [24]), the $\bar{\lambda}\mu\tilde{\mu}$ -calculus (Curien & Herbelin [3]), ...

The first calculus which respects the intrinsic symmetry of classical logic is λ^{Sym} . It is somehow different from the previous calculi since the main connector is not the arrow as usual but the connectors *or* and *and*. The symmetry of the calculus comes from the de Morgan laws.

The second calculus respecting this symmetry has been $\bar{\lambda}\mu\tilde{\mu}$. The logical part is the (classical) sequent calculus instead of natural deduction.

Natural deduction is not, intrinsically, symmetric but Parigot has introduced the so called *Free deduction* [17] which is completely symmetric. The $\lambda\mu$ -calculus comes from there. To get a confluent calculus he had, in his terminology, to fix the inputs on the left. To keep the symmetry, it is enough to keep the same terms and to add a new reduction rule (called the μ' -reduction) which is the symmetric rule of the μ -reduction and also corresponds to the elimination of a cut. We get then a symmetric calculus that is called the *symmetric $\lambda\mu$ -calculus*.

The μ' -reduction has been considered by Parigot for the following reasons. The $\lambda\mu$ -calculus (with the β -reduction and the μ -reduction) has good properties : confluence in the un-typed version, subject reduction and strong normalization in the typed calculus. But this system has, from a computer science point of view, a drawback: the unicity of the representation of data is lost. It is known that, in the λ -calculus, any term of type N (the usual type for the integers) is β -equivalent to a Church integer. This no more true in the $\lambda\mu$ -calculus and we can find normal terms of type N that are not Church integers. Parigot has remarked that by adding the μ' -reduction and some simplification rules the unicity of the representation of data is recovered and subject reduction is preserved, at least for the simply typed system, even though the confluence is lost.

Barbanera & Berardi proved the strong normalization of the λ^{Sym} -calculus by using candidates of reducibility but, unlike the usual construction (for example for Girard's system F), the definition of the interpretation of a type needs a rather complex fix-point operation. Yamagata [25] has used the same technic to prove the strong normalization of the symmetric $\lambda\mu$ -calculus where the types are those of system F and Parigot, again using the same ideas, has extended Barbanera & Berardi's result to a logic with second order quantification. These proofs are thus highly non arithmetical.

We consider here the $\lambda\mu$ -calculus with the rules β , μ and μ' . It was known that, for the un-typed calculus, the μ -reduction is strongly normalizing (see [23]) but the strong normalization of the $\mu\mu'$ -reduction for the un-typed calculus was an open problem raised long ago by Parigot. We give here a proof of this result. Studying this reduction by itself is interesting since a μ (or μ')-reduction can be seen as a way "to put the arguments of the μ where they are used" and it is useful to know that this is terminating. We also give an *arithmetical* proof of the strong normalization of the $\beta\mu\mu'$ -reduction for the simply typed calculus. We finally show (this is also a new result) that, in the un-typed calculus, if M_1, \dots, M_n are strongly normalizing for the $\beta\mu\mu'$ -reduction, then so is $(x M_1 \dots M_n)$.

The proofs of strong normalization that are given here are extensions of the ones given by the first author for the simply typed λ -calculus. This proof can be found either in [7] (where it appears among many other things) or as a simple unpublished note on the web page of the first author (www.lama.univ-savoie.fr/~david).

The same proofs can be done for the $\bar{\lambda}\mu\tilde{\mu}$ -calculus and these proofs are, in fact, much simpler for this calculus since some difficult problems that appear in the $\lambda\mu$ -calculus do not appear in the $\bar{\lambda}\mu\tilde{\mu}$ -calculus: this is mainly due to the fact that, in the latter, there is a right-hand side and a left-hand side (the terms and the environments) whereas, in the $\lambda\mu$ -calculus, this distinction is impossible since a term on the right of an application can go on the left of an application after some reductions. The proof of the strong normalization of the $\mu\tilde{\mu}$ -reduction can be found in [22]. The proof is done (by using candidates of reducibility and a fix point operator) for a typed calculus but, in fact, since the type system is such that every term is typable, the result is valid for every term. A proof of the strong normalization of the $\bar{\lambda}\mu\tilde{\mu}$ -typed calculus (again using candidates of reducibility and a fix point operator) can also be found there. Due to the lack of space, we do not give our proofs of these results here but they will appear in [11].

The paper is organized as follows. In section 2 we give the syntax of the terms and the reduction rules. An arithmetical proof of strong normalization is given in section 3 for the $\mu\mu'$ -reduction of the un-typed calculus and, in section 4, for the $\beta\mu\mu'$ -reduction of the simply typed calculus. In section 5, we give an example showing that the proofs of strong normalization using candidates of reducibility *must* somehow be different from the usual ones and we show that, in the un-typed calculus, if M_1, \dots, M_n are strongly normalizing for the $\beta\mu\mu'$ -reduction, then so is $(x M_1 \dots M_n)$. We conclude with some future work.

2 The symmetric $\lambda\mu$ -calculus

2.1 The un-typed calculus

The set (denoted as \mathcal{T}) of $\lambda\mu$ -terms or simply terms is defined by the following grammar where x, y, \dots are λ -variables and α, β, \dots are μ -variables:

$$\mathcal{T} ::= x \mid \lambda x \mathcal{T} \mid (\mathcal{T} \mathcal{T}) \mid \mu \alpha \mathcal{T} \mid (\alpha \mathcal{T})$$

Note that we adopt here a more liberal syntax (also called de Groote's calculus) than in the original calculus since we do not ask that a $\mu\alpha$ is immediately followed by a (βM) (denoted $[\beta]M$ in Parigot's notation).

Definition 1. Let M be a term.

1. $cxy(M)$ is the number of symbols occurring in M .
2. We denote by $N \leq M$ (resp. $N < M$) the fact that N is a sub-term (resp. a strict sub-term) of M .
3. If \vec{P} is a sequence P_1, \dots, P_n of terms, $(M \vec{P})$ will denote $(M P_1 \dots P_n)$.

2.2 The typed calculus

The types are those of the simply typed $\lambda\mu$ -calculus i.e. are built from atomic formulas and the constant symbol \perp with the connector \rightarrow . As usual $\neg A$ is an abbreviation for $A \rightarrow \perp$.

The typing rules are given by figure 1 below where Γ is a context, i.e. a set of declarations of the form $x : A$ and $\alpha : \neg A$ where x is a λ (or intuitionistic) variable, α is a μ (or classical) variable and A is a formula.

$$\begin{array}{c}
 \frac{}{\Gamma, x : A \vdash x : A} ax \\
 \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x M : A \rightarrow B} \rightarrow_i \quad \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash (M N) : B} \rightarrow_e \\
 \frac{\Gamma, \alpha : \neg A \vdash M : \perp}{\Gamma \vdash \mu\alpha M : A} \perp_e \quad \frac{\Gamma, \alpha : \neg A \vdash M : A}{\Gamma, \alpha : \neg A \vdash (\alpha M) : \perp} \perp_i
 \end{array}$$

Figure 1.

Note that, here, we also have changed Parigot's notation but these typing rules are those of his classical natural deduction. Instead of writing

$$M : (A_1^{x_1}, \dots, A_n^{x_n} \vdash B, C_1^{\alpha_1}, \dots, C_m^{\alpha_m})$$

we have written

$$x_1 : A_1, \dots, x_n : A_n, \alpha_1 : \neg C_1, \dots, \alpha_m : \neg C_m \vdash M : B$$

Definition 2. Let A be a type. We denote by $lg(A)$ the number of arrows in A .

2.3 The reduction rules

The cut-elimination procedure (on the logical side) corresponds to the reduction rules (on the terms) given below. There are three kinds of cuts.

- A *logical cut* occurs when the introduction of the connective \rightarrow is immediately followed by its elimination. The corresponding reduction rule (denoted by β) is:

$$(\lambda x M N) \triangleright M[x := N]$$

- A *classical cut* occurs when \perp_e appears as the left premiss of a \rightarrow_e . The corresponding reduction rule (denoted by μ) is:

$$(\mu\alpha M N) \triangleright \mu\alpha M[\alpha =_r N]$$

where $M[\alpha =_r N]$ is obtained by replacing each sub-term of M of the form (αU) by $(\alpha (U N))$. This substitution is called a μ -substitution.

- A *symmetric classical cut* occurs when \perp_e appears as the right premiss of a \rightarrow_e . The corresponding reduction rule (denoted by μ') is:

$$(M \mu \alpha N) \triangleright \mu \alpha N[\alpha =_l M]$$

where $N[\alpha =_l M]$ is obtained by replacing each sub-term of N of the form (αU) by $(\alpha (M U))$. This substitution is called a μ' -substitution.

Remarks

1. It is shown in [18] that the $\beta\mu$ -reduction is confluent but neither $\mu\mu'$ nor $\beta\mu'$ is. For example $(\mu\alpha x \mu\beta y)$ reduces both to $\mu\alpha x$ and to $\mu\beta y$. Similarly $(\lambda z x \mu\beta y)$ reduces both to x and to $\mu\beta y$.
2. The reductions on terms correspond to the elimination of cuts on the proofs.
 - The β -reduction is the usual one.
 - The μ -reduction is as follows. If M corresponds to a proof of \perp assuming $\alpha : \neg(A \rightarrow B)$ and N corresponds to a proof of A , then $M[\alpha =_r N]$ corresponds to the proof M of \perp assuming $\alpha : \neg B$ but where, each time we used the hypothesis $\alpha : \neg(A \rightarrow B)$ with a proof U of $A \rightarrow B$ to get \perp , we replace this by the following proof of \perp . Use U and N to get a proof of B and then $\alpha : \neg B$ to get a proof of \perp .
 - Similarly, the μ' -reduction is as follows. If N corresponds to a proof of \perp assuming $\alpha : \neg A$ and M corresponds to a proof of $A \rightarrow B$, then $N[\alpha =_l M]$ corresponds to the proof N of \perp assuming $\alpha : \neg B$ but where, each time we used the hypothesis $\alpha : \neg A$ with a proof U of A to get \perp , we replace this by the following proof of \perp . Use U and M to get a proof of B and then $\alpha : \neg B$ to get a proof of \perp .
3. Unlike for a β -substitution where, in $M[x := N]$, the variable x has disappeared it is important to note that, in a μ or μ' -substitution, the variable α has not disappeared. Moreover its type has changed. If the type of N is A and, in M , the type of α is $\neg(A \rightarrow B)$ it becomes $\neg B$ in $M[\alpha =_r N]$. If the type of M is $A \rightarrow B$ and, in N , the type of α is $\neg A$ it becomes $\neg B$ in $N[\alpha =_l M]$.

In the next sections we will study various reductions : the $\mu\mu'$ -reduction in section 3 and the $\beta\mu\mu'$ -reduction in sections 4, 5. The following notions will correspond to these reductions.

Definition 3. Let \triangleright be a notion of reduction and M be a term.

1. The transitive (resp. reflexive and transitive) closure of \triangleright is denoted by \triangleright^+ (resp. \triangleright^*).
2. If M is in SN i.e. M has no infinite reduction, $\eta(M)$ will denote the length of the longest reduction starting from M and $\eta c(M)$ will denote $(\eta(M), cxy(M))$.
3. We denote by $N \prec M$ the fact that $N \leq M'$ for some M' such that $M \triangleright^* M'$ and either $M \triangleright^+ M'$ or $N < M'$. We denote by \preceq the reflexive closure of \prec .

Remarks

- It is easy to check that the relation \preceq is transitive and that $N \preceq M$ iff $N \leq M'$ for some M' such that $M \triangleright^* M'$.
- If $M \in SN$ and $N \prec M$, then $N \in SN$ and $\eta c(N) < \eta c(M)$. It follows that the relation \preceq is an order on the set SN .
- Many proofs will be done by induction on some k -uplet of integers. In this case the order we consider is the lexicographic order.

3 The $\mu\mu'$ -reduction is strongly normalizing

In this section we consider the $\mu\mu'$ -reduction, i.e. $M \triangleright M'$ means M' is obtained from M by one step of the $\mu\mu'$ -reduction. The main points of the proof of the strong normalization of $\mu\mu'$ are the following.

- We first show (cf. lemma 6) that a μ or μ' -substitution cannot create a μ .
- It is easy to show (see lemma 8) that if $M \in SN$ but $M[\sigma] \notin SN$ where σ is a μ or μ' -substitution, there are an α in the domain of σ and some $M' \prec M$ such that $M'[\sigma] \in SN$ and (say σ is a μ -substitution) $(M'[\sigma] \sigma(\alpha)) \notin SN$. This is sufficient to give a simple proof of the strongly normalization of the μ -reduction. But this is not enough to do a proof of the strongly normalization of the $\mu\mu'$ -reduction. We need a stronger (and more difficult) version of this: lemma 9 ensure that, if $M[\sigma] \in SN$ but $M[\sigma][\alpha =_r P] \notin SN$ then the real cause of non SN is, in some sense, $[\alpha =_r P]$.
- Having these results, we show, essentially by induction on $\eta c(M) + \eta c(N)$, that if $M, N \in SN$ then $(M N) \in SN$. The point is that there is, in fact, no deep interactions between M and N i.e. in a reduct of $(M N)$ we always know what is coming from M and what is coming from N .

Definition 4. – The set of simultaneous substitutions of the form $[\alpha_1 =_{s_1} P_1 \dots, \alpha_n =_{s_n} P_n]$ where $s_i \in \{l, r\}$ will be denoted by Σ .

- For $s \in \{l, r\}$, the set of simultaneous substitutions of the form $[\alpha_1 =_s P_1 \dots \alpha_n =_s P_n]$ will be denoted by Σ_s .
- If $\sigma = [\alpha_1 =_{s_1} P_1 \dots, \alpha_n =_{s_n} P_n]$, we denote by $dom(\sigma)$ (resp. $Im(\sigma)$) the set $\{\alpha_1, \dots, \alpha_n\}$ (resp. $\{P_1, \dots, P_n\}$).
- Let $\sigma \in \Sigma$. We say that $\sigma \in SN$ iff for every $N \in Im(\sigma)$, $N \in SN$.

Lemma 5. *If $(M N) \triangleright^* \mu\alpha P$, then either $M \triangleright^* \mu\alpha M_1$ and $M_1[\alpha =_r N] \triangleright^* P$ or $N \triangleright^* \mu\alpha N_1$ and $N_1[\alpha =_l M] \triangleright^* P$.*

Proof By induction on the length of the reduction $(M N) \triangleright^* \mu\alpha P$. □

Lemma 6. *Let M be a term and $\sigma \in \Sigma$. If $M[\sigma] \triangleright^* \mu\alpha P$, then $M \triangleright^* \mu\alpha Q$ for some Q such that $Q[\sigma] \triangleright^* P$.*

Proof By induction on M . M cannot be of the form $(\beta M')$ or $\lambda x M'$. If M begins with a μ , the result is trivial. Otherwise $M = (M_1 M_2)$ and, by lemma 5, either $M_1[\sigma] \triangleright^* \mu\alpha R$ and $R[\alpha =_r M_2[\sigma]] \triangleright^* P$ or $M_2[\sigma] \triangleright^* \mu\alpha R$ and $R[\alpha =_l M_1[\sigma]] \triangleright^* P$. Look at the first case (the other one is similar). By the induction hypothesis $M_1 \triangleright^* \mu\alpha Q$ for some Q such that $Q[\sigma] \triangleright^* R$ and thus $M \triangleright^* \mu\alpha Q[\alpha =_r M_2]$. Since $Q[\alpha =_r M_2][\sigma] = Q[\sigma][\alpha =_r M_2[\sigma]] \triangleright^* R[\alpha =_r M_2[\sigma]] \triangleright^* P$ we are done. □

Lemma 7. *Assume $M, N \in SN$ and $(M N) \notin SN$. Then either $M \triangleright^* \mu\alpha M_1$ and $M_1[\alpha =_r N] \notin SN$ or $N \triangleright^* \mu\beta N_1$ and $N_1[\beta =_l M] \notin SN$.*

Proof By induction on $\eta(M) + \eta(N)$. Since $(M N) \notin SN$, $(M N) \triangleright P$ for some P such that $P \notin SN$. If $P = (M' N)$ where $M \triangleright M'$ we conclude by the induction hypothesis since $\eta(M') + \eta(N) < \eta(M) + \eta(N)$. If $P = (M N')$ where $N \triangleright N'$ the proof is similar. If $M = \mu\alpha M_1$ and $P = \mu\alpha M_1[\alpha =_r N]$ or $N = \mu\beta N_1$ and $P = \mu\beta N_1[\beta =_l M]$ the result is trivial. □

Lemma 8. *Let M be term in SN and $\sigma \in \Sigma_s$ be in SN . Assume $M[\sigma] \notin SN$. Then, for some $(\alpha P) \preceq M$, $P[\sigma] \in SN$ and, if $s = l$ (resp. $s = r$), $(\sigma(\alpha) P[\sigma]) \notin SN$ (resp. $(P[\sigma] \sigma(\alpha)) \notin SN$).*

Proof We only prove the case $s = l$ (the other one is similar). Let $M_1 \preceq M$ be such that $M_1[\sigma] \notin SN$ and $\eta c(M_1)$ is minimal. By the minimality, M_1 cannot be $\lambda x M_2$ or $\mu\alpha M_2$. It cannot be either $(N_1 N_2)$ because otherwise, by the minimality, the $N_i[\sigma]$ would be in SN and thus, by lemma 7 and 6, we would have, for example,

$N_1 \triangleright^* \mu\alpha N'_1$ and $N'_1[\sigma][\alpha =_r N_2[\sigma]] = N'_1[\alpha =_r N_2][\sigma] \notin SN$ but this contradicts the minimality of M_1 since $\eta(N'_1[\alpha =_r N_2]) < \eta(M_1)$. Then $M_1 = (\alpha P)$ and the the minimality of M_1 implies that $P[\sigma] \in SN$. \square

Remark

From these results it is easy to prove, by induction on the term, the strong normalization of the μ -reduction. It is enough to show that, if $M, N \in SN$, then $(M N) \in SN$. Otherwise, we construct below a sequence (M_i) of terms and a sequence (σ_i) of substitutions such that, for every i , σ_i has the form $[\alpha_1 =_r N, \dots, \alpha_n =_r N]$, $M_i[\sigma_i] \notin SN$ and $M_{i+1} \prec M_i \prec M$. The sequence (M_i) contradicts the fact that $M \in SN$. Since $(M N) \notin SN$, by lemma 7, $M \triangleright^* \mu\alpha M_1$ and $M_1[\alpha =_r N] \notin SN$. Assume we have constructed M_i and σ_i . Since $M_i[\sigma_i] \notin SN$, by lemma 8, there is $M'_i \prec M_i$ such that $M'_i[\sigma_i] \in SN$ and $(M'_i[\sigma] N) \notin SN$. By lemmas 6 and 7, $M'_i \triangleright^* \mu\alpha M_{i+1}$ and $M_{i+1}[\sigma_i + \alpha =_r N] \notin SN$.

In the remark above, the fact that $(M N) \notin SN$ gives an infinite μ -reduction in M . This not the same for the the $\mu\mu'$ -reduction and, if we try to do the same, the substitutions we get are more complicated. In particular, it is not clear that we get an infinite sequence either of the form $\dots \prec M_2 \prec M_1 \prec M$ or of the form $\dots \prec N_2 \prec N_1 \prec N$. Lemma 9 below will give the answer since it will ensure that, at each step, we may assume that the cause of non SN is the last substitution.

Lemma 9. *Let M be a term and $\sigma \in \Sigma_s$. Assume δ is free in M but not free in $Im(\sigma)$. If $M[\sigma] \in SN$ but $M[\sigma][\delta =_s P] \notin SN$, there is $M' \prec M$ and σ' such that $M'[\sigma'] \in SN$ and, if $s = r$, $(M'[\sigma'] P) \notin SN$ and, if $s = l$, $(P M'[\sigma']) \notin SN$.*

Proof Assume $s = r$ (the other case is similar). Let $Im(\sigma) = \{N_1, \dots, N_k\}$. Assume M, δ, σ, P satisfy the hypothesis. Let $\mathcal{U} = \{U / U \preceq M\}$ and $\mathcal{V} = \{V / V \preceq N_i \text{ for some } i\}$. Define inductively the sets Σ_m and Σ_n of substitutions by the following rules:

$$\begin{aligned} \rho \in \Sigma_m &\text{ iff } \rho = \emptyset \text{ or } \rho = \rho' + [\beta =_r V[\tau]] \text{ for some } V \in \mathcal{V}, \tau \in \Sigma_n \text{ and } \rho' \in \Sigma_m \\ \tau \in \Sigma_n &\text{ iff } \tau = \emptyset \text{ or } \tau = \tau' + [\alpha =_l U[\rho]] \text{ for some } U \in \mathcal{U}, \rho \in \Sigma_m \text{ and } \tau' \in \Sigma_n \end{aligned}$$

Denote by C the conclusion of the lemma, i.e. there is $M' \prec M$ and σ' such that $M'[\sigma'] \in SN$, and $(M'[\sigma'] P) \notin SN$.

We prove something more general.

(1) Let $U \in \mathcal{U}$ and $\rho \in \Sigma_m$. Assume $U[\rho] \in SN$ and $U[\rho][\delta =_r P] \notin SN$. Then, C holds.

(2) Let $V \in \mathcal{V}$ and $\tau \in \Sigma_n$. Assume $V[\tau] \in SN$ and $V[\tau][\delta =_r P] \notin SN$. Then, C holds.

The conclusion C follows from (1) with M and σ . The properties (1) and (2) are proved by a simultaneous induction on $\eta c(U[\rho])$ (for the first case) and $\eta c(V[\tau])$ (for the second case).

Look first at (1)

- if $U = \lambda x U'$ or $U = \mu\alpha U'$: the result follows from the induction hypothesis with U' and ρ .

- if $U = (U_1 U_2)$: if $U_i[\rho][\delta =_r P] \notin SN$ for $i = 1$ or $i = 2$, the result follows from the induction hypothesis with U_i and ρ . Otherwise, by lemma 6 and 7, say $U_1 \triangleright^* \mu\alpha U'_1$ and, letting $U' = U'_1[\alpha =_r u_2]$, $U'[\rho][\delta =_r P] \notin SN$ and the result follows from the induction hypothesis with U' and ρ .

- if $U = (\delta U_1)$: if $U_1[\rho][\delta =_r P] \in SN$, then $M' = U_1$ and $\sigma' = \rho[\delta =_r P]$ satisfy the desired conclusion. Otherwise, the result follows from the induction hypothesis with U_1 and ρ .

- if $U = (\alpha U_1)$: if $\alpha \notin dom(\rho)$ or $U_1[\rho][\delta =_r P] \notin SN$, the result follows from the induction hypothesis with U_1 and ρ . Otherwise, let $\rho(\alpha) = V[\tau]$. If $V[\tau][\delta =_r$

$P] \notin SN$, the result follows from the induction hypothesis with V and τ (with (2)). Otherwise, by lemma 6 and 7, there are two cases to consider.

- $U_1 \triangleright^* \mu\alpha_1 U_2$ and $U_2[\rho'][\delta =_r P] \notin SN$ where $\rho' = \rho + [\alpha_1 =_r V[\tau]]$. The result follows from the induction hypothesis with U_2 and ρ' .

- $V \triangleright^* \mu\beta V_1$ and $V_1[\tau'][\delta =_r P] \notin SN$ where $\tau' = \tau + [\beta =_l U_1[\rho]]$. The result follows from the induction hypothesis with V_1 and τ' (with (2)).

The case (2) is proved in the same way. Note that, since δ is not free in the N_i , the case $b = (\delta V_1)$ does not appear. \square

Theorem 10. *Every term is in SN.*

Proof By induction on the term. It is enough to show that, if $M, N \in SN$, then $(M N) \in SN$. We prove something more general: let σ (resp. τ) be in Σ_r (resp. Σ_l) and assume $M[\sigma], N[\tau] \in SN$. Then $(M[\sigma] N[\tau]) \in SN$. Assume it is not the case and choose some elements such that $M[\sigma], N[\tau] \in SN$, $(M[\sigma] N[\tau]) \notin SN$ and $(\eta(M) + \eta(N), cxy(M) + cxy(N))$ is minimal. By lemma 7, either $M[\sigma] \triangleright^* \mu\delta M_1$ and $M_1[\delta =_r N[\tau]] \notin SN$ or $N[\tau] \triangleright^* \mu\beta N_1$ and $N_1[\beta =_l M[\sigma]] \notin SN$. Look at the first case (the other one is similar). By lemma 6, $M \triangleright^* \mu\delta M_2$ for some M_2 such that $M_2[\sigma] \triangleright^* M_1$. Thus, $M_2[\sigma][\delta =_r N[\tau]] \notin SN$. By lemma 9 with M_2, σ and $N[\tau]$, let $M' \prec M_2$ and σ' be such that $M'[\sigma'] \in SN$, $(M'[\sigma'] N[\tau]) \notin SN$. This contradicts the minimality of the chosen elements since $\eta c(M') < \eta c(M)$. \square

4 The simply typed symmetric $\lambda\mu$ -calculus is strongly normalizing

In this section, we consider the simply typed calculus with the $\beta\mu\mu'$ -reduction i.e. $M \triangleright M'$ means M' is obtained from M by one step of the $\beta\mu\mu'$ -reduction. To prove the strong normalization of the $\beta\mu\mu'$ -reduction, it is enough to show that, if $M, N \in SN$, then $M[x := N]$ also is in SN . This is done by induction on the type of N . The proof very much looks like the one for the $\mu\mu'$ -reduction and the induction on the type is used for the cases coming from a β -reduction. The two new difficulties are the following.

- A β -substitution may create a μ , i.e. the fact that $M[x := N] \triangleright^* \mu\alpha P$ does not imply that $M \triangleright^* \mu\alpha Q$. Moreover the μ may come from a complicated interaction between M and N and, in particular, the alternation between M and N can be lost. Let e.g. $M = (M_1 (x (\lambda y_1 \lambda y_2 \mu\alpha M_4) M_2 M_3))$ and $N = \lambda z (z N_1)$. Then $M[x := N] \triangleright^* (M_1 (\mu\alpha M_4' M_3)) \triangleright^* \mu\alpha M_4'[\alpha =_r M_3][\alpha =_l M_1]$. To deal with this situation, we need to consider some new kind of $\mu\mu'$ -substitutions (see definition 13). Lemma 16 gives the different ways in which a μ may appear. The difficult case in the proof (when a μ is created and the control between M and N is lost) will be solved by using a typing argument.

- The crucial lemma (lemma 18) is essentially the same as the one (lemma 9) for the $\mu\mu'$ -reduction but, in its proof, some cases cannot be proved “by themselves” and we need an argument using the types. For this reason, its proof is done using the additional fact that we already know that, if $M, N \in SN$ and the type of N is small, then $M[x := N]$ also is in SN . Since the proof of lemma 19 is done by induction on the type, when we will use lemma 18, the additional hypothesis will be available.

Lemma 11. 1. *If $(M N) \triangleright^* \lambda x P$, then $M \triangleright^* \lambda y M_1$ and $M_1[y := N] \triangleright^* \lambda x P$.*
2. *If $(M N) \triangleright^* \mu\alpha P$, then either $(M \triangleright^* \lambda y M_1$ and $M_1[y := N] \triangleright^* \mu\alpha P)$ or $(M \triangleright^* \mu\alpha M_1$ and $M_1[\alpha =_r N] \triangleright^* P)$ or $(N \triangleright^* \mu\alpha N_1$ and $N_1[\alpha =_l M] \triangleright^* P)$.*

Proof (1) is trivial. (2) is as in lemma 5. \square

Lemma 12. *Let $M \in SN$ and $\sigma = [x_1 := N_1, \dots, x_k := N_k]$. Assume $M[\sigma] \triangleright^* \lambda y P$. Then, either $M \triangleright^* \lambda y P_1$ and $P_1[\sigma] \triangleright^* P$ or $M \triangleright^* (x_i \overrightarrow{Q})$ and $(N_i \overrightarrow{Q}[\sigma]) \triangleright^* \lambda y P$.*

Proof By induction on $\eta c(M)$. The only non immediate case is $M = (R S)$. By lemma 11, there is a term R_1 such that $R[\sigma] \triangleright^* \lambda z R_1$ and $R_1[z := S[\sigma]] \triangleright^* \lambda y P$. By the induction hypothesis (since $\eta c(R) < \eta c(M)$), we have two cases to consider.

(1) $R \triangleright^* \lambda z R_2$ and $R_2[\sigma] \triangleright^* R_1$, then $R_2[z := S][\sigma] \triangleright^* \lambda y P$. By the induction hypothesis (since $\eta(R_2[z := S]) < \eta(M)$),

- either $R_2[z := S] \triangleright^* \lambda y P_1$ and $P_1[\sigma] \triangleright^* P$; but then $M \triangleright^* \lambda y P_1$ and we are done.

- or $R_2[z := S] \triangleright^* (x_i \overrightarrow{Q})$ and $(N_i \overrightarrow{Q}[\sigma]) \triangleright^* \lambda y P$, then $M \triangleright^* (x_i \overrightarrow{Q})$ and again we are done.

(2) $R \triangleright^* (x_i \overrightarrow{Q})$ and $(N_i \overrightarrow{Q}[\sigma]) \triangleright^* \lambda z R_1$. Then $M \triangleright^* (x_i \overrightarrow{Q} S)$ and the result is trivial. \square

Definition 13. – An address is a finite list of symbols in $\{l, r\}$. The empty list is denoted by $[]$ and, if a is an address and $s \in \{l, r\}$, $[s :: a]$ denotes the list obtained by putting s at the beginning of a .

– Let a be an address and M be a term. The sub-term of M at the address a (denoted as M_a) is defined recursively as follows : if $M = (P Q)$ and $a = [r :: b]$ (resp. $a = [l :: b]$) then $M_a = Q_b$ (resp. P_b) and undefined otherwise.

– Let M be a term and a be an address such that M_a is defined. Then $M\langle a = N \rangle$ is the term M where the sub-term M_a has been replaced by N .

– Let M, N be some terms and a be an address such that M_a is defined. Then $N[\alpha =_a M]$ is the term N in which each sub-term of the form (αU) is replaced by $(\alpha M\langle a = U \rangle)$.

Remarks and examples

- Let $N = \lambda x(\alpha \lambda y(x \mu \beta(\alpha y)))$, $M = (M_1 (M_2 M_3))$ and $a = [r :: l]$. Then $N[\alpha =_a M] = \lambda x(\alpha (M_1 (\lambda y(x \mu \beta(\alpha (M_1 (y M_3)))) M_3)))$.

- Let $M = (P ((R (x T)) Q))$ and $a = [r :: l :: r :: l]$. Then $N[\alpha =_a M] = N[\alpha =_r T][\alpha =_l R][\alpha =_r Q][\alpha =_r P]$.

- Note that the sub-terms of a term having an address in the sense given above are those for which the path to the root consists only on applications (taking either the left or right son).

- Note that $[\alpha =_{[l]} M]$ is not the same as $[\alpha =_l M]$ but $[\alpha =_l M]$ is the same as $[\alpha =_{[r]} (M N)]$ where N does not matter. More generally, the term $N[\alpha =_a M]$ does not depend of M_a .

- Note that $M\langle a = N \rangle$ can be written as $M'[x_a := N]$ where M' is the term M in which M_a has been replaced by the fresh variable x_a and thus (this will be used in the proof of lemma 19) if M_a is a variable x , $(\alpha U)[\alpha =_a M] = (\alpha M_1[y := U[\alpha =_a M]])$ where M_1 is the term M in which the particular occurrence of x at the address a has been replaced by the fresh name y and the other occurrences of x remain unchanged.

Lemma 14. *Assume $M, N \in SN$ and $(M N) \notin SN$. Then, either $(M \triangleright^* \lambda y P$ and $P[y := N] \notin SN)$ or $(M \triangleright^* \mu \alpha P$ and $P[\alpha =_r N] \notin SN)$ or $(N \triangleright^* \mu \alpha P$ and $P[\alpha =_l M] \notin SN)$.*

Proof By induction on $\eta(M) + \eta(N)$. \square

In the rest of this section, we consider the typed calculus. To simplify the notations, we do not write explicitly the type information but, when needed, we denote by $\text{type}(M)$ the type of the term M .

Lemma 15. *If $\Gamma \vdash M : A$ and $M \triangleright^* N$ then $\Gamma \vdash N : A$.*

Proof Straight forward. \square

Lemma 16. Let n be an integer, $M \in SN$, $\sigma = [x_1 := N_1, \dots, x_k := N_k]$ where $lg(type(N_i)) = n$ for each i . Assume $M[\sigma] \triangleright^* \mu\alpha P$. Then,

1. either $M \triangleright^* \mu\alpha P_1$ and $P_1[\sigma] \triangleright^* P$
2. or $M \triangleright^* Q$ and, for some i , $N_i \triangleright^* \mu\alpha N'_i$ and $N'_i[\alpha =_a Q[\sigma]] \triangleright^* P$ for some address a in Q such that $Q_a = x_i$.
3. or $M \triangleright^* Q$, $Q_a[\sigma] \triangleright^* \mu\alpha N'$ and $N'[\alpha =_a Q[\sigma]] \triangleright^* P$ for some address a in Q such that $lg(type(Q_a)) < n$.

Proof By induction on $\eta c(M)$. The only non immediate case is $M = (R S)$. Since $M[\sigma] \triangleright^* \mu\alpha P$, the application $(R[\sigma] S[\sigma])$ must be reduced. Thus there are three cases to consider.

- It is reduced by a μ' -reduction, i.e. there is a term S_1 such that $S[\sigma] \triangleright^* \mu\alpha S_1$ and $S_1[\alpha =_l R[\sigma]] \triangleright^* P$. By the induction hypothesis:
 - either $S \triangleright^* \mu\alpha Q$ and $Q[\sigma] \triangleright^* S_1$, then $M \triangleright^* \mu\alpha Q[\alpha =_l R]$ and $Q[\alpha =_l R][\sigma] \triangleright^* P$.
 - or $S \triangleright^* Q$ and, for some i , $N_i \triangleright^* \mu\alpha N'_i$, $Q_a = x_i$ for some address a in Q and $N'_i[\alpha =_a Q[\sigma]] \triangleright^* S_1$. Then $M \triangleright^* (R Q) = Q'$ and letting $b = [r :: a]$ we have $N'_i[\alpha =_b Q'[\sigma]] \triangleright^* P$.
 - or $S \triangleright^* Q$, $Q_a[\sigma] \triangleright^* \mu\alpha N'$ for some address a in Q such that $lg(type(Q_a)) < n$ and $N'[\alpha =_a Q[\sigma]] \triangleright^* S_1$. Then $M \triangleright^* (R Q) = Q'$ and letting $b = [r :: a]$ we have $N'[\alpha =_b Q'[\sigma]] \triangleright^* P$ and $lg(type(Q'_b)) < n$.
- It is reduced by a μ -reduction. This case is similar to the previous one.
- It is reduced by a β -reduction, i.e. there is a term U such that $R[\sigma] \triangleright^* \lambda y U$ and $U[y := S[\sigma]] \triangleright^* \mu\alpha P$. By lemma 12, there are two cases to consider.
 - either $R \triangleright^* \lambda y R_1$ and $R_1[\sigma][y := S[\sigma]] = R_1[y := S][\sigma] \triangleright^* \mu\alpha P$. The result follows from the induction hypothesis sine $\eta(R_1[y := S]) < \eta(M)$.
 - or $R \triangleright^* (x_i \overrightarrow{R_1})$. Then $Q = (x_i \overrightarrow{R_1} S)$ and $a = []$ satisfy the desired conclusion since then $lg(type(M)) < n$. \square

Definition 17. Let A be a type. We denote by Σ_A the set of substitutions of the form $[\alpha_1 =_{a_1} M_1, \dots, \alpha_n =_{a_n} M_n]$ where the type of the α_i is $\neg A$.

Remark

Since in such substitutions the type of the variables changes, when we consider the term $N[\sigma]$ where $\sigma \in \Sigma_A$, we mean that the type of the α_i is A in N i.e. before the substitution. Also note that considering $N[\alpha =_a M]$ implies that the type of M_a is A .

Lemma 18. Let n be an integer and A be a type such that $lg(A) = n$. Let N, P be terms and $\tau \in \Sigma_A$. Assume that,

- for every $M, N \in SN$ such that $lg(type(N)) < n$, $M[x := N] \in SN$.
- $N[\tau] \in SN$ but $N[\tau][\delta =_a P] \notin SN$.
- δ is free and has type $\neg A$ in N but δ is not free in $Im(\tau)$.

Then, there is $N' \prec N$ and $\tau' \in \Sigma_A$ such that $N'[\tau'] \in SN$ and $P\langle a = N'[\tau'] \rangle \notin SN$.

Proof Essentially as in lemma 9. Denote by (H) the first assumption i.e. for every $M, N \in SN$ such that $lg(type(N)) < n$, $M[x := N] \in SN$.

Let $\tau = [\alpha_1 =_{a_1} M_1, \dots, \alpha_n =_{a_n} M_n]$, $\mathcal{U} = \{U / U \preceq N\}$ and $\mathcal{V} = \{V / V \preceq M_i$ for some $i\}$. Define inductively the sets Σ_m and Σ_n of substitutions by the following rules:

$\rho \in \Sigma_n$ iff $\rho = \emptyset$ or $\rho = \rho' + [\alpha =_a V[\sigma]]$ for some $V \in \mathcal{V}$, $\sigma \in \Sigma_m$, $\rho' \in \Sigma_n$ and α has type $\neg A$.

$\sigma \in \Sigma_m$ iff $\sigma = \emptyset$ or $\sigma = \sigma' + [x := U[\rho]]$ for some $U \in \mathcal{U}$, $\rho \in \Sigma_n$, $\sigma' \in \Sigma_m$ and x has type A .

Denote by C the conclusion of the lemma. We prove something more general.

(1) Let $U \in \mathcal{U}$ and $\rho \in \Sigma_n$. Assume $U[\rho] \in SN$ and $U[\rho][\delta =_a P] \notin SN$. Then, C holds.

(2) Let $V \in \mathcal{V}$ and $\sigma \in \Sigma_m$. Assume $V[\sigma] \in SN$ and $V[\sigma][\delta =_a P] \notin SN$. Then, C holds.

The conclusion C follows from (1) with N and τ . The properties (1) and (2) are proved by a simultaneous induction on $\eta c(U[\rho])$ (for the first case) and $\eta c(V[\tau])$ (for the second case).

The proof is as in lemma 9. The new case to consider is, for $V[\sigma]$, when $V = (V_1 V_2)$ and $V_i[\sigma][\delta =_a P] \in SN$.

- Assume first the interaction between V_1 and V_2 is a β -reduction. If $V_1 \triangleright^* \lambda x V_1'$, the result follows from the induction hypothesis with $V_1'[x := V_2][\sigma]$. Otherwise, by lemma 12, $V_1 \triangleright^* (x \vec{W})$. Let $\sigma(x) = U[\rho]$. Then $(U[\rho] \vec{W}[\sigma]) \triangleright^* \lambda y Q$ and $Q[y := V_2[\sigma]][\delta =_a P] \notin SN$. But, since the type of x is A , the type of y is less than A and since $Q[\delta =_a P]$ and $V_2[\sigma][\delta =_a P]$ are in SN this contradicts (H).

- Assume next the interaction between V_1 and V_2 is a μ or μ' -reduction. We consider only the case μ (the other one is similar). If $V_1 \triangleright^* \mu \alpha V_1'$, the result follows from the induction hypothesis with $V_1'[\alpha =_r V_2][\sigma]$. Otherwise, by lemma 16, there are two cases to consider.

- $V_1 \triangleright^* Q$, $Q_c = x$ for some address c in Q and $x \in \text{dom}(\sigma)$, $\sigma(x) = U[\rho]$, $U \triangleright^* \mu \alpha U_1$ and $U_1[\rho][\alpha =_c Q[\sigma]][\alpha =_r V_2[\sigma]][\delta =_a P] \notin SN$. Let $V' = (Q V_2)$ and $b = l :: c$. The result follows then from the induction hypothesis with $U_1[\rho']$ where $\rho' = \rho + [\alpha =_b V'[\sigma]]$.

- $V_1 \triangleright^* Q$, $Q_c[\sigma][\delta =_a P] \triangleright^* \mu \alpha R$ for some address c in Q such that $lg(\text{type}(Q_c)) < n$, $R[\alpha =_c Q[\sigma][\delta =_a P]][\alpha =_r V_2[\sigma][\delta =_a P]] \notin SN$. Let $V' = (Q' V_2)$ where Q' is the same as Q but Q_c has been replaced by a fresh variable y and $b = l :: c$. Then $R[\alpha =_b V'[\sigma][\delta =_a P]] \notin SN$. Let R' be such that $R' \prec R$, $R'[\alpha =_b V'[\sigma][\delta =_a P]] \notin SN$ and $\eta c(R')$ is minimal. It is easy to check that $R' = (\alpha R'')$, $R''[\alpha =_b V'[\sigma][\delta =_a P]] \in SN$ and $V'[\sigma][\delta =_a P] \notin SN$ where $\sigma' = \sigma + y := R''[\alpha =_b V'[\sigma]]$. If $V'[\sigma][\delta =_a P] \notin SN$, we get the result by the induction hypothesis since $\eta c(V'[\sigma]) < \eta c(V[\sigma])$. Otherwise this contradicts the assumption (H) since $V'[\sigma][\delta =_a P], R''[\alpha =_b V'[\sigma][\delta =_a P]] \in SN$, $V'[\sigma][\delta =_a P][y := R''[\alpha =_b V'[\sigma][\delta =_a P]]] \notin SN$ and the type of y is less than n .

□

Lemma 19. *If $M, N \in SN$, then $M[x := N] \in SN$.*

Proof We prove something a bit more general: let A be a type, M, N_1, \dots, N_k be terms and τ_1, \dots, τ_k be substitutions in Σ_A . Assume that, for each i , N_i has type A and $N_i[\tau_i] \in SN$. Then $M[x_1 := N_1[\tau_1], \dots, x_k := N_k[\tau_k]] \in SN$. This is proved by induction on $(lg(A), \eta(M), \text{cxt}_y(M), \Sigma \eta(N_i), \Sigma \text{cxt}_y(N_i))$ where, in $\Sigma \eta(N_i)$ and $\Sigma \text{cxt}_y(N_i)$, we count each occurrence of the substituted variable. For example if $k = 1$ and x_1 has n occurrences, $\Sigma \eta(N_i) = n \cdot \eta(N_1)$.

If M is $\lambda y M_1$ or (αM_1) or $\mu \alpha M_1$ or a variable, the result is trivial. Assume then that $M = (M_1 M_2)$. Let $\sigma = [x_1 := N_1[\tau_1], \dots, x_k := N_k[\tau_k]]$. By the induction hypothesis, $M_1[\sigma], M_2[\sigma] \in SN$. By lemma 14 there are 3 cases to consider.

- $M_1[\sigma] \triangleright^* \lambda y P$ and $P[y := M_2[\sigma]] \notin SN$. By lemma 12, there are two cases to consider.
 - $M_1 \triangleright^* \lambda y Q$ and $Q[\sigma] \triangleright^* P$. Then $Q[y := M_2][\sigma] = Q[\sigma][y := M_2[\sigma]] \triangleright^* P[y := M_2[\sigma]]$ and, since $\eta(Q[y := M_2]) < \eta(M)$, this contradicts the induction hypothesis.

- $M_1 \triangleright^* (x_i \overrightarrow{Q})$ and $(N_i \overrightarrow{Q[\sigma]}) \triangleright^* \lambda y P$. Then, since the type of N_i is A , $lg(type(y)) < lg(A)$. But $P, M_2[\sigma] \in SN$ and $P[y := M_2[\sigma]] \notin SN$. This contradicts the induction hypothesis.
- $M_1[\sigma] \triangleright^* \mu \alpha P$ and $P[\alpha =_r M_2[\sigma]] \notin SN$. By lemma 16, there are three cases to consider.
 - $M_1 \triangleright^* \mu \alpha Q$ and $Q[\sigma] \triangleright^* P$. Then, $Q[\alpha =_r M_2[\sigma]] = Q[\sigma][\alpha =_r M_2[\sigma]] \triangleright^* P[\alpha =_r M_2[\sigma]]$ and, since $\eta(Q[\alpha =_r M_2]) < \eta(M)$, this contradicts the induction hypothesis.
 - $M_1 \triangleright^* Q$, $N_i[\tau_i] \triangleright^* \mu \alpha L'$ and $Q_a = x_i$ for some address a in Q such that $L'[\alpha =_a Q[\sigma]] \triangleright^* P$ and thus $L'[\alpha =_b M'[\sigma]] \notin SN$ where $b = (l :: a)$ and $M' = (Q M_2)$.
By lemma 6, $N_i \triangleright^* \mu \alpha L$ and $L[\tau_i] \triangleright^* L'$. Thus, $L[\tau_i][\alpha =_b M'[\sigma]] \notin SN$. By lemma 18, there is $L_1 \prec L$ and τ' such that $L_1[\tau'] \in SN$ and $M'[\sigma]\langle b = L_1[\tau'] \rangle \notin SN$. Let M'' be M' where the variable x_i at the address b has been replaced by the fresh variable y and let $\sigma_1 = \sigma + y := L_1[\tau']$. Then $M''[\sigma_1] = M'[\sigma]\langle b = L_1[\tau'] \rangle \notin SN$.
If $M_1 \triangleright^+ Q$ we get a contradiction from the induction hypothesis since $\eta(M'') < \eta(M)$. Otherwise, M'' is the same as M up to the change of name of a variable and σ_1 differs from σ only at the address b . At this address, x_i was substituted in σ by $N_i[\tau_i]$ and in σ_1 by $L_1[\tau']$ but $\eta c(L_1) < \eta c(N_i)$ and thus we get a contradiction from the induction hypothesis.
 - $M \triangleright^* Q$, $Q_a[\sigma] \triangleright^* \mu \alpha L$ for some address a in Q such that $lg(type(Q_a)) < lg(A)$ and $L[\alpha =_a Q[\sigma]] \triangleright^* P$. Then, $L[\alpha =_b M'[\sigma]] \notin SN$ where $b = [l :: a]$ and $M' = (Q M_2)$.
By lemma 18, there is an L' and τ' such that $L'[\tau'] \in SN$ and $M'[\sigma]\langle b = L'[\tau'] \rangle \notin SN$. Let M'' be M' where the variable x_i at the address b has been replaced by the fresh variable y . Then $M''[\sigma][y := L'[\tau']] = M'[\sigma]\langle b = L'[\tau'] \rangle \notin SN$.
But $\eta(M'') \leq \eta(M)$ and $cxy(M'') < cxy(M)$ since, because of its type, Q_a cannot be a variable and thus, by the induction hypothesis, $M''[\sigma] \in SN$. Since $M''[\sigma][y := L'[\tau']] \notin SN$ and $lg(type(L')) < lg(A)$, this contradicts the induction hypothesis.
- $M_2[\sigma] \triangleright^* \mu \alpha P$ and $P[\alpha =_l M_1[\sigma]] \notin SN$. This case is similar to the previous one. \square

Theorem 20. *Every typed term is in SN.*

Proof By induction on the term. It is enough to show that if $M, N \in SN$, then $(M N) \in SN$. Since $(M N) = (x y)[x := M][y := N]$ where x, y are fresh variables, the result follows by applying theorem 19 twice and the induction hypothesis. \square

5 Why the usual candidates do not work ?

In [21], the proof of the strong normalization of the $\lambda\mu$ -calculus is done by using the *usual* (i.e. defined without a fix-point operation) candidates of reducibility. This proof could be easily extended to the symmetric $\lambda\mu$ -calculus if we knew the following properties for the un-typed calculus:

1. If N and $(M[x := N] \overrightarrow{P})$ are in SN , then so is $(\lambda x M N \overrightarrow{P})$.
2. If N and $(M[\alpha =_r N] \overrightarrow{P})$ are in SN , then so is $(\mu \alpha M N \overrightarrow{P})$.
3. If \overrightarrow{P} are in SN , then so is $(x \overrightarrow{P})$.

These properties are easy to show for the $\beta\mu$ -reduction but they were not known for the $\beta\mu\mu'$ -reduction.

The properties (1) and (2) are false. Here is a counter-example. Let $M_0 = \lambda x(x P \underline{0})$ and $M_1 = \lambda x(x P \underline{1})$ where $\underline{0} = \lambda x\lambda y y$, $\underline{1} = \lambda x\lambda y x$, $\Delta = \lambda x(x x)$ and $P = \lambda x\lambda y\lambda z (y (z \underline{1} \underline{0}) (z \underline{0} \underline{1}) \lambda d \underline{1} \Delta \Delta)$. Let $M = \lambda f(f (x M_1) (x M_0))$, $M' = \lambda f(f (\beta \lambda x(x M_1)) (\beta \lambda x(x M_0)))$ and $N = (\alpha \lambda z(\alpha z))$. Then,

- $M[x := \mu\alpha N] \in SN$ but $(\lambda x M \mu\alpha N) \notin SN$.
- $M'[\beta =_r \mu\alpha N] \in SN$ but $(\mu\beta M' \mu\alpha N) \notin SN$.

This comes from the fact that $(M_0 M_0)$ and $(M_1 M_1)$ are in SN but $(M_1 M_0)$ and $(M_0 M_1)$ are not in SN . More details can be found in [10].

The third property is true and its proof is essentially the same as the one of the strong normalization of $\mu\mu'$. This comes from the fact that, since $(x M_1 \dots M_n)$ never reduces to a λ , there is no “dangerous” β -reduction. In particular, the β -reductions we have to consider in the proofs of the crucial lemmas, are uniquely those that appear in the reductions $M \preceq M'$. We give this proof below.

Lemma 21. *The term $(x M_1 \dots M_n)$ never reduces to a term of the form $\lambda y M$.*

Proof By induction on n . Use lemma 11. \square

Definition 22. - Let M_1, \dots, M_n be terms and $1 \leq i \leq n$. Then, the term M in which every sub-term of the form (αU) is replaced by $(\alpha (x M_1 \dots M_{i-1} U M_{i+1} \dots M_n))$ will be denoted by $M[\alpha =_i (M_1 \dots M_n)]$.

- We will denote by Σ_x the set of simultaneous substitutions of the form $[\alpha_1 =_{i_1} (M_1^1 \dots M_n^1), \dots, \alpha_k =_{i_k} (M_1^k \dots M_n^k)]$.

Remark

These substitutions are special cases of the one defined in section 4 (see definition 13). For example $M[\alpha =_2 (M_1 M_2 M_3)] = M[\alpha =_l (x M_1)][\alpha =_r M_3] = M[\alpha =_a (x M_1 M_2 M_3)]$ where $a = [l :: r]$.

Lemma 23. *Assume $(x M_1 \dots M_n) \triangleright^* \mu\alpha M$. Then, there is an i such that $M_i \triangleright^* \mu\alpha P$ and $P[\alpha =_i (M_1 \dots M_n)] \triangleright^* M$.*

Proof By induction on n .

- $n = 1$. By lemma 11, $M_1 \triangleright^* \mu\alpha P$ and $P[\alpha =_l x] = P[\alpha =_1 (M_1)] \triangleright^* M$.

- $n \geq 2$. Assume $(x M_1 \dots M_{n-1} M_n) \triangleright^* \mu\alpha M$. By lemmas 11 and 21,

- either $(x M_1 \dots M_{n-1}) \triangleright^* \mu\alpha N$ and $N[\alpha =_r M_n] \triangleright^* M$. By the induction hypothesis, there is an i such that $M_i \triangleright^* \mu\alpha P$ and $P[\alpha =_i (M_1 \dots M_{n-1})] \triangleright^* N$. Then $P[\alpha =_i (M_1 \dots M_{n-1} M_n)] = P[\alpha =_i (M_1 \dots M_{n-1})][\alpha =_r M_n] \triangleright^* N[\alpha =_r M_n] \triangleright^* M$.

- or $M_n \triangleright^* \mu\alpha N$ and $N[\alpha =_l (x M_1 \dots M_{n-1})] \triangleright^* M$. Then $N[\alpha =_l (x M_1 \dots M_{n-1})] = N[\alpha =_n (M_1 \dots M_{n-1} M_n)] \triangleright^* M$. \square

Lemma 24. *Assume $M_1, \dots, M_n \in SN$ and $(x M_1 \dots M_n) \notin SN$. Then, there is an $1 \leq i \leq n$ such that $M_i \triangleright^* \mu\alpha U$ and $U[\alpha =_i (M_1 \dots M_n)] \notin SN$.*

Proof Let k be the least such that $(x M_1 \dots M_{k-1}) \in SN$ and $(x M_1 \dots M_k) \notin SN$. By lemmas 14 and 21,

- either $M_k \triangleright^* \mu\alpha U$ and $U[\alpha =_l (x M_1 \dots M_{k-1})] \notin SN$. Then, $i = k$ satisfies the desired property since $U[\alpha =_k (M_1 \dots M_n)] = U[\alpha =_l (x M_1 \dots M_{k-1})][\alpha =_r M_{k+1}] \dots [\alpha =_r M_n]$.

- or $(x M_1 \dots M_{k-1}) \triangleright^* \mu\alpha P$ and $P[\alpha =_r M_k] \notin SN$. By lemma 23, let $i \leq k-1$ be such that $M_i \triangleright^* \mu\alpha U$ and $U[\alpha =_i (M_1 \dots M_{k-1})] \triangleright^* P$. Then $U[\alpha =_i (M_1 \dots M_n)] \notin SN$ since $U[\alpha =_i (M_1 \dots M_n)] = U[\alpha =_i (M_1 \dots M_{k-1})][\alpha =_r M_k][\alpha =_r M_{k+1}] \dots [\alpha =_r M_n]$ reduces to $P[\alpha =_r M_k][\alpha =_r M_{k+1}] \dots [\alpha =_r M_n]$. \square

Lemma 25. *Let M be a term and $\sigma \in \Sigma_x$. If $M[\sigma] \triangleright^* \mu\alpha P$ (resp. $M[\sigma] \triangleright^* \lambda x P$), then $M \triangleright^* \mu\alpha Q$ (resp. $M \triangleright^* \lambda x Q$) for some Q such that $Q[\sigma] \triangleright^* P$.*

Proof As in lemma 6. □

Lemma 26. *Let M be a term and $\sigma \in \Sigma_x$. Assume δ is free in M but not free in $Im(\sigma)$. If $M[\sigma] \in SN$ but $M[\sigma][\delta =_i (P_1 \dots P_n)] \notin SN$, there is $M' \prec M$ and σ' such that $M'[\sigma'] \in SN$ and $(x P_1 \dots P_{i-1} M'[\sigma'] P_{i+1} \dots P_n) \notin SN$.*

Proof As in lemma 9. □

Theorem 27. *Assume M_1, \dots, M_n are in SN . Then $(x M_1 \dots M_n) \in SN$.*

Proof We prove a more general result: Let M_1, \dots, M_n be terms and $\sigma_1, \dots, \sigma_n$ be in Σ_x . If $M_1[\sigma_1], \dots, M_n[\sigma_n] \in SN$, then $(x M_1[\sigma_1] \dots M_n[\sigma_n]) \in SN$. The proof is done exactly as in theorem 10 using lemmas 24, 25 and 26. □

6 Future work

- Parigot has introduced other simplification rules in the $\lambda\mu$ -calculus. They are as follows : $(\alpha \mu\beta M) \rightarrow_\rho M[\beta := \alpha]$ and, if α is not free in M , $\mu\alpha(\alpha M) \rightarrow_\theta M$. It would be interesting to extend our proofs to these reductions. The rule θ causes no problem since it is strongly normalizing and it is easy to see that this rule can be postponed (i.e. if $M \rightarrow_{\beta\mu\mu'\rho\theta}^* M_1$ then $M \rightarrow_{\beta\mu\mu'\rho}^* M_2 \rightarrow_\theta^* M_1$ for some M_2). However it is not the same for the rule ρ which cannot be postponed. Moreover a basic property (if $M[\alpha =_s N] \triangleright^* \mu\beta P$, then $M \triangleright^* \mu\beta Q$ for some Q such that $Q[\alpha =_s N] \triangleright^* P$) used in the proofs is no more true if the ρ -rule is used. It seems that, in this case, the μ can only come either from M or from N i.e. without deep interaction between M and N and thus that our proofs can be extended to this case but, due to the lack of time, we have not been able to check the details.
- We believe that our technique, will allow to give explicit bounds for the length of the reductions of a typed term. This is a goal we will try to manage.

References

1. F. Barbanera and S. Berardi. *A symmetric lambda-calculus for classical program extraction*. In M. Hagiya and J.C. Mitchell, editors, Proceedings of theoretical aspects of computer software, TACS'94. LNCS (789), pp. 495-515. Springer Verlag, 1994.
2. R. Constable and C. Murthy. *Finding computational content in classical proofs*. In G. Huet and G. Plotkin, editors, Logical Frameworks, pp. 341-362, Cambridge University Press, 1991.
3. P.L. Curien and H. Herbelin. *The Duality of Computation*. Proc. International Conference on Functional Programming, September 2000, Montral, IEEE, 2000.
4. J.-Y. Girard. *A new constructive logic: classical logic*. MSCS (1), pp. 255-296, 1991.
5. P. de Groote. *A CPS-translation of the lambda-mu-calculus*. In S. Tison, editor, 19th International Colloquium on Trees in Algebra and Programming, CAAP'94, volume 787 of Lecture Notes in Computer Science, pp 85-99. Springer, 1994.
6. P. de Groote. *A simple calculus of exception handling*. In M. Dezani and G. Plotkin, editors, Second International Conference on Typed Lambda Calculi and Applications, TLCA'95, volume 902 of Lecture Notes in Computer Science, pp. 201-215. Springer, 1995.
7. R. David. *Normalization without reducibility*. Annals of Pure and Applied Logic (107), pp. 121-130, 2001.
8. R. David and K. Nour. *A short proof of the strong normalization of the simply typed $\lambda\mu$ -calculus*. Schedae Informaticae n12, pp. 27-34, 2003.

9. R. David and K. Nour. *A short proof of the strong normalization of classical natural deduction with disjunction*. The Journal of Symbolic Logic n 68.4, pp. 1277 - 1288, 2003.
10. R. David and K. Nour. *Why the usual candidates of reducibility do not work for the symmetric $\lambda\mu$ -calculus*. To appear in ENTCS (2005).
11. R. David and K. Nour. *Arithmetical proofs of the strong normalization of the $\bar{\lambda}\bar{\mu}$ -calculus*. Manuscript 2004.
12. J.-L. Krivine. *Classical logic, storage operators and 2nd order lambda-calculus*. Annals of Pure and Applied Logic (68), pp. 53-78, 1994.
13. C.R. Murthy. *An evaluation semantics for classical proofs*. In Proceedings of the sixth annual IEEE symposium on logic in computer science, pp. 96-107, 1991.
14. K. Nour. *La valeur d'un entier classique en $\lambda\mu$ -calcul*. Archive for Mathematical Logic (36), pp. 461-471, 1997.
15. K. Nour. *A non-deterministic classical logic (the $\lambda\mu^{++}$ -calculus)*. Mathematical Logic Quarterly (48), pp. 357 - 366, 2002.
16. K. Nour and K. Saber. *A semantical proof of the strong normalization theorem of full propositionnal classical natural deduction*. Manuscript (2004).
17. M. Parigot. *Free Deduction: An Analysis of "Computations" in Classical Logic*. Proceedings. Lecture Notes in Computer Science, Vol. 592, Springer, pp. 361-380, 1992.
18. M. Parigot. *$\lambda\mu$ -calculus: An algorithm interpretation of classical natural deduction*. Lecture Notes in Artificial Intelligence (624), pp. 190-201. Springer Verlag, 1992.
19. M. Parigot. *Strong normalization for second order classical natural deduction*. In Proceedings, Eighth Annual IEEE Symposium on Logic in Computer Science, pp. 39-46, Montreal, Canada, 19-23 June 1993. IEEE Computer Society Press.
20. M. Parigot. *Classical proofs as programs*. In G. Gottlob, A. Leitsch, and D. Mundici, eds., Proc. of 3rd Kurt Godel Colloquium, KGC'93, vol. 713 of Lecture Notes in Computer Science, pp. 263-276. Springer-Verlag, 1993.
21. M. Parigot. *Proofs of strong normalization for second order classical natural deduction*. Journal of Symbolic Logic, 62 (4), pp. 1461-1479, 1997.
22. E. Polonovsky. *Substitutions explicites, logique et normalisation*. PhD thesis. Paris 7, 2004.
23. W. Py. *Confluence en $\lambda\mu$ -calcul*. PhD thesis. University of Chambéry, 1998.
24. N.J. Rehof and M.H. Sorensen. *The λ_{Δ} -calculus*. In M. Hagiya and J.C. Mitchell, editors, Proceedings of the international symposium on theoretical aspects of computer software, TACS'94, LNCS (789), pp. 516-542. Springer Verlag, 1994.
25. Y. Yamagata. *Strong Normalization of Second Order Symmetric Lambda-mu Calculus*. TACS 2001, Lecture Notes in Computer Science 2215, pp. 459-467, 2001.