Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 5(4): 3702–3713.
DOI: 10.3934/math.2020239
Received: 04 October 2019
Accepted: 09 April 2020
Published: 20 April 2020

Research article

Normalization proofs for the un-typed $\mu\mu'$-calculus

Péter Battyányi[1], **Karim Nour**[2,*]

[1] Department of Computer Science, Faculty of Informatics, University of Debrecen, Kassai út 26, 4028 Debrecen, Hungary

[2] Université Grenoble Alpes, Université Savoie Mont Blanc, CNRS, LAMA, LIMD, 73000 Chambéry, France

* **Correspondence:** Email: karim.nour@univ-smb.fr.

Abstract: A long-standing open problem of Parigot has been solved by David and Nour, namely, they gave a syntactical and arithmetical proof of the strong normalization of the untyped $\mu\mu'$-reduction. In connection with this, we present in this paper a proof of the weak normalization of the μ and μ'-rules. The algorithm works by induction on the complexity of the term. Our algorithm does not necessarily give a unique normal form: sometimes we may get different normal forms depending on our choice. We also give a simpler proof of the strong normalization of the same reduction. Our proof is based on a notion of a norm defined on the terms.

Keywords: $\lambda\mu$-calculus; $\mu\mu'$-calculus; strong normalization; weak normalization; normalization algorithm
Mathematics Subject Classification: 03B40, 03B70, 03F05, 68Q42

1. Introduction

Natural deduction is not intrinsically symmetric but Parigot has introduced the so called "Free Deduction" [1], which is completely symmetric. The $\lambda\mu$-calculus derives 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 counterpart of the μ-reduction. The μ'-reduction also corresponds to the elimination of a cut. We then get a symmetric calculus that is, together with the β-rule, 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 some 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 is no more true in the $\lambda\mu$-calculus: we can find normal terms of type N that are not Church integers. Parigot has remarked in [2] 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. In [3], Nour presents three different methods for finding the value of a classical integer in the $\lambda\mu$-calculus. It turns out that the introduction of the μ'-rule greatly simplifies the problem: a $\lambda\mu$-normal term of integer type reduces to its value when we add the μ'-rule to the calculus. We are concerned with a more relaxed form of the $\lambda\mu$-calculus proposed by de Groote [4]. His aim was twofold by this proposition: on the one hand, the more flexible μ'-reduction considered as a call-by-value rule allows to retain the property of confluency [4, 5] and enables the definition of an abstract machine for the $\lambda\mu$-calculus [4], and, on the other hand, Saurin [6] showed that the separation property could be restored, which has been lost in the Parigot-style $\lambda\mu$-calculus.

We consider here the de Groote-style $\lambda\mu$-calculus with the only rules μ and μ'. It was known that, for the un-typed calculus, the μ-reduction is strongly normalizing [5] but the strong normalization of the $\mu\mu'$-reduction in the un-typed setting was an open problem raised long ago by Parigot [2]. In his thesis, Polonovsky [7] has proved that the $\mu\tilde{\mu}$-calculus has the strong normalization property, which is the untyped part of a calculus corresponding to the classical Gentzen-style deduction via the Curry-Howard isomorphism and very similar in nature to the $\mu\mu'$-calculus. His proof was based on a modified Tait-Girard reasoning using reducibility candidates. As mentioned before, Py [5] has given a simple proof of the strong normalization of the μ-rule in itself, while David and Nour has presented in [8] an arithmetical proof of the termination of the $\mu\mu'$-reduction. 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. They also gave an arithmetical proof of the strong normalization of the $\beta\mu\mu'$-reduction in the simply typed calculus.

The proof in [8] consists of two main components: first the strong normalization of the untyped $\mu\mu'$-reduction is demonstrated and then a proof of the normalization of the $\beta\mu\mu'$-reduction is given along the same lines. In the proof in question, the normalization of the untyped $\mu\mu'$-calculus is obtained by a quite sophisticated argument with sets of alternating μ- and μ'-substitutions. Then this argument is extended to the β-reduction by reasoning with types: the lengths of the types in the range of the substitution is also taken into account as one of the members of the multiset underlying the induction. The proof of the strong normalization of the $\mu\mu'$-reduction is almost as difficult as that of the $\beta\mu\mu'$-reduction: the case of the β-rule differs from that of the $\mu\mu'$-reduction only in considerations on the lengths of the types. In our paper, we present simpler proofs for the $\mu\mu'$-rule. First of all, we demonstrate that the reduction is weakly normalizing. We give an algorithm that necessarily leads to a normal form. We remark that the algorithm does not necessarily lead to a unique normal form: depending on our choice we may obtain different normal forms of the same term. Then we prove strong normalization: instead of tracing back to the substitution that could in principle cause the term to not be strongly normalizing, as was accomplished by David and Nour [8], we establish a norm for the $\mu\mu'$-terms that is decreasing and strict inequality holds for certain subterms of the reducts. This leads to a contradiction. Intuitively, our norm gives an upper estimation on the lengths of the reduction sequences consisting of μ or μ'-redexes that are created when performing an uppermost μ or μ'-redex. By the introduction of that norm, we can eliminate the appearance of alternating sets of substitutions defined by mutual induction in [8].

Our proofs are arithmetical, which means that we use combinatorial reasoning that can be formalized in first order Peano arithmetic.

The paper is organized as follows: In the next section we give the necessary definitions, in Section 3 we demonstrate that the untyped $\mu\mu'$-reduction is weakly normalizing. We give an algorithm for obtaining a normal form of an arbitrary $\mu\mu'$-term. Then we turn to the proof of the strong normalization. Our proof considerably simplifies that of David and Nour [8] by finding a norm which estimates from above the lengths of the $\mu\mu'$-reduction sequences created by reducing the uppermost μ or μ'-redexes. We conclude with some future work.

2. The $\mu\mu'$-calculus

In this section we present the part of the $\lambda\mu\mu'$-calculus that interests us.

Definition 2.1 (Terms)**.**

1. *Let $\mathcal{V}_\lambda = \{x, y, z, \dots\}$ denote a set of λ-variables and $\mathcal{V}_\mu = \{\alpha, \beta, \gamma, \dots\}$ denote a set of μ-variables, respectively. The term formation rules are the following.*

$$\mathcal{T} \ := \ \mathcal{V}_\lambda \ \mid \ (\mathcal{T})\mathcal{T} \ \mid \ [\mathcal{V}_\mu]\mathcal{T} \ \mid \ \mu\mathcal{V}_\mu.\mathcal{T}$$

2. *The complexity of a term is defined inductively as follows:* $cxty(x) = 1$, $cxty((M)N) = cxty(M) + cxty(N) + 1$ *and* $cxty(\mu\alpha.M) = cxty([\alpha]M) = cxty(M) + 1$.
3. *For every term M, we define by induction on M the set of free μ-variables of M:* $Fv(x) = \emptyset$, $Fv((M)N) = Fv(M) \cup Fv((N)$, $Fv([\alpha]M) = Fv(M) \cup \{\alpha\}$ *and* $Fv(\mu\alpha.M) = Fv(M) \setminus \{\alpha\}$.
4. *In a term the μ operator binds the variables. We therefore consider the terms modulo equivalence under renaming of variables bound by μ.*

Remark 2.2. To better understand the intuition behind the formation of terms, we will present a typed version of this calculus, though we are concerned with the untyped calculus throughout the paper.

The types are built from atomic types and the constant $\perp$ with the connectors $\neg$ and $\rightarrow$. In the definition below Γ denotes a (possibly empty) context, that is, a set of declarations of the form $x : A$ (resp. $\alpha : \neg A$) for a λ-variable x (resp. a μ-variable α) and type A such that a λ-variable x (resp. a μ-variable α) occurs at most once in an expression $x : A$ (resp. $\alpha : \neg A$) of Γ. The typing rules are as follows.

$$\frac{}{\Gamma, x : A \ \vdash \ x : A}\ ax \qquad\qquad \frac{\Gamma \ \vdash \ M : A \rightarrow B \qquad \Gamma \ \vdash \ N : A}{\Gamma \ \vdash \ (M)N : B}\ \rightarrow_e$$

$$\frac{\Gamma, \alpha : \neg A \ \vdash \ M : A}{\Gamma, \alpha : \neg A \ \vdash \ [\alpha]M : \perp}\ \perp_i \qquad\qquad \frac{\Gamma, \alpha : \neg A \ \vdash \ M : \perp}{\Gamma \ \vdash \ \mu\alpha.M : A}\ \perp_e$$

We say that the term M is typable with type A, if there is a set of declarations Γ such that $\Gamma \ \vdash \ M : A$ holds.

The typed version of $\mu\mu'$-calculus restricts the set of terms. For example we can not write a term of the form $([\alpha]M)N$.

Definition 2.3 (Substitution)**.**

1. *A simultaneous substitution is an expression of the form*

$$\sigma = [\alpha_1 :=_{s_1} N_1, \dots, \alpha_k :=_{s_k} N_k]$$

where $\forall\, 1 \le i \le k,\ s_i \in \{l, r\}$, α_i *is a* μ*-variable and* N_i *is a term.*

2. *If* $\sigma = [\alpha_1 :=_{s_1} N_1, \dots, \alpha_k :=_{s_k} N_k]$, α *a* μ*-variable,* N *a term and* $s \in \{l, r\}$, *we denote by* $\sigma + \alpha :=_s N$ *the simultaneous substitution* $[\alpha_1 :=_{s_1} N_1, \dots, \alpha_k :=_{s_k} N_k, \alpha :=_s N]$.
3. *Let* $\sigma = [\alpha_1 :=_{s_1} N_1, \dots, \alpha_k :=_{s_k} N_k]$ *and* M *a term. We define by induction the term* $M\sigma$. *We can assume that the variables of the terms* N_i *are not linked by* μ *in the term* M.
   - *If* $M = x$, *then* $M\sigma = x$.
   - *If* $M = (M_1)M_2$, *then* $M\sigma = (M_1\sigma)M_2\sigma$.
   - *If* $M = \mu\alpha.M'$, *then* $M\sigma = \mu\alpha.M'\sigma$.
   - *If* $M = [\alpha]M'$ *and* $\alpha \notin \{\alpha_1, \dots, \alpha_k\}$, *then* $M\sigma = [\alpha]M'\sigma$.
   - *If* $M = [\alpha_i]M'$ *for* $1 \le i \le k$ *and* $s_i = r$, *then* $M\sigma = [\alpha_i](M'\sigma)N_i$.
   - *If* $M = [\alpha_i]M'$ *for* $1 \le i \le k$ *and* $s_i = l$, *then* $M\sigma = [\alpha_i](N_i)M'\sigma$.

Definition 2.4 (Redex)**.**

1. *A* μ*-redex is a term of the form* $(\mu\alpha.M)N$ *and we call* $\mu\alpha.M[\alpha :=_r N]$ *its contractum. Intuitively,* $M[\alpha :=_r N]$ *is obtained from* M *by replacing every subterm in* M *of the form* $[\alpha]U$ *by* $[\alpha](U)N$.
2. *A* μ'*-redex is a term of the form* $(N)\mu\alpha.M$ *and we call* $\mu\alpha.M[\alpha :=_l N]$ *its contractum. Intuitively,* $M[\alpha :=_l N]$ *is obtained from* M *by replacing every subterm in* M *of the form* $[\alpha]U$ *by* $[\alpha](N)U$.
3. *If a term contains no* μ*-redex (resp.* μ'*-redex), then it is said to be* μ*-normal (resp.* μ'*-normal). A term is said to be in normal form (or simply normal) if it is* μ *and* μ'*-normal. We denote by* NF *(resp.* NF_μ *or* $NF_{\mu'}$*) the set of normal terms (resp. of* μ*-normal terms or* μ'*-normal terms).*

Remark 2.5.

1. The intuitive meaning of the reduction of $(\mu\alpha.M)N$ to $\mu\alpha.M[\alpha :=_r N]$ is that the argument N of the function $\mu\alpha.M$ is passed as an argument to all the functions in M named by the symbol $[\alpha]$.
2. The intuitive meaning of the reduction of $(N)\mu\alpha.M$ to $\mu\alpha.M[\alpha :=_l N]$ is that the function N having $\mu\alpha.M$ as argument becomes the functional part of the application by every subterm of M named by the symbol $[\alpha]$.

Definition 2.6 (Reduction)**.**

1. *Let* M *and* M' *be two terms.*
   - *We write* $M \to_\mu M'$, *if* M' *is obtained from* M *by replacing a* μ*-redex of* M *by its contractum.*
   - *We write* $M \to_{\mu'} M'$, *if* M' *is obtained from* M *by replacing a* μ'*-redex of* M *by its contractum.*
2. *Let* $\to$ *stand for one of the relations* $\to_\mu, \to_{\mu'}$. *We denote by* $\twoheadrightarrow$ *(resp.* $\twoheadrightarrow_\mu$ *or* $\twoheadrightarrow_{\mu'}$*) the reflexive and transitive closure of* $\to$ *(resp.* $\to_\mu$ *or* $\to_{\mu'}$*). For example,* $M \twoheadrightarrow M'$ *if* $M \to M_1 \to M_2 \to \cdots \to M_k = M'$. *Finally, We denote by* $\twoheadrightarrow^+$ *the transitive closure of* $\to$, *i.e.* $M \twoheadrightarrow^+ M'$ *if there is* $k > 0$ *such that* $M \to M_1 \to M_2 \to \cdots \to M_k = M'$.

Remark 2.7.

1. We can show that the reduction $\twoheadrightarrow_\mu$ (resp. $\twoheadrightarrow_{\mu'}$) is confluent. On the other hand, the reduction $\twoheadrightarrow$ is not. Indeed, if x, y are two different λ-variables and $M = (\mu\alpha.x)\mu\beta.y$, then $M \rightarrow_\mu \mu\alpha.x$ and $M \rightarrow_{\mu'} \mu\beta.y$.
2. We will also be able to verify that the reduction $\twoheadrightarrow$ preserves types, i.e. if $\Gamma \vdash M : A$ and $M \twoheadrightarrow N$, then $\Gamma \vdash N : A$.
3. Note that reducing a redex in a term can:
   - modify the other redexes (perform substitutions in some of them and duplicate others),
   - create new redexes.

   We will immediately give some examples of the creation of redexes.

$$\begin{array}{lcl} ((\mu\alpha.M)N)O & \rightarrow_\mu & (\mu\alpha.M[\alpha :=_r N])O, \\ (O)(\mu\alpha.M)N & \rightarrow_\mu & (O)\mu\alpha.M[\alpha :=_r N], \\ (O)(N)\mu\alpha.M & \rightarrow_{\mu'} & (O)\mu\alpha.M[\alpha :=_l N], \\ ((N)\mu\alpha.M)O & \rightarrow_{\mu'} & (\mu\alpha.M[\alpha :=_l N])O. \end{array}$$

Definition 2.8 (Normalization)**.**

1. *A λ-term M is said to be weakly normalizable (resp. μ-weakly normalizable or μ'-weakly normalizable) if there exists $M' \in NF$ (resp. $M' \in NF_\mu$ or $M' \in NF_{\mu'}$) such that $M \twoheadrightarrow M'$ (resp $M \twoheadrightarrow_\mu M'$ or $M \twoheadrightarrow_{\mu'} M'$). We denote by WN (resp WN_μ or $WN_{\mu'}$) the set of weakly normalizable (resp. μ-weakly normalizable or μ'-weakly normalizable) terms.*
2. *A term M is said to be strongly normalizable, if there exists no infinite reduction paths starting from M. That is, any possible sequence of reductions eventually leads to a normal term. We denote by SN the set of strongly normalizable terms.*

In the sequel, we study the $\mu\mu'$-calculus with respect to the properties "weak normalization" and "strong normalization".

3. Every term is in WN

Although we know that the $\mu\mu'$-calculus has the strong normalization property, we will present a very simple demonstration of the weak normalization of this calculus. The goal is to present a simple algorithm for finding one of the normal forms of a given term.

The following result means that a simultaneous substitution in a μ or μ'-normal term cannot create a μ abstraction.

Lemma 3.1. *Let M be a term and σ a simultaneous substitution.*

1. *If $M \in NF_\mu$ and $M\sigma \twoheadrightarrow_\mu \mu\alpha.M'$ for some term M', then $M = \mu\alpha.M''$ for some term M''.*
2. *If $M \in NF_{\mu'}$ and $M\sigma \twoheadrightarrow_{\mu'} \mu\alpha.M'$ for some term M', then $M = \mu\alpha.M''$ for some term M''.*

Proof. We only prove the first item by induction on $cxty(M)$.

- It is clear that $M \neq x$ and $M \neq [\alpha]M_1$.
- If $M = (M_1)M_2$, then $M\sigma = (M_1\sigma)M_2\sigma$ and, since $M \in NF_\mu$ we have $M_1 \neq \mu\alpha.M_1'$, therefore, by IH, $M_1\sigma \not\twoheadrightarrow_\mu \mu\alpha.M_1''$, hence, $M\sigma \not\twoheadrightarrow_\mu \mu\alpha.M'$. A contradiction.

We deduce that $M = \mu\alpha.M''$ for some term M''. □

The following result is fundamental. In particular, it allows μ-normalizing a substitution when the terms in the image are normal and also characterizes the μ'-redexes of the μ-normal form obtained. We will see later that the iteration of the algorithm resulting from this lemma constitutes a normalization algorithm.

Lemma 3.2. *Let $M, N \in NF$, $\sigma = [\alpha_1 :=_r N, \ldots, \alpha_k :=_r N]$ and $\sigma' = [\beta_1 :=_l N, \ldots, \beta_k :=_l N]$.*

1. *Then $\exists\, M' \in NF_\mu$ such that $M\sigma \twoheadrightarrow_\mu M'$ and the μ'-redexes of M' are of the form $[\beta](V)N$ if $N = \mu\gamma.U$ for some term U. In particular $M\sigma \in WN_\mu$ and $M\sigma \in WN$ if $N \neq \mu\gamma.U$.*
2. *Then $\exists\, M' \in NF_{\mu'}$ such that $M\sigma' \twoheadrightarrow_{\mu'} M'$ and the μ-redexes of M' are of the form $[\beta](N)U$ if $N = \mu\gamma.V$ for some term V. In particular $M\sigma \in WN_{\mu'}$ and $M\sigma' \in WN$ if $N \neq \mu\gamma.V$.*

Proof. We only prove the first item by induction on $cxty(M)$.

- The result is obvious if $M = x$.
- If $M = \mu\alpha.M'$ or $M = [\beta]M'$ where $\beta \neq \alpha_i$ and $1 \leq i \leq k$, it is enough to apply the IH on M'.
- If $M = [\alpha_i]M'$ where $1 \leq i \leq k$, then $M\sigma = [\alpha_i](M'\sigma)N$. By IH, $\exists\, M'' \in NF_\mu$ such that $M'\sigma \twoheadrightarrow_\mu M''$. We distinguish two cases.
  - If $M'' \neq \mu\alpha.W$, then $M\sigma \twoheadrightarrow_\mu (M'')N \in NF_\mu$.
  - If $M'' = \mu\alpha.W$, then, by Lemma 3.1, $M' = \mu\alpha.W'$ and $M\sigma = [\alpha_i](\mu\alpha.W'\sigma)N \to_\mu [\alpha_i]\mu\alpha.W'[\sigma+\alpha :=_r N]$. By IH, we have $W'[\sigma+\alpha :=_r N] \in WN_\mu$, then $M\sigma \in WN_\mu$.

  The requirement for the μ'-redex can be checked in both of the above cases.
- If $M = (M_1)M_2$, then $M\sigma = (M_1\sigma)M_2\sigma$ and, by IH, $\exists\, M_1', M_2' \in NF_\mu$ such that $M_1\sigma \twoheadrightarrow_\mu M_1'$, $M_2\sigma \twoheadrightarrow_\mu M_2'$, thus $M \twoheadrightarrow_\mu (M_1')M_2'$. Since $M \in NF$, by Lemma 3.1, M_1' and M_2' do not begin with μ, hence $M \in WN_\mu$. We can easily check the property for the μ'-redexes.

□

The goal of Lemma 3.4 is to finish the normalization of a term after the application of Lemma 3.2.

Definition 3.3 (μ- and μ'-good)**.**

1. *A term M is said to be μ-good if $M \in NF_\mu$ and its μ'-redexes are of the form $[\beta](V)\mu\gamma.U$.*
2. *A term M is said to be μ'-good if $M \in NF_{\mu'}$ and its μ-redexes are of the form $[\beta](\mu\gamma.V)U$.*

Lemma 3.4.

1. *If M is μ-good, then $\exists\, M' \in NF$ such that $M \twoheadrightarrow_{\mu'} M'$ and $M' = \mu\gamma.V$ for some term V iff $M = \mu\gamma.U$ for some term U; in particular $M \in WN$.*
2. *If M is μ'-good, , then $\exists\, M' \in NF$ such that $M \twoheadrightarrow_\mu M'$ and $M' = \mu\gamma.V$ for some term V iff $M = \mu\gamma.U$ for some term U; in particular $M \in WN$.*

Proof. We only prove the first point by induction on $cxty(M)$.

- The result is obvious if $M = x$.

- If $M = \mu\alpha.N$ or $M = [\alpha]N$ where N is not a μ'-redex, then N is μ-good and it is enough to apply the IH on N.
- If $M = [\alpha](V)\mu\gamma.U$, then U and V are μ-goods and, by IH, $\exists\, U', V' \in NF$ such that $U \twoheadrightarrow_{\mu'} U'$, $V \twoheadrightarrow_{\mu'} V'$ and $V' \neq \mu\delta.W$ (since $M \in NF_\mu$), thus $M \twoheadrightarrow_{\mu'} [\alpha]\mu\gamma.U'[\gamma :=_l V']$, therefore, by point 2 of Lemma 3.2, $U'[\gamma :=_l V'] \in WN$ and $M \in WN$.
- If $M = (M_1)M_2$, then M_1 and M_2 are μ-goods and, by IH, $\exists\, M_1', M_2' \in NF$ such that $M_1 \twoheadrightarrow_{\mu'} M_1'$, $M_2 \twoheadrightarrow_{\mu'} M_2'$, $M_1' \neq \mu\delta.W_1$ (since $M \in NF_\mu$) and $M_2' \neq \mu\delta.W_2$ (since M is good), thus $M \twoheadrightarrow_{\mu'} (M_1')M_2' \in WN$ and $M \in WN$.

□

We can now prove our weak normalization result.

Theorem 3.5. *The $\mu\mu'$-calculus weakly normalizes, i.e. for every term M, we have $M \in WN$.*

Proof. Let M be a term. We prove by induction on $cxty(M)$ that $M \in WN$.

- The result is obvious if $M = x$.
- If $M = \mu\alpha.M'$ or $M = [\alpha]M'$, it is enough to apply the IH on M'.
- If $M = (M_1)M_2$, then, by IH, $\exists\, M_1', M_2' \in NF$ such that $M_1 \twoheadrightarrow M_1'$ and $M_2 \twoheadrightarrow M_2'$. We distinguish four cases.
  - If $M_1' \neq \mu\alpha.M_1''$ and $M_2' \neq \mu\beta.M_2''$, then $M \twoheadrightarrow (M_1')M_2' \in NF$.
  - If $M_1' = \mu\alpha.M_1''$ and $M_2' \neq \mu\beta.M_2''$, then $M \twoheadrightarrow (\mu\alpha.M_1'')M_2' \rightarrow_\mu \mu\alpha.M_1''[\alpha :=_r M_2']$. By Lemma 3.2, $M_1''[\alpha :=_r M_2'] \in WN$, then $M \in WN$.
  - If $M_1' \neq \mu\alpha.M_1''$ and $M_2' = \mu\beta.M_2''$, then $M \twoheadrightarrow (M_1')\mu\beta.M_2'' \rightarrow_{\mu'} \mu\beta.M_2''[\beta :=_l M_1']$. By Lemma 3.2, $M_2''[\beta :=_l M_1'] \in WN$, then $M \in WN$.
  - If $M_1' = \mu\alpha.M_1''$ and $M_2' = \mu\beta.M_2''$, then we can conclude in two different ways.
    - We have $M \twoheadrightarrow (\mu\alpha.M_1'')\mu\beta.M_2'' \rightarrow_\mu \mu\alpha.M_1''[\alpha :=_r \mu\beta.M_2'']$. By Lemma 3.2, $\exists\, T \in NF_\mu$ such that $M_1''[\alpha :=_r \mu\beta.M_2''] \twoheadrightarrow_\mu T$ and the μ'-redexes of T are of the form $[\gamma](V)\mu\beta.M_2''$, then, T is μ-good and, by Lemma 3.4, $T \in WN$, thus $M \in WN$.
    - We have $M \twoheadrightarrow (\mu\alpha.M_1'')\mu\beta.M_2'' \rightarrow_{\mu'} \mu\beta.M_2''[\beta :=_l \mu\alpha.M_1'']$. By Lemma 3.2, $\exists\, T \in NF_{\mu'}$ such that $M_2''[\beta :=_l \mu\alpha.M_1''] \twoheadrightarrow_{\mu'} T$ and the μ-redexes of T are of the form $[\gamma](\mu\alpha.M_1'')U$, then, T is μ'-good and, by Lemma 3.4, $T \in WN$, thus $M \in WN$.

□

Remark 3.6. We summarize what we did to present a normalization algorithm.

1. Lemma 3.2 gives an algorithm to μ-normalize (resp. μ'-normalize) terms of the form $M\sigma$ where $M \in NF$, $\sigma = [\alpha_1 :=_r N, \ldots, \alpha_k :=_r N]$ (resp. $\sigma = [\alpha_1 :=_l N, \ldots, \alpha_k :=_l N]$) and $N \in NF$.
2. Lemma 3.4 gives an algorithm (using the first one) to normalize μ-normal (resp. μ'-normal) terms having only μ'-redexes (resp. μ-redexes) of the form $[\beta](V)\mu\gamma.U$ (resp. $[\beta](\mu\gamma.V)U$).
3. Theorem 3.5 gives an algorithm to normalize a term M (by induction on $cxty(M)$). In the case where M is an application, the first two algorithms are successively used. Case 3 in the proof of Theorem 3.5 leaves us with a nondeterministic choice concerning the process of finding a normal

form of M. This means that instead of one uniquely determined normal form we end up with one of the possible normal forms of M.

4. Every term is in SN

In this section, we improve the result of the previous section by showing that every term is strongly normalizable.

First, we begin by stating classical and simple properties that will be used in our proofs. We will not detail the proofs of the first two lemmas. The reader is referred to [2] and [5] for the proofs of these results.

Lemma 4.1. *Let M, N be terms, $s \in \{r, l\}$ and σ a simultaneous substitution. Then $M[\alpha :=_s N]\sigma = M\sigma[\alpha :=_s N\sigma]$.*

Proof. By induction on $cxty(M)$. □

Lemma 4.2. *Let M, N, M', N' be terms such that $M \twoheadrightarrow M'$ and $N \twoheadrightarrow N'$, and $s \in \{r, l\}$. Then $M[\alpha :=_s N] \twoheadrightarrow M[\alpha :=_s N']$, $M[\alpha :=_s N] \twoheadrightarrow M'[\alpha :=_s N]$ and $M[\alpha :=_s N] \twoheadrightarrow M'[\alpha :=_s N']$.*

Proof. We prove the first two properties for one step reductions by induction on $cxty(M)$. The third one comes directly from the first two properties. □

Lemmas 4.3, 4.4, 4.6 and 4.9 below can also be found in [8]. They help us explain why a term might not be in SN. In order to make our presentation self-contained, we recall the proofs of the lemmas from [8], perhaps, in a bit more detailed way. Lemma 4.3 says that an application reduces to a μ-abstraction only if either its left or right member reduces to a μ-abstraction.

Lemma 4.3. *Let $(M)N \twoheadrightarrow \mu\alpha.P$. Then either $M \twoheadrightarrow \mu\alpha.M_1$ and $M_1[\alpha :=_r N] \twoheadrightarrow P$ or $N \twoheadrightarrow \mu\alpha.N_1$ and $N_1[\alpha :=_l M] \twoheadrightarrow P$.*

Proof. By induction on the length of the reduction $(M)N \twoheadrightarrow \mu\alpha.P$.
If $(M)N \rightarrow O \twoheadrightarrow \mu\alpha.P$, we distinguish several cases.

- If $O = (M')N$ where $M \rightarrow M'$ or $O = (M)N'$ where $N \rightarrow N'$, then we simply apply IH on O, then, by Lemma 4.2, we obtain the result.
- If $M = \mu\alpha.M_1$ and $O = \mu\alpha.M_1[\alpha :=_r N]$, then $M_1[\alpha :=_r N] \twoheadrightarrow P$.
- If $N = \mu\alpha.N_1$ and $O = \mu\alpha.N_1[\alpha :=_l M]$, then $N_1[\alpha :=_r M] \twoheadrightarrow P$.

□

The next lemma generalizes Lemma 3.1.

Lemma 4.4. *Let M be a term and σ a simultaneous substitution. If $M\sigma \twoheadrightarrow \mu\alpha.P$ for some P, then there exists a Q such that $M \twoheadrightarrow \mu\alpha.Q$ and $Q\sigma \twoheadrightarrow P$.*

Proof. By induction on $cxty(M)$. The only possibilities are $M = \mu\alpha.M_1$ and $M = (M_1)M_2$. The former case is trivial. In the latter, we have $(M_1\sigma)M_2\sigma \twoheadrightarrow \mu\alpha.P$. By Lemma 4.3, either $M_1\sigma \twoheadrightarrow \mu\alpha.N_1$, $N_1[\alpha :=_r M_2\sigma] \twoheadrightarrow P$ or $M_2\sigma \twoheadrightarrow \mu\alpha.N_2$, $N_2[\alpha :=_l M_1\sigma] \twoheadrightarrow P$. Suppose the former holds. By IH, there is an R such that $M_1 \twoheadrightarrow \mu\alpha.R$ and $R\sigma \twoheadrightarrow N_1$. Then, by Lemmas 4.1 and 4.2, $M \twoheadrightarrow \mu\alpha.R[\alpha :=_r M_2]$ and $R[\alpha :=_r M_2]\sigma = R\sigma[\alpha :=_r M_2\sigma] \twoheadrightarrow N_1[\alpha :=_r M_2]\sigma \twoheadrightarrow P$. Our assertion holds with $Q = R[\alpha :=_r M_2]$. □

Definition 4.5 (Function η)**.** *If a term M is strongly normalizable, then, since the reduction tree of M is locally finite by König Lemma, we can denote the length of the longest reduction sequence of M by $\eta(M)$.*

The previous definition helps us to demonstrate properties of strongly normalizable terms. Indeed if $M \to M'$, then $\eta(M) > \eta(M')$.

Lemma 4.6. *Let M, N be terms such that $M, N \in SN$ and $(M)N \notin SN$. Then either $M \twoheadrightarrow \mu\alpha.M'$ such that $\mu\alpha.M'[\alpha =_r N] \notin SN$ or $N \twoheadrightarrow \mu\beta.N'$ such that $\mu\beta.N'[\beta =_l M] \notin SN$.*

Proof. By induction on $\eta(M) + \eta(N)$. If $M \to M'$ and $(M')N \notin SN$, then $\eta(M') < \eta(M)$ and the IH applies. The situation is similar when $N \to N'$ and $(M)N' \notin SN$. If $M = \mu\alpha.M_1$ and $\mu\alpha.M_1[\alpha :=_r N] \notin SN$, or $N = \mu\beta.N_1$ and $\mu\beta.N_1[\beta :=_l M] \notin SN$, then the result is obvious. □

Definition 4.7 (Relation $\sqsubset$ and $\prec$)**.**

1. *We use the notation $N \sqsubseteq M$ if N is a subterm of M, and the notation $N \sqsubset M$ if N is a subterm of M other than M.*
2. *Let M, N be terms. The notation $N \prec M$ will signify the fact that there is an M' such that $M \twoheadrightarrow M' \sqsupseteq N$ holds and either $M \twoheadrightarrow^+ M'$ or $N \sqsubset M$ is valid. The symbol $\preceq$ will be the reflexive closure of $\prec$. The relations $\prec$ and $\preceq$ are transitive. Moreover, $N \preceq M$ iff there is an M' such that $M \twoheadrightarrow M'$ and $N \sqsubseteq M'$.*

Remark 4.8. The above definition should be made precise by talking about subterm occurrences addressed by a finite list of index symbols. To facilitate understanding, we ignore the exact treatment of subterm occurrences together with the problem of variable collisions induced by the substitutions. Obviously, a nameless representation of terms would eliminate all these difficulties.

Lemma 4.9. *Let $M, N \in SN$ such that $M[\alpha :=_r N] \notin SN$ (resp. $M[\alpha :=_l N] \notin SN$) for some α. Then there is an $[\alpha]M_1 \preceq M$ for which $M_1[\alpha :=_r N] \in SN$ (resp. $M_1[\alpha :=_l N] \in SN$) and $(M_1[\alpha :=_r N])N \notin SN$ (resp. $(M_1[\alpha :=_l N])N) \notin SN$).*

Proof. The proof proceeds by induction on $\langle cxty(M), \eta(M)\rangle$. Let us only treat the case of $M[\alpha :=_r N] \notin SN$.

- If $M = \mu\beta.M_1$, the result is trivial.
- If $M = [\beta]M_1$, then we have two cases to distinguish.
  - If $\beta = \alpha$, then if $M_1[\alpha :=_r N] \notin SN$, the IH applies. Otherwise our assertion follows with the M_1 under discussion.
  - If $\beta \neq \alpha$, then the IH gives the result.
- If $M = (M_1)M_2$, then, by Lemma 4.6, either $M_1[\alpha :=_r N] \twoheadrightarrow \mu\beta.M'_1$ and $\mu\beta.M'_1[\beta :=_r M_2[\alpha :=_r N]] \notin SN$ or $M_2[\alpha :=_r N] \twoheadrightarrow \mu\gamma.M'_2$ and $\mu\gamma.M'_2[\gamma :=_l M_1[\alpha :=_r N]] \notin SN$. Suppose the former case holds, the latter being similar. Then, by Lemma 4.4, there is an M_3 such that $M_1 \twoheadrightarrow \mu\beta.M_3$ and $M_3[\alpha :=_r N] \twoheadrightarrow M'_1$. By this we have, by Lemmas 4.1 and 4.2, $\mu\beta.M_3[\beta :=_r M_2][\alpha :=_r N] = \mu\beta.M_3[\alpha :=_r N][\beta :=_r M_2[\alpha :=_r N]] \twoheadrightarrow \mu\beta.M'_1[\beta :=_r M_2[\alpha :=_r N]] \notin SN$. But then, since $\eta(\mu\beta.M_3[\beta :=_r M_2]) < \eta(M)$, we can apply the IH.

$\square$

Our new proof of the strong normalization of the $\mu\mu'$-calculus is based on the introduction of a norm for the terms which does not increase through a reduction.

Definition 4.10 (Norm for terms)**.** *Let M be a term. Let us define a norm for M, denoted by $|M|$, by induction on $cxty(M)$:*

$$|M| = \begin{cases} 0 & \text{if} \;\; M = x, \\ |M_1| + |M_2| & \text{if} \;\; M = (M_1)M_2, \\ max\{|M_2| \mid [\alpha]M_2 \sqsubseteq M_1\} + 1 & \text{if} \;\; M = \mu\alpha.M_1 \text{ and } \alpha \in Fv(M_1), \\ 0 & \text{if} \;\; M = \mu\alpha.M_1 \text{ and } \alpha \notin Fv(M_1), \\ 0 & \text{if} \;\; M = [\alpha]M_1. \end{cases}$$

For every M the norm of M is a natural number. Intuitively, by this norm we can find an upper bound for the lengths of the reduction sequences consisting of the redexes created from top to bottom when performing an uppermost μ- or μ'-redex.

The main idea of our proof is to demonstrate that the norm is non-increasing with respect to μ and μ'-redexes, and, furthermore, we show that it strictly decreases on certain subterms of the contractum. Namely, assuming the uppermost redex is $(\mu\alpha.M)N$, then the subterms $U[\alpha :=_r N]$, where $[\alpha]U \sqsubseteq M$, have smaller norms than $(\mu\alpha.M)N$. Similarly for $(N)\mu\alpha.M$. This will lead to a contradiction, since if we assume that we have an application $(M)N$ of minimal norm which is not in SN and we may assume that $M = \mu\alpha.M'$. Then, by Lemma 4.9, we can find a subterm of the contractum of $(M)N$ which is not in SN and is such that its norm is strictly less than that of $(M)N$.

The following lemma is very simple but worth noting.

Lemma 4.11. *Let $[\alpha]M_1 \sqsubseteq M[\beta :=_s N]$ and $s \in \{r, l\}$.*

1. *If $\alpha \neq \beta$, then $M_1 = M_2[\beta :=_s N]$ where $[\alpha]M_2 \sqsubseteq M$.*
2. *If $\alpha = \beta$, then $M_1 = (M_2[\beta :=_s N])N$ if $s = r$ and $M_1 = (N)M_2[\beta :=_s N]$ if $s = l$ where $[\alpha]M_2 \sqsubseteq M$.*

Proof. By induction on $cxty(M)$. $\square$

We now show how the norm we defined behaves with the reductions.

Lemma 4.12. *Let M, N be terms and $s \in \{r, l\}$. Then $|M[\alpha :=_s N]| = |M|$.*

Proof. We accomplish the proof for $s = r$. The proof goes by induction on $cxty(M)$.
The only interesting case is $M = \mu\beta.M_1$.

- If $\beta \in Fv(M_1)$, then, by Lemma 4.11 and applying the IH,

$$\begin{aligned} |\mu\beta.M_1[\alpha :=_r N]| &= max\{|M_1'| \;/\; [\beta]M_1' \sqsubseteq M_1[\alpha :=_r N]\} + 1 \\ &= max\{|M_2[\alpha :=_r N]| \;/\; [\beta]M_2 \sqsubseteq M_1\} + 1 \;=\; |\mu\beta.M_1|. \end{aligned}$$

- In case of $\beta \notin Fv(M_1)$, the equation $|\mu\beta.M_1[\alpha :=_r N]| = |\mu\beta.M_1| = 0$ is valid.

$\square$

The next lemma implies that, if we have a redex $(\mu\alpha.M)N$, then, for every $[\alpha]M_1 \sqsubseteq M$, the norm of $([\alpha]M_1)[\alpha :=_r N]$ will be strictly less than that of $(\mu\alpha.M)N$. The situation is similar for the topmost μ'-redex.

Lemma 4.13. *Let M, M', N be terms and $s \in \{r, l\}$. If $[\alpha]M' \sqsubseteq M$, then $|\mu\alpha.M| > |M'[\alpha :=_s N]|$.*

Proof. By Lemma 4.12, we have $|\mu\alpha.M| = max\{|P| \ / \ [\alpha]P \sqsubseteq M\} + 1 > |M'| = |M'[\alpha :=_s N]|$. □

The following lemma states that the norm of the contractum is not greater than that of the redex.

Lemma 4.14. *Let M, N be terms. Then $|(\mu\alpha.M)N| \geq |\mu\alpha.M[\alpha :=_r N]|$ and $|(M)\mu\alpha.N| \geq |\mu\alpha.N[\alpha :=_l M]|$.*

Proof. We deal with the case of the μ-reduction only.

- Let us suppose first $\alpha \in Fv(M)$. By Lemmas 4.11 and 4.12,

$$\begin{aligned}
|\mu\alpha.M[\alpha :=_r N]| &= max\{|M_1| \ / \ [\alpha]M_1 \sqsubseteq M[\alpha :=_r N]\} + 1 \\
&= max\{|(M_2[\alpha :=_r N])N| \ / \ [\alpha]M_2 \sqsubseteq M\} + 1 \\
&= max\{|M_2| \ / \ [\alpha]M_2 \sqsubseteq M\} + |N| + 1 \\
&= |\mu\alpha.M| + |N| \ = \ |(\mu\alpha.M)N|.
\end{aligned}$$

- If $\alpha \notin Fv(M)$, then $|(\mu\alpha.M)N| = |\mu\alpha.M| + |N| = |N| \geq |\mu\alpha M[\alpha :=_r N]| = 0$.

□

As a consequence, we can assert that the norm is not increasing regarding the reduction sequences.

Lemma 4.15. *Let $M \twoheadrightarrow N$. Then $|M| \geq |N|$.*

Proof. It is enough to show that $M \to N$ implies $|M| \geq |N|$. The proof goes by induction on $cxty(M)$. The only interesting case is $M = (M_1)M_2$.

- If $M_1 \to M_1'$ or $M_2 \to M_2'$, it's obvious.
- If $M_1 = \mu\alpha.M_3$, $N = \mu\alpha.M_3[\alpha :=_r M_2]$, or $M_2 = \mu\beta.M_3$, $N = \mu\beta.M_3[\alpha :=_l M_1]$, applying Lemma 4.14, we obtain the result.

□

We can now prove the strong normalization result.

Theorem 4.16. *The $\mu\mu'$-calculus has the strong normalization property, i.e. for every term M, we have $M \in SN$.*

Proof. It is enough to prove that, for arbitrary $M, N \in SN$, $(M)N \in SN$ as well. Let $M, N \in SN$ such that $(M)N \notin SN$, with the property that $\langle |(M)N|, \eta(M) + \eta(N) \rangle$ is minimal. Then, by Lemma 4.9, either $M \twoheadrightarrow \mu\alpha.M'$ and $\mu\alpha.M'[\alpha =_r N] \notin SN$ or $N \twoheadrightarrow \mu\beta.N'$ and $\mu\beta.N'[\beta =_l M] \notin SN$. Suppose the former is valid. On account of Lemma 4.15, $M \twoheadrightarrow^+ \mu\alpha M'$ contradicts the minimality of $(M)N$, so we may assume $M = \mu\alpha M'$. In accordance with Lemma 4.9, there exists an $[\alpha]M_1 \preceq M'$ such that $M_1[\alpha :=_r N] \in SN$ and $(M_1[\alpha :=_r N])N \notin SN$. By Lemma 4.13, this contradicts the minimality of $(M)N$ again. □

5. Conclusions

We have presented in this paper a rather simple algorithm for normalizing a term. We have also found a very simple proof of the strong normalization of the $\mu\mu'$-calculus. We know that if we add to our calculus the simplification rules ρ and θ, we lose the strong normalization property [9]. It is therefore interesting to show that this calculus is weakly normalizing and to look for algorithms that terminate on every term. As future work, it also seems to be promising to study the weak and the strong normalization properties of the $\lambda\mu\mu'$-calculus (with the reduction β) in a typed frame.

Conflict of interest

The authors declare no conflict of interest in this paper.

References

1. M. Parigot, *Free Deduction: An Analysis of "Computations" in Classical Logic*, In: A. Voronkov, editors. Logic Programming Lecture Notes in Artificial Intelligence 592, Berlin, Heidelberg: Springer-Verlag, 1992, 361–380.
2. M. Parigot, *$\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction*, In: A. Voronkov, editors. Logic Programming and Automated Reasoning, LPAR 1992, Lecture Notes in Artificial Intelligence 624, Berlin, Heidelberg: Springer-Verlag, 1992, 190–201.
3. K. Nour, *La valeur d'un entier classique en $\lambda\mu$-calcul*, Archive Math. Logic, **36** (1997), 461–473.
4. P. de Groote, *An environment machine for the $\lambda\mu$-calculus*, Math. Struct. Comput. Sci., **8** (1998), 637–669.
5. W. Py, Confluence en $\lambda\mu$-calcul [dissertation], University of Chambéry, 1998, 117.
6. A. Saurin, *Böhm theorem and Böhm trees for the $\Lambda\mu$-calculus*, Theor. Comput. Sci., **435** (2012), 106–138.
7. E. Polonovsky, Substitutions explicites, logique et normalisation [dissertation], Paris 7, 2004, 257.
8. R. David, K. Nour, *Arithmetical proofs of strong normalization results for symmetric lambda calculi*, Fundamenta Informaticae, **77** (2007), 489–510.
9. P. Battyányi, Normalization properties of symmetric logical calculi [dissertation], University of Chambéry, 2007, 118.