



HAL
open science

Normalization in the simply typed λ -calculus

$\lambda\mu\mu'\rho\theta\epsilon$ -calculus

Péter Battyányi, Karim Nour

► **To cite this version:**

Péter Battyányi, Karim Nour. Normalization in the simply typed λ -calculus $\lambda\mu\mu'\rho\theta\epsilon$ -calculus. *Mathematical Structures in Computer Science*, 2022, 32, pp.1066 - 1098. 10.1017/s096012952200041x . hal-04727997

HAL Id: hal-04727997

<https://hal.science/hal-04727997v1>

Submitted on 9 Oct 2024

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.

PAPER

Normalization in the simply typed $\lambda\mu\mu'\rho\theta\varepsilon$ -calculus

Péter Battyányi^{1*}  and Karim Nour² 

¹Department of Computer Science, Faculty of Informatics, University of Debrecen, Kassai út 26, 4028 Debrecen, Hungary
and ²Université Savoie Mont Blanc, CNRS, LAMA, LIMD, 73000 Chambéry, France

*Corresponding author. Email: battyanyi.peter@inf.unideb.hu

(Received 19 November 2020; revised 1 July 2022; accepted 10 December 2022)

*Dedicated to the memory of Prof. René David, who passed away
this year and who was present, as a doctoral advisor of the first
author, at the beginning of this work*

Abstract

In this paper, in connection with the program of extending the Curry–Howard isomorphism to classical logic, we study the $\lambda\mu$ -calculus of Parigot emphasizing the difference between the original version of Parigot and the version of de Groote in terms of normalization properties. In order to talk about a satisfactory representation of the integers, besides the usual β -, μ -, and μ' -reductions, we consider the $\lambda\mu$ -calculus augmented with the reduction rules ρ , θ and ε . We show that we need all of these rules for this purpose. Then we prove that, with the syntax of Parigot, the calculus enjoys the strong normalization property even when we add the rules ρ , θ , and ε , while the $\lambda\mu$ -calculus presented with the more flexible de Groote-style syntax, in contrast, has only the weak normalization property. In particular, we present a normalization algorithm for the $\beta\mu\mu'\rho\theta\varepsilon$ -reduction in the de Groote-style calculus.

Keywords: $\lambda\mu$ -calculus; classical logic; strong normalization; weak normalization; normalization algorithm

1. Introduction

The works of Murthy (1991) and Griffin (1990) revealed in the nineties that the Curry–Howard correspondence can be extended to classical logic. Since then, many different calculi have appeared with the aim of representing natural deduction or Gentzen-style derivations in classical logic (Barbanera and Berardi 1994; Curien and Herbelin 2000; Munch-Maccagnoni 2013; Parigot 1992; Rehof and Sørensen 1994; Saurin 2008; Wadler 2003, 2005). The systems based on the sequent calculus, like the $\bar{\lambda}\mu\tilde{\mu}$ -calculus of Curien–Herbelin (Curien and Herbelin 2000), have the advantage of possessing symmetric reduction rules, which makes their study more pleasant. This is also the case for the system of Barbanera–Berardi (Barbanera and Berardi 1994), which is exclusively based on two connectives \wedge and \vee . Parigot’s $\lambda\mu$ -calculus (Parigot 1992) was originally defined as a calculus for establishing connections between logical calculi and natural deduction-style proofs in second-order predicate logic. The $\lambda\mu$ -calculus can be considered as a simple extension of the λ -calculus: one obtains the $\lambda\mu$ -calculus by adding classical variables (so-called μ -variables) to the λ -calculus together with their term formation rules and reduction rules.

The $\lambda\mu$ -calculus has two distinct presentations. One of them is the original calculus of Parigot, and the other one was introduced by de Groote by relaxing some of the term formation rules. The

investigation of the difference between the two presentations in terms of normalization properties can be considered as the main aim of the paper. It turns out that, despite their great resemblance, the two calculi differ in important respects: if one adds to the β -, μ -, and μ' -reductions some structural rules, e.g., the ρ - or ε -rules, which are necessary to ensure the uniqueness of the representation of data, one can find that the Parigot-style calculus retains its strong normalization property, whereas the de Groote-style calculus does not. We will return to this question in several parts of the paper, and we will give the precise definitions for the different calculi in Sections 2 and 4.

Regarding the Parigot-style $\lambda\mu$ -calculus, originally, his system contains only two reductions (the β - and μ -rules). They stem from cut eliminations with respect to the implicational fragment of natural deduction both in the intuitionistic and in the classical case. He adds two more rules (ρ and θ), which he calls simplification rules, and which correspond to cut elimination regarding negation. From a logical point of view, the $\lambda\mu$ -calculus makes it possible to easily encode the mathematical proofs. However, the calculus is not symmetric, which makes the examination of its properties more difficult. We note that we are concerned with the simply typed version of the $\lambda\mu$ -calculus throughout the paper.

The simply typed $\lambda\mu$ -calculus enjoys many useful proof-theoretical properties, like confluence (Parigot 1993; Py 1998), subject reduction, and strong normalization (Parigot 1993). The calculus has, from the computer science point of view, a severe drawback: it does not meet the requirement of the unique representation of data. In the second-order typed λ -calculus, any term having the integer type \mathcal{N} is β -equal to a Church numeral. This property enables the automatic generation of programs for the integers. In fact, a λ -term of type $\mathcal{N} \rightarrow \mathcal{N}$ is a program that represents a function for the integers: it takes a Church-integer and returns, after reduction, another one. This is no longer true in the $\lambda\mu$ -calculus: we can find normal terms of integer type that are not Church numerals. This should not come as a surprise, since we are able to prove that a type is \mathcal{N} by indirect reasoning. Hence, when we consider a type even as simple as \mathcal{N} , it is only the intuitionistic proof of the type that we can recover by utilizing exclusively the existing reduction rules.

There are several possible solutions to this problem. One can define an algorithm that, to every classical integer, assigns its value, which is represented by a Church numeral (Nour 1997; Parigot 1993). The other possibility is to apply output operators in order to transform a classical integer into an intuitionistic one, namely, into a Church numeral (Nour 1997; Parigot 1993). We choose here another solution which was also proposed by Parigot (1993). If we add the reduction rule μ' to the $\lambda\mu$ -calculus, which is the symmetric counterpart of the μ -rule and can be justified by the underlying logical rules, we get closer to reducing the Church integers to normal forms by simply applying the rules of the calculus. We discuss this question in more detail in Section 3. Similar assertions hold for the other algebraic data types defined formally in Krivine (1994).

The calculus augmented with the μ' -rule preserves the subject reduction property (at least for the simply typed version), but confluence is lost. The μ' -rule, however, plays an important role in the presentation of the call-by-value $\lambda\mu$ -calculus. It permits a calculated value, as the functional part, to find its argument. De Groote (1998) managed to integrate this rule with the $\lambda\mu$ -calculus while preserving confluence. Py studied the call-by-value μ' -reduction in his thesis (Py 1998). In particular, he proved confluence and strong normalization properties for μ - and μ' -reductions. He also enriched the syntax with new operators so that the subject reduction property was preserved and computational contents were given to the quantifiers.

Parigot (1997) proved by an application of the Tait-Girard reducibility argument that the strong normalization property holds for the $\lambda\mu$ -calculus with second-order type assignments. It was a long standing open problem of Parigot whether the $\mu\mu'$ -rules in the untyped calculus enjoy the strong normalization property. This was solved by David and the second author (David and Nour 2005, 2007). Subsequently, we have found in Battyányi and Nour (2020) another proof for this

result. Until now, no simple weak normalization algorithm has been known for the $\mu\mu'$ -reduction. In this paper, we also provide one.

As was mentioned, there has mainly been two different presentations of the $\lambda\mu$ -calculus. Originally, Parigot defined his calculus with severe restrictions on the term formation rules. His syntax, however, does not make it possible to encode with typed $\lambda\mu$ -terms any proof of the classical tautology $\neg\neg A \rightarrow A$ without free variables of type $\neg\perp$ (similarly for some other classical tautologies). Furthermore, in this syntax with many reduction rules Böhm's theorem does not hold. In fact, Py (1998) gave two distinct closed normal terms M and N for which there is no term L , such that $(L)M$ reduces to the first projection $\lambda x.\lambda y.x$ and $(L)N$ reduces to the second projection $\lambda x.\lambda y.y$. De Groote proposed a syntax that is less restrictive than that of Parigot's by splitting the rule concerning negation into two: one for the introduction and one for the elimination of negation and he even omitted certain restrictions that applied to the original calculus of Parigot. In this calculus, new rules can be considered like the ε -rule, which allows to handle two consecutive proofs both leading to absurdity. The version proposed by de Groote (1998) allowed him to construct an abstract machine for the $\lambda\mu$ -calculus. Additionally, Saurin (2012) proved that the de Groote version enjoys the separation property. Taking all these into account, we have adopted the de Groote-style calculus for the majority of our work. Our results also show that the two calculi are quite different: one of them has the strong normalization property while the other one only possesses the weak normalization property. The paper is structured in the following way.

- In the second section, we present the definitions necessary for our discussion.
- The third section demonstrates the role of the structural rules: by making use of the additional rules we can prove that every term of integer type that weakly normalises behaves like a Church numeral in the sense that when it occurs in applications formed with terms of appropriate types it reduces to the well-known form of Church numerals.
- The fourth section contains a proof that the Parigot-style $\lambda\mu$ -calculus enjoys the strong normalization property. The proof generalizes Py's method (Py 1998): we prove that the simplification rules can be strongly postponed with respect to the β -, μ - and μ' -rules and then we resort to the already mentioned result of David and the second author, which states that the de Groote-style $\lambda\mu\mu'$ -calculus enjoys the strong normalization property (David and Nour 2005, 2007). The Parigot-style $\lambda\mu\mu'$ -calculus is a subsystem of the de Groote-style $\lambda\mu\mu'$ -calculus, hence, its strong normalization property follows.
- In the subsequent sections, we discuss the normalization properties of the de Groote version of the symmetric calculi $\mu\mu'$ and $\lambda\mu\mu'$ when augmented with the structural rules ρ , θ , and ε . We show by an example that $\mu\mu'\rho\varepsilon$ -reduction is not strongly normalizing. Then we construct explicit algorithms in order to demonstrate that the weak normalization property for $\mu\mu'\rho\varepsilon$ -reduction is still valid. We point out that our algorithm allows finding several normal forms for the same term.
- In the last section, we extend our result to the case of β -reduction in the setting of typed calculus. More precisely, we show that, if we alternate our weak normalization algorithm for $\mu\mu'\rho\varepsilon$ -reduction together with an arbitrary one for β -reduction, the process necessarily terminates. In fact, performing these reductions strictly reduces the maximal rank of β -redexes. The proofs of this section are rather technical and are based on a fine analysis of the origin of the redexes in the reduction sequences. We observe that θ -reduction does not give rise to any particular problem and we can postpone it. Our method allows us to give a weak normalization algorithm for the complete calculus.

Some of the results of this paper, especially the results in Sections 4, 5, and 6, have already been presented in the thesis of the first author (Battyányi 2007). We would like to clarify, however, that these results have not been published before and some of the proofs of the thesis were even incorrect. We found that it was very difficult to correct them using the same methods. The

method applied here for proving the different variants of weak normalization properties are totally different. The proof of weak normalization of $\mu\mu'\rho\theta\varepsilon$ -reduction is direct and gives an easy algorithm. The demonstration of weak normalization of $\beta\mu\mu'\rho\theta\varepsilon$ -reduction does not utilize residuals (which are quite complicated to properly formulate), rather we proceed, in most of the cases, by a simple induction on the structure of terms.

2. The $\lambda\mu$ -Calculus

The $\lambda\mu$ -calculus was introduced by Parigot (1992) with the intention of providing a term encoding for classical natural deduction. In our paper, we restrict ourselves to the simply typed calculus. The $\lambda\mu$ -calculus possesses two kinds of variables: one of them is the original set of λ -variables, the so called intuitionistic variables, and the other kind of variables is the set of μ -variables, which are termed classical variables. Parigot’s calculus has certain restrictions imposed on the term formation rules: namely, a μ -abstraction must be followed by an application with a μ -variable and vice versa. That is, a term $\mu\alpha.M$ can only be of the form $\mu\alpha.[\beta]N$ for some N .

De Groote (1998) has proposed a new version of the $\lambda\mu$ -calculus by modifying this syntax. Namely, the term formation rules became more flexible: a μ -abstraction can now be followed by an arbitrary term (in the untyped version); hence, it does not need to be of the form $\mu\alpha.[\beta]N$. Since in the greater part of the paper we are concerned with the de Groote version, we introduce the de Groote-style calculus in what follows. In Section 4, we deal with the Parigot-style calculus. We postpone defining the Parigot-style syntax until then.

Definition 2.1 ($\lambda\mu$ -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 $\lambda\mu$ -term formation rules are the following.

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

We decided to adopt Krivine’s notation for the applications, i.e., we write $(M)N$ instead of (MN) if we apply M to N .

2. In a $\lambda\mu$ -term, the λ and μ operators bind the variables in their scope. We therefore consider terms modulo equivalence, which allows to rename the variables bound by a λ - or a μ -abstraction.
3. For every $\lambda\mu$ -term M , we define by induction on M the set $\text{fv}(M)$ of free μ -variables of M : $\text{fv}(x) = \emptyset$, $\text{fv}(\lambda x.M) = \text{fv}(M)$, $\text{fv}((M)N) = \text{fv}(M) \cup \text{fv}(N)$, $\text{fv}([\alpha]M) = \text{fv}(M) \cup \{\alpha\}$ and $\text{fv}(\mu\alpha.M) = \text{fv}(M) \setminus \{\alpha\}$.

The notion of subterm plays a very important role in our paper. Many of our subsequent results require a fine analysis of the propagation and creation of certain subterms. A subterm N of a term M is a term that appears in the formation tree for M . Obviously, M can contain several identical subterms. However, when we consider a subterm N of M , we intend to take only one particular occurrence of N in M .

Definition 2.2 (Subterms).

1. We apply the notation $N \preceq M$ if N occurs as a subterm in the $\lambda\mu$ -term M . We call this occurrence strict and write $N < M$ if, in addition, $M \neq N$.
2. Let $N \preceq M$ and $\alpha \in \mathcal{V}_\mu$. An occurrence of the $\lambda\mu$ -term N in M is said to be named with α , if it is preceded by $[\alpha]$, that is, $[\alpha]N \preceq M$ for that occurrence.

Definition 2.3 (Type system). *The types are built from a set $\mathcal{V}_{\mathbb{T}}$ of atomic types and the constant \perp with the connectives \neg and \rightarrow . The type formation rules are the following.*

$$\mathbb{T} := \mathcal{V}_{\mathbb{T}} \cup \{\perp\} \mid \neg\mathbb{T} \mid \mathbb{T} \rightarrow \mathbb{T}$$

In the definition below Γ denotes a (possibly empty) context, that is, a finite 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.

$$\frac{}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \rightarrow B} \qquad \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash (M)N : B}$$

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

We say that the $\lambda\mu$ -term M is typable with type A , if there is a context Γ and a derivation tree such that the uppermost nodes of the tree are axioms and the bottom node is $\Gamma \vdash M : A$.

Observe that, in the typed $\lambda\mu$ -calculus, not every term is accepted as well typed. For example, we cannot write a $\lambda\mu$ -term of the form $([\alpha]M)N$ or $\mu\alpha.\lambda x.M$.

In this paper, we work with several kinds of substitutions. All of them are implicitly understood with the necessary renaming of bound variables to avoid the capturing of the free variables coming from the substitution. We present μ -substitution in detail. We need it in Definition 2.6.

Definition 2.4 (μ -substitution).

1. *A μ -substitution is an expression of the form $\sigma = [\alpha :=_s N]$ where $s \in \{l, r\}$, α is a μ -variable and N is a $\lambda\mu$ -term.*
2. *Let $\sigma = [\alpha :=_s N]$ and let M be a $\lambda\mu$ -term. We define by induction the $\lambda\mu$ -term $M\sigma$. We adopt the convention of renaming bound variables before a substitution so that no variable collision occurs. Then we can assume that the free variables of the $\lambda\mu$ -term N and variable α are not bound by any μ -abstraction in the $\lambda\mu$ -term M .*
 - *If $M = x$, then $M\sigma = x$.*
 - *If $M = \lambda x.M'$, then $M\sigma = \lambda x.M'\sigma$.*
 - *If $M = (M_1)M_2$, then $M\sigma = (M_1\sigma)M_2\sigma$.*
 - *If $M = \mu\beta.M'$, then $M\sigma = \mu\beta.M'\sigma$.*
 - *If $M = [\beta]M'$ and $\beta \neq \alpha$, then $M\sigma = [\beta]M'\sigma$.*
 - *If $M = [\alpha]M'$ and $s = r$, then $M\sigma = [\alpha](M'\sigma)N$.*
 - *If $M = [\alpha]M'$ and $s = l$, then $M\sigma = [\alpha](N)M'\sigma$.*

We adopt the convention that substitution has higher precedence than application and abstraction.

In order to define ε -reduction, we need the following notion.

Definition 2.5 (α -translation). Let M be a $\lambda\mu$ -term and α a μ -variable. We define the α -translation M_α of M by induction on M .

- If $M = x$, then $M_\alpha = x$.
- If $M = \lambda x.M'$, then $M_\alpha = \lambda x.M'_\alpha$.
- If $M = (P)Q$, then $M_\alpha = (P_\alpha)Q_\alpha$.
- If $M = \mu\beta.M'$, then $M_\alpha = \mu\beta.M'_\alpha$.
- If $M = [\beta]M'$ and $\beta \neq \alpha$, then $M_\alpha = [\beta]M'_\alpha$.
- If $M = [\alpha]M'$, then $M_\alpha = M'_\alpha$.

Intuitively, M_α is the result of replacing every subterm $[\alpha]N$ in M with N .

In what follows, we define the redexes examined in this paper and the reductions induced by them. Next, we recall some results about the $\lambda\mu$ -calculus. Regarding the necessary notions and definitions and the λ -calculus analogues of the results listed below, the reader is referred to the standard textbooks, for example Girard et al. (1989), Krivine (1993).

Definition 2.6 (Redex).

1. A β -redex is a $\lambda\mu$ -term of the form $(\lambda x.M)N$ and we call $M[x := N]$ its contractum. The $\lambda\mu$ -term $M[x := N]$ is obtained from M by replacing every free occurrence of x in M by N . This substitution is sometimes referred to as β -substitution.
2. A μ -redex is a $\lambda\mu$ -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 $[\alpha]P$ by $[\alpha](P)N$.
3. A μ' -redex is a $\lambda\mu$ -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 $[\alpha]P$ by $[\alpha](N)P$.
4. A ρ -redex is a $\lambda\mu$ -term of the form $[\beta]\mu\alpha.M$ and we call $M[\alpha := \beta]$ its contractum. The $\lambda\mu$ -term $M[\alpha := \beta]$ is obtained from M by replacing every free occurrence of α by β .
5. A θ -redex is a $\lambda\mu$ -term of the form $\mu\alpha.[\alpha]M$ where $\alpha \notin \text{fv}(M)$ and we call M its contractum.
6. An ε -redex is a $\lambda\mu$ -term of the form $\mu\alpha.\mu\beta.M$ and we call $\mu\alpha.M_\beta$ its contractum.

The six reductions presented in Definition 2.6 can be justified by the cut elimination rules in natural deduction. This is why we will have type preservation property in the calculus (Theorem 2.10), which is also known as subject reduction property.

Remark 2.7.

1. The intuitive meaning of reducing $(\mu\alpha.M)N$ 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 with the symbol $[\alpha]$.
2. The intuitive meaning of reducing $(N)\mu\alpha.M$ to $\mu\alpha.M[\alpha :=_l N]$ is that the function N having $\mu\alpha.M$ as argument becomes the functional part of the applications $(N)P$, where P is a subterm of M named with the symbol $[\alpha]$.

Definition 2.8. (Reduction and normalization). Let $\mathcal{R} \subseteq \{\beta, \mu, \mu', \rho, \theta, \varepsilon\}$.

1. Let M and M' be two $\lambda\mu$ -terms. We write $M \rightarrow_{\mathcal{R}} M'$, if M' is obtained from M by replacing an r -redex in M , where $r \in \mathcal{R}$, by its contractum. The reductions (on the redexes) take the following forms (the θ -redex has an additional condition).

$$\begin{aligned}
 (\lambda x.M)N &\rightarrow_{\beta} M[x := N] \\
 (\mu\alpha.M)N &\rightarrow_{\mu} \mu\alpha.M[\alpha :=_r N] \\
 (N)\mu\alpha.M &\rightarrow_{\mu'} \mu\alpha.M[\alpha :=_l N] \\
 [\beta]\mu\alpha.M &\rightarrow_{\rho} M[\alpha := \beta] \\
 \mu\alpha.[\alpha]M &\rightarrow_{\theta} M \quad \text{if } \alpha \notin \text{fv}(M) \\
 \mu\alpha.\mu\beta.M &\rightarrow_{\varepsilon} \mu\alpha.M_{\beta}
 \end{aligned}$$

2. We denote by $\twoheadrightarrow_{\mathcal{R}}$ the reflexive, transitive closure of $\rightarrow_{\mathcal{R}}$. I.e., $M \twoheadrightarrow_{\mathcal{R}} M'$ iff $M \rightarrow_{\mathcal{R}} M_1 \rightarrow_{\mathcal{R}} M_2 \rightarrow \dots \rightarrow_{\mathcal{R}} M_k = M'$. Finally, we write $M \xrightarrow{n}_{\mathcal{R}} M'$ iff $M \rightarrow_{\mathcal{R}} M_1 \rightarrow_{\mathcal{R}} M_2 \rightarrow_{\mathcal{R}} \dots \rightarrow_{\mathcal{R}} M_n = M'$ and $M \xrightarrow{+}_{\mathcal{R}} M'$ iff there is an $n > 0$ such that $M \xrightarrow{n}_{\mathcal{R}} M'$.
3. We denote by $\mathcal{NF}_{\mathcal{R}}$ the set of all $\lambda\mu$ -terms in \mathcal{R} -normal form, i.e., $\lambda\mu$ -terms that do not contain an r -redex for any $r \in \mathcal{R}$.
4. A $\lambda\mu$ -term M is said to be \mathcal{R} -weakly normalizable if there exists $M' \in \mathcal{NF}_{\mathcal{R}}$ such that $M \twoheadrightarrow_{\mathcal{R}} M'$. We denote by $\mathcal{WN}_{\mathcal{R}}$ the set of \mathcal{R} -weakly normalizable terms.
5. A $\lambda\mu$ -term M is said to be \mathcal{R} -strongly normalizable, if there exists no infinite \mathcal{R} -reduction paths starting from M . That is, any possible sequence of reductions eventually leads to a normal term. We denote by $\mathcal{SN}_{\mathcal{R}}$ the set of \mathcal{R} -strongly normalizable terms.

Remark 2.9. When talking about reductions, it is customary to define residuals, that is, the results of performing reductions in a term. However, in Section 6, when we examine the outcome of a reduction starting from a term N and leading to N' , we are not interested in knowing about the residuals of a given subterm of N occurring in N' (which is a one-to-many correspondence). Rather, we consider redexes of N' and we are interested in their possible origin in N , that is, we trace back the reduction path, and this is a one-to-one correspondence. From the technical point of view, we do not need to formulate the notions of origin or residual to state the results of that section.

We list here the most important results that we are going to use in our paper. For the proofs of the next theorems, the interested reader is referred to Battyányi and Nour (2020), David and Nour (2005), Py (1998).

Theorem 2.10. *The $\beta\mu\mu'\rho\theta\varepsilon$ -reduction preserves types, that is, if $\Gamma \vdash M : A$ and $M \twoheadrightarrow_{\beta\mu\mu'\rho\theta\varepsilon} M'$, then $\Gamma \vdash M' : A$.*

Theorem 2.11. *The $\mu\mu'$ -reduction is strongly normalizing for the untyped $\lambda\mu$ -terms.*

Theorem 2.12. *The $\beta\mu\mu'$ -reduction is strongly normalizing for the typed $\lambda\mu$ -terms.*

In the sequel, let IH be an abbreviation for the phrase “the induction hypothesis”.

3. Natural Numbers

The aim of this section is to justify our choices regarding the reduction rules. The major benefit of the “proof as program” paradigm (also known as Curry–Howard isomorphism) is that the proof itself ensures that the program extracted is correct (Krivine 1993; Leivant 1983). For example, in intuitionistic second-order λ -calculus, to program functions on integers, it suffices to encode the proof of the totality of the function to extract the correct program. Since the only λ -terms

up to β -reduction that have an integer type are the Church numerals, to run the program one just has to apply it to a Church numeral and then reduce it to find the exact result. In classical logic the situation is different. Without μ' -reduction we can have closed $\lambda\mu$ -terms of integer type that are not Church numerals themselves (see Example 3.8). For data types, like integers, Parigot (1993) observed that the classical integers (the terms that encode proofs of integer type) may contain subterms which can be discarded by simple reduction rules without changing the type. This phenomenon comes from the fact that one does not need double negation elimination in order to prove that these terms are well typed. Hence, the idea is to add rules to tidy up the terms for classical integers. We must keep in mind, however, that we are not able to do the same thing in higher order logics, since, in that case, the μ' -rule does not preserve the types. We emphasize that this method does not mean that every classical proof can be transformed into an intuitionistic one but, at least for many of the data types in the simply typed lambda calculus, it is possible. In higher order logics there exists another solution: one can give computational content to the quantifiers in such a way that the μ' -rule preserves the types (Py 1998).

Henceforth, we adopt the strategy of adding certain rules to the calculus, most notably the μ' -rule. In this section, we prove that in the presence of all of our mentioned rules, the only $\lambda\mu$ -terms, regarding the de Groote syntax, that can be typed with integer type are the $\lambda\mu$ -terms that behave as Church numerals. In order to obtain the result, all the rules are necessary, that is, we consider $\beta\mu\mu'\rho\theta\varepsilon$ -reduction. The results of this section can be generalized to the various algebraic data types defined by Krivine in (1994). We omit their treatment in our paper.

Definition 3.1 (Natural numbers).

1. Let X be a variable type and $\mathcal{N} = X \rightarrow ((X \rightarrow X) \rightarrow X)$ be the integer type.
2. For every $n \in \mathbb{N}$ and $M, N \in \mathcal{T}$, we define by induction the $\lambda\mu$ -term $(M)^n N$ as follows: $(M)^0 N = N$ and, $\forall k \in \mathbb{N}$, $(M)^{k+1} N = (M)(M)^k N$; then $(M)^n N = \underbrace{(M) \dots (M)}_{n \text{ times}} N$.
3. For every $n \in \mathbb{N}$, we define the Church natural number (or Church numeral) as $\underline{n} = \lambda x.\lambda f.(f)^n x$.
4. Let $\underline{s} = \lambda n.\lambda x.\lambda f.(f)((n)x)f$. The λ -term \underline{s} is considered to be the successor function for Church numerals.

We remark that the successor function really behaves as the arithmetical operator on natural numbers: we can verify that, for all $n \in \mathbb{N}$, $(\underline{s})\underline{n} \rightarrow_{\beta} \underline{n + 1}$.

The following results are valid for λ -terms only. Later, we shall see examples of $\lambda\mu$ -terms of type \mathcal{N} (see Example 3.3).

Theorem 3.2. Let M be a λ -term. If $\vdash M : \mathcal{N}$, then, $\exists n \in \mathbb{N}$, $M \rightarrow_{\beta} \underline{n}$.

Proof. See Krivine (1993). □

With the introduction of the μ' -rule, we have lost confluence. It is easy to construct terms leading to different normal forms. More interestingly, we are able to define a $\lambda\mu$ -term of integer type that can evaluate to different integers. This permits us the coding of algorithms that can yield multiple possible results. The following example shows how a typed $\lambda\mu$ -term can be reduced to two different Church numerals. This phenomenon is not a shortcoming of the calculus. On the contrary, it even allows us the construction of programs that generate a set of solutions for a given problem.

Example 3.3. Let $n, m \in \mathbb{N}$ and $M = \mu\alpha.[\alpha](\mu\gamma.[\alpha]n)\mu\beta.[\alpha]m$. We have

- $\alpha : \neg\mathcal{N}, \beta : \neg\mathcal{N} \vdash m : \mathcal{N}$ and $\alpha : \neg\mathcal{N}, \gamma : \neg(\mathcal{N} \rightarrow \mathcal{N}) \vdash n : \mathcal{N}$, then $\alpha : \neg\mathcal{N} \vdash \mu\beta.[\alpha]m : \mathcal{N}$ and $\alpha : \neg\mathcal{N} \vdash \mu\gamma.[\alpha]n : \mathcal{N} \rightarrow \mathcal{N}$, thus $\alpha : \neg\mathcal{N} \vdash (\mu\gamma.[\alpha]n)\mu\beta.[\alpha]m : \mathcal{N}$, therefore $\vdash M : \mathcal{N}$,
- $M \rightarrow_{\mu} \mu\alpha.[\alpha]\mu\gamma.[\alpha]n \rightarrow_{\rho} \mu\alpha.[\alpha]n \rightarrow_{\theta} n$,
- $M \rightarrow_{\mu'} \mu\alpha.[\alpha]\mu\beta.[\alpha]m \rightarrow_{\rho} \mu\alpha.[\alpha]m \rightarrow_{\theta} m$.

Lemmas 3.4 and 3.5 are technical and indispensable for obtaining the characterization of $\lambda\mu$ -terms of type \mathcal{N} .

Lemma 3.4. Let $\Gamma = x : X, f : X \rightarrow X, \alpha : \neg X$.

1. There is no $\lambda\mu$ -term M such that $M = (M_1)M_2 \in \mathcal{NF}_{\beta\mu\mu'\rho\theta\varepsilon}$ for any terms M_1, M_2 and $\Gamma \vdash M : A \rightarrow B$.
2. There is no $\lambda\mu$ -term M such that $M = (M_1)M_2 \in \mathcal{NF}_{\beta\mu\mu'\rho\theta\varepsilon}$ for any terms M_1, M_2 and $\Gamma \vdash M : \perp$.

Proof.

1. By induction on M . If $\exists M = (M_1)M_2 \in \mathcal{NF}_{\beta\mu\mu'\rho\theta\varepsilon}$ and $\Gamma \vdash M : A \rightarrow B$, then, $\Gamma \vdash M_1 : C \rightarrow (A \rightarrow B)$ for some C . Since M is normal and, by inspecting the typing rules, $M_1 \neq x, M_1 \neq f, M_1 \neq [\alpha]M'_1, M_1 \neq \mu\beta.M'_1$ and $M_1 \neq \lambda y.M'_1$. We deduce that $M_1 = (M'_1)M'_2$ which is impossible by IH.
2. If $\exists M = (M_1)M_2 \in \mathcal{NF}_{\beta\mu\mu'\rho\theta\varepsilon}$ and $\Gamma \vdash M : \perp$, then, $\Gamma \vdash M_1 : B \rightarrow \perp$. Since M is normal and, by inspecting the typing rules, $M_1 \neq x, M_1 \neq f, M_1 \neq [\alpha]M'_1, M_1 \neq \mu\beta.M'_1$ and $M_1 \neq \lambda y.M'_1$. We deduce that $M_1 = (M'_1)M'_2$, which is impossible by 1. □

Lemma 3.5. Let $\Gamma = x : X, f : X \rightarrow X, \alpha : \neg X$. If $M \in \mathcal{NF}_{\beta\mu\mu'\rho\theta\varepsilon}$, $M \neq \mu\beta.M'$ and $\Gamma \vdash M : X$, then $\exists n \in \mathbb{N}, M = (f)^n x$.

Proof. By induction on M .

- Because of the typing rules and by our assumption, $M \neq [\alpha]M', M \neq \mu\alpha.M'$ and $M \neq \lambda y.M'$.
- If $M = x$, then $M = (f)^0 x$.
- If $M = (M_1)M_2$, then $\Gamma \vdash M_1 : A \rightarrow X$ and $\Gamma \vdash M_2 : A$. Since M_1 is normal and, if we take into account the typing rules, we find that $M_1 \neq [\alpha]M'_1, M_1 \neq \mu\gamma.M'_1$ and $M_1 \neq \lambda y.M'_1$. By Lemma 3.4, we have $M_1 \neq (M'_1)M'_2$. If M_1 is a λ -variable, then $M_1 = f$ and $A = X$. Since $\Gamma \vdash M_2 : X$ and, by the fact that M is normal, $M_2 \neq \mu\gamma.M'_2$, then, by IH, $\exists n \in \mathbb{N}, M_2 = (f)^n x$, thus $M = (f)^{n+1} x$. □

We are now in a position to prove the main result in this section.

Theorem 3.6. If $M \in \mathcal{NF}_{\beta\mu\mu'\rho\theta\varepsilon}$ and $x : X, f : X \rightarrow X \vdash M : X$, then $\exists n \in \mathbb{N}, M = (f)^n x$.

Proof. If $M = \mu\alpha.M'$, then $\Gamma = x : X, f : X \rightarrow X, \alpha : \neg X \vdash M' : \perp$. We prove by examining the form of the $\lambda\mu$ -term M' that $M \neq \mu\alpha.M'$.

- Since M is normal, and because of the typing rules, we have $M' \neq x, M' \neq f, M' \neq \mu\beta.M''$ and $M' \neq \lambda y.M''$.

- If $M' = [\alpha]M''$, then $M'' \neq \mu\beta.N$ and $\Gamma \vdash M'' : X$. By Lemma 3.5, $\exists n \in \mathbb{N}$, $M'' = (f)^n x$ and $M = \mu\alpha.[\alpha](f)^n x$ which is a θ -redex. Contradiction.
- If $M' = (M'_1)M'_2$, then we obtain a contradiction by Lemma 3.4.

We deduce that $M \neq \mu\alpha.M'$, and we apply Lemma 3.5. □

Hence, we obtain the following characterization of $\lambda\mu$ -terms of type \mathcal{N} .

Corollary 3.7. Let x, f be two different λ -variables. If $\vdash M : \mathcal{N}$ and $((M)x)f \in \mathcal{WN}_{\beta\mu\mu'\rho\theta\varepsilon}$, then $\exists n \in \mathbb{N}$, $((M)x)f \rightarrow_{\beta\mu\mu'\rho\theta\varepsilon} (f)^n x$.

Proof. Let $N \in \mathcal{NF}_{\beta\mu\mu'\rho\theta\varepsilon}$ such that $((M)x)f \rightarrow_{\beta\mu\mu'\rho\theta\varepsilon} N$. We have $x : X, f : X \rightarrow X \vdash ((M)x)f : X$, then, by Theorem 2.10, $x : X, f : X \rightarrow X \vdash N : X$, and, by Theorem 3.6, $\exists n \in \mathbb{N}$, $N = (f)^n x$. □

In the light of our subsequent results, the assumption $((M)x)f \in \mathcal{WN}_{\beta\mu\mu'\rho\theta\varepsilon}$ can in fact be omitted in the above corollary. Namely, we will show in the subsequent sections that $\beta\mu\mu'\rho\theta\varepsilon$ -reduction is weakly normalizing. Hence, if we wish to obtain the corresponding Church numeral, it is enough to apply M to the numeral $\underline{0}$ and to the successor \underline{s} , thus one arrives at the result $(\underline{s})^n \underline{0}$, which reduces to \underline{n} .

The following example presents a $\lambda\mu$ -term of integer type for which we need all reduction rules to obtain its value.

Example 3.8. Let $M = \lambda x. \mu\alpha. [\alpha] \lambda g. (g) \mu\beta. \mu\gamma. [\alpha] \lambda f. (f)x$.

- Let $x : X, f : X \rightarrow X, g : X \rightarrow X, \alpha : \neg((X \rightarrow X) \rightarrow X), \beta : \neg X$ and $\gamma : \neg \perp$, then we have the typing relation $\vdash M : \mathcal{N}$.
- In the rest of the paper, in order to make the reductions more readable, at each step we underline the successive redexes. We have

$$\begin{aligned}
 ((M)x)f &= ((\lambda x. \mu\alpha. [\alpha] \lambda g. (g) \mu\beta. \mu\gamma. [\alpha] \lambda f. (f)x)x)f \\
 &\rightarrow_{\beta} \underline{\mu\alpha. [\alpha] \lambda g. (g) \mu\beta. \mu\gamma. [\alpha] \lambda f. (f)x} f \\
 &\rightarrow_{\mu} \mu\alpha. [\alpha] \underline{\lambda g. (g) \mu\beta. \mu\gamma. [\alpha] \lambda f. (f)x} f \\
 &\rightarrow_{\beta} \mu\alpha. [\alpha] \underline{\lambda g. (g) \mu\beta. \mu\gamma. [\alpha] (f)x} f \\
 &\rightarrow_{\beta} \mu\alpha. [\alpha] (f) \underline{\mu\beta. \mu\gamma. [\alpha] (f)x} \\
 &\rightarrow_{\varepsilon} \mu\alpha. [\alpha] (f) \underline{\mu\beta. [\alpha] (f)x} \\
 &\rightarrow_{\mu'} \mu\alpha. [\alpha] \underline{\mu\beta. [\alpha] (f)x} \\
 &\rightarrow_{\rho} \underline{\mu\alpha. [\alpha] (f)x} \\
 &\rightarrow_{\theta} (f)x.
 \end{aligned}$$

Remark 3.9. We recover the Church numerals directly from M if we make use of the ν -rule, which was introduced by Parigot (1993). The rule can be presented as follows:

$$\mu\alpha.M \rightarrow_{\nu} \lambda x. \mu\alpha. M[\alpha :=_r x] \quad \text{if } \alpha \text{ is of type } A \rightarrow B.$$

If we consider again Example 3.8, we obtain the following reduction sequence.

$$\begin{aligned}
 M &= \lambda x. \mu\alpha. [\alpha] \lambda g. (g) \mu\beta. \mu\gamma. [\alpha] \lambda f. (f)x \\
 &\rightarrow_{\nu} \lambda x. \lambda h. \mu\alpha. [\alpha] \underline{\lambda g. (g) \mu\beta. \mu\gamma. [\alpha] \lambda f. (f)x} h
 \end{aligned}$$

$$\begin{aligned}
 &\rightarrow_{\beta} \lambda x. \lambda h. \mu \alpha. [\alpha] (\lambda g. (g) \mu \beta. \mu \gamma. [\alpha] (h) x) h \\
 &\rightarrow_{\beta} \lambda x. \lambda h. \mu \alpha. [\alpha] (h) \mu \beta. \mu \gamma. [\alpha] (h) x \\
 &\rightarrow_{\varepsilon} \lambda x. \lambda h. \mu \alpha. [\alpha] (h) \mu \beta. [\alpha] (h) x \\
 &\rightarrow_{\mu'} \lambda x. \lambda h. \mu \alpha. [\alpha] \mu \beta. [\alpha] (h) x \\
 &\rightarrow_{\rho} \lambda x. \lambda h. \mu \alpha. [\alpha] (h) x \\
 &\rightarrow_{\theta} \lambda x. \lambda h. (h) x \\
 &= \perp.
 \end{aligned}$$

We prefer not to use the ν -rule, however, since it contains a constraint on the type of a bound variable. If we considered the untyped calculus together with the ν -rule, then the calculus would diverge for all $\lambda\mu$ -terms containing a μ . We prefer a calculus where reduction does not depend on the types of terms, hence we omit the ν -rule. In addition, the ν -rule is not strictly necessary for our main goal in this section, which is the demonstration that the introduction of certain rules suffices for finding the value of a term of integer type. We need $\beta\mu\mu'\rho\theta\varepsilon$ -reduction for this purpose, this is the statement of Corollary 3.7.

4. The Parigot-Style $\lambda\mu$ -Calculus

Prior to turning to our main task and detailing the normalization results for $\beta\mu\mu'\rho\theta\varepsilon$ -reduction, we make a little detour and examine the original version of the $\lambda\mu$ -calculus due to Parigot (1992). We will refer to the calculus discussed in this section as the Parigot-style $\lambda\mu$ -calculus as opposed to the one appearing in Definition 2.1, which was developed by de Groote (1998). The term formation rules of Parigot’s calculus differ from the ones discussed so far in the following manner: in Parigot’s calculus the forms of $\lambda\mu$ -terms are restricted in such a way that μ -abstractions and applications with μ -variables are tied together. This is achieved by distinguishing *named* and *unnamed* $\lambda\mu$ -terms. Named $\lambda\mu$ -terms are formulated from unnamed terms by an application of a μ -variable. Conversely, μ -abstracting over named terms returns unnamed terms. This is formalized in the definition below.

$$\begin{aligned}
 \mathcal{T} &:= \mathcal{T}_u \cup \mathcal{T}_n \\
 \mathcal{T}_u &:= \mathcal{V}_\lambda \mid \mu \mathcal{V}_\mu. \mathcal{T}_n \mid \lambda \mathcal{V}_\lambda. \mathcal{T}_u \mid (\mathcal{T}_u) \mathcal{T}_u \\
 \mathcal{T}_n &:= [\mathcal{V}_\mu] \mathcal{T}_u
 \end{aligned}$$

The typing rules are the same as in Definition 2.3, except that the last two rules are replaced with the following single rule:

$$\frac{\Gamma, \alpha : \neg A, \beta : \neg B \vdash M : B}{\Gamma, \beta : \neg B \vdash \mu \alpha. [\beta] M : A}$$

Concerning the reduction rules, due to the syntax of $\lambda\mu$ -terms, the ε -rule cannot be defined here, hence it is omitted.

To better understand the difference between the Parigot and de Groote-style versions, we present an example that will be discussed in detail in the rest of the paragraph. We can check that the $\lambda\mu$ -term $T = \lambda x. \mu \alpha. x$ is of type $\perp \rightarrow X$. This term can be formulated in the de Groote syntax only, since, following the $\mu\alpha$, there is no $[\beta]$. The $\lambda\mu$ -term T behaves quite interestingly. We can verify that, when we apply T to an arbitrarily long finite sequence of $\lambda\mu$ -terms, after some steps

of μ -reduction we obtain $\mu\alpha.P$, provided P was the first element of the list. Roughly speaking, T takes the first element of a list and erases the others. The second author has proved with Saber (Nour and Saber 2009) that every $\lambda\mu$ -term of type $\perp \rightarrow X$ enjoys the same property. The simplest proof of the formula $\perp \rightarrow X$ in the Parigot-style calculus turns out to be $T' = \lambda x.\mu\alpha.[\varphi]x$. The free variable φ is of type $\neg\perp$, and it is quite embarrassing to acknowledge that we need to resort to a free variable of type $\neg\perp$ in order to construct a term of type $\perp \rightarrow X$, which is a classical tautology. Anyway, the $\lambda\mu$ -term T' behaves in the same way as T .

In what follows, we prove that Parigot's $\lambda\mu$ -calculus with $\rho\theta$ -reduction possesses the strong normalization property. First of all, we show that ρ - and θ -reduction can be strongly postponed with respect to β -, μ -, and μ' -reduction.

Definition 4.1 (Strong postponement). *Let \rightarrow_1 and \rightarrow_2 be two reduction relations. The reduction \rightarrow_2 can be strongly postponed with respect to \rightarrow_1 , if, for every $M, N, P \in \mathcal{T}$ such that $M \rightarrow_2 P \rightarrow_1^+ N$, we have $M \rightarrow_1^+ Q \rightarrow_2 N$ for some Q .*

Our first aim is to demonstrate the following theorem.

Theorem 4.2. *In the Parigot-style calculus, $\rho\theta$ -reduction can be strongly postponed with respect to $\mu\mu'\beta$.*

We prove the theorem in two steps. First, we establish that θ -reduction can be strongly postponed with respect to all the other rules, namely, β , μ , μ' , ρ . We then show that the ρ -rule can be strongly postponed with respect to μ , μ' , and β .

First of all, we define two reductions that are special cases of μ - and μ' -reductions. These new rules appear in the permutation lemmas that we are going to prove and are easy to handle.

Definition 4.3 (μ_0 and μ'_0 -reduction). *A μ_0 -redex (resp. μ'_0 -redex) is a $\lambda\mu$ -term of the form $(\mu\alpha.M)N$ (resp. $(N)\mu\alpha.M$) where α occurs at most once in M , and we call $\mu\alpha.M[\alpha :=_r N]$ (resp. $\mu\alpha.M[\alpha :=_l N]$) its contractum. We write $M \rightarrow_{\mu_0} M'$ (resp. $M \rightarrow_{\mu'_0} M'$), if M' is obtained from M by replacing a μ_0 -redex (resp. μ'_0 -redex) in M by its contractum. As usual, we define the reductions $\twoheadrightarrow_{\mu_0}$, $\twoheadrightarrow_{\mu'_0}$, $\twoheadrightarrow^n_{\mu_0}$, $\twoheadrightarrow^n_{\mu'_0}$, $\twoheadrightarrow^+_{\mu_0}$ and $\twoheadrightarrow^+_{\mu'_0}$.*

4.1 The case of θ -reduction

We show that, in the Parigot-style $\lambda\mu$ -calculus, θ -reduction can be strongly postponed with respect to $\beta\mu\mu'\rho$ -reduction. We prove for β -, μ -, μ' -, and ρ -reductions, one by one, that the execution of a θ -reduction followed by either of the mentioned reductions can be replaced by a reduction sequence where the respective reductions precede θ -reduction (Lemmas 4.4, 4.5, 4.6, 4.7). Additionally, new μ_0 - or μ'_0 -redexes may be created, but these redexes can be executed before the θ -redex. We can generalize the results on swapping θ -reduction with $\beta\mu\mu'\rho$ -reductions to reduction sequences by reasoning by induction on the length of the reduction sequences. This is stated in Theorem 4.9.

The following lemma shows how θ -reduction relates to the β -rule. In the next lemma, we observe the usefulness of the new μ_0 -reduction rule.

Lemma 4.4. *Let $M \rightarrow_{\theta} P \rightarrow_{\beta} N$. Then either we have a Q such that $M \rightarrow_{\beta} Q \twoheadrightarrow_{\theta} N$, or there are R, Q for which $M \rightarrow_{\mu_0} R \rightarrow_{\beta} Q \rightarrow_{\theta} N$.*

Proof. By induction on M .

- The cases $M = [\alpha]M_1$, $M = \mu\alpha.M_1$ and $M = \lambda x.M_1$ are straightforward.
- Assume $M = (M_1)M_2$.
 - If $M_1 \rightarrow_\theta M'_1$, the only nontrivial case is $M'_1 = \lambda x.M_3$ and $M_1 = \mu\alpha.[\alpha]M'_1$ with $\alpha \notin \text{fv}(M'_1)$. Now we obtain $M = (\mu\alpha.[\alpha]\lambda x.M_3)M_2 \rightarrow_{\mu_0} \mu\alpha.[\alpha](\lambda x.M_3)M_2 \rightarrow_\beta \mu\alpha.[\alpha]M_3[x := M_2] \rightarrow_\theta M_3[x := M_2]$.
 - If $M_2 \rightarrow_\theta M'_2$, the only interesting case is $M_1 = \lambda x.M_3$, $N = M_3[x := M'_2]$. Then, with $Q = M_3[x := M_2]$, we have $M \rightarrow_\beta Q \rightarrow_\theta N$. □

The next lemma shows how θ -reduction commutes with μ -reduction. We notice the appearance of μ_0 -reduction.

Lemma 4.5.

1. Let $M \rightarrow_\theta P \rightarrow_\mu N$. Then either we have a Q such that $M \rightarrow_\mu Q \rightarrow_\theta N$ or there are R, Q such that $M \rightarrow_\mu R \rightarrow_\mu Q \rightarrow_\theta N$.
2. Let $M \rightarrow_\theta P \rightarrow_{\mu_0} N$. Then either $M \rightarrow_{\mu_0} Q \rightarrow_\theta N$ for some Q or there are R, Q such that $M \rightarrow_{\mu_0} R \rightarrow_{\mu_0} Q \rightarrow_\theta N$.
3. Let $M \rightarrow_\theta^n P \rightarrow_{\mu_0} N$. Then there is a Q such that $M \rightarrow_{\mu_0} Q \rightarrow_\theta^n N$.

Proof.

1. By induction on M .

- The cases $M = [\alpha]M_1$ and $M = \lambda x.M_1$ are straightforward. Regarding when $M = \mu\alpha.M_1$, if $M_1 \rightarrow_\theta M'_1$, then the IH applies. Otherwise, $M_1 = [\alpha]M_2$ and $M \rightarrow_\theta M_2 = P \rightarrow_\beta N$. Then $M \rightarrow_\beta \mu\alpha.[\alpha]N \rightarrow_\theta N$.
- Assume $M = (M_1)M_2$.
 - If $M_1 = \mu\alpha.[\alpha]M'_1 \rightarrow_\theta M'_1$, the only interesting case is $M'_1 = \mu\beta.M_3$. Since $\alpha \notin M'_1$, we have $M = (\mu\alpha.[\alpha]\mu\beta.M_3)M_2 \rightarrow_{\mu_0} \mu\alpha.[\alpha](\mu\beta.M_3)M_2 \rightarrow_\mu \mu\alpha.[\alpha]\mu\beta.M_3[\beta :=_r M_2] \rightarrow_\theta \mu\beta.M_3[\beta :=_r M_2]$.
 - If $M_2 = \mu\alpha.[\alpha]M'_2 \rightarrow_\theta M'_2$, the only interesting case is $M_1 = \mu\beta.M_3$, $N = \mu\beta.M_3[\beta :=_r M'_2]$, where, with $Q = \mu\beta.M_3[\beta :=_r M_2]$, we obtain $M \rightarrow_\mu Q \rightarrow_\theta N$.

3. The proof follows the pattern of point 1 of this lemma with the necessary changes implemented.
4. Applying case 2 of this lemma and proceeding by induction on n . □

The following lemma illustrates how θ -reduction commutes with μ' - and μ'_0 -reduction.

Lemma 4.6.

1. Let $M \rightarrow_\theta P \rightarrow_{\mu'} N$. Then either $M \rightarrow_{\mu'} Q \rightarrow_\theta N$ for some Q or there are R, Q such that $M \rightarrow_{\mu'_0} R \rightarrow_{\mu'} Q \rightarrow_\theta N$.
2. Let $M \rightarrow_\theta P \rightarrow_{\mu'_0} N$. Then either we have $M \rightarrow_{\mu'_0} Q \rightarrow_\theta N$ for some Q or there are R, Q such that $M \rightarrow_{\mu'_0} R \rightarrow_{\mu'_0} Q \rightarrow_\theta N$.
3. Let $M \rightarrow_\theta^n P \rightarrow_{\mu'_0} N$. Then there is a Q such that $M \rightarrow_{\mu'_0} Q \rightarrow_\theta^n N$.

Proof. Similar to Lemma 4.5. □

The lemma below demonstrates that θ -reduction can be strongly postponed with respect to ρ -reduction.

Lemma 4.7. Let $M \rightarrow_{\theta} P \rightarrow_{\rho} N$. Then we have a Q such that $M \rightarrow_{\rho} Q \rightarrow_{\theta} N$.

Proof. An obvious induction on M . □

We are now in a position to prove the next theorem, which describes how θ -reduction commutes with all the other rules.

Theorem 4.8. Let $M \rightarrow_{\theta} P \rightarrow_{\beta\mu\mu'\rho} N$. Then there exists Q such that $M \xrightarrow{+}_{\beta\mu\mu'\rho} Q \rightarrow_{\theta} N$.

Proof. Let $M \xrightarrow{n}_{\theta} P \rightarrow_{\beta\mu\mu'\rho} N$. The proof proceeds by induction on n and by applying the previous lemmas. Let us suppose first that $M \xrightarrow{n}_{\theta} P \rightarrow_{\mu} N$.

- $n = 0$: the result is immediate.
- $n = m + 1 > 0$: let $M \xrightarrow{m}_{\theta} K \rightarrow_{\theta} P$ for some K . By Lemma 4.5, either $K \rightarrow_{\mu} R \rightarrow_{\theta} N$ or $K \xrightarrow{\mu_0} R \rightarrow_{\mu} L \rightarrow_{\theta} N$. In the former case, the IH applies to $M \xrightarrow{m}_{\theta} K \rightarrow_{\mu} R$. In the latter case, by Lemma 4.5, there exists an S such that $M \xrightarrow{\mu_0} S \xrightarrow{m}_{\theta} R$, and we can apply the IH to $S \xrightarrow{m}_{\theta} R \rightarrow_{\mu} L$.

The cases $P \rightarrow_{\mu'} N$ and $P \rightarrow_{\beta} N$ are analogous. Finally, the case $P \rightarrow_{\rho} N$ is straightforward by Lemma 4.7. □

We have obtained the strong postponement of \rightarrow_{θ} with respect to $\rightarrow_{\beta\mu\mu'\rho}$. This is stated in the next theorem.

Theorem 4.9. Let $M \rightarrow_{\theta} P \xrightarrow{+}_{\beta\mu\mu'\rho} N$. Then there exists a Q for which $M \xrightarrow{+}_{\beta\mu\mu'\rho} Q \rightarrow_{\theta} N$.

Proof. By induction on the reduction sequence $P \xrightarrow{+}_{\beta\mu\mu'\rho} N$ using Theorem 4.8. □

Remark 4.10. We notice that the above proofs, hence Theorem 4.9, are also valid in the de Groote-style calculus. By contrast, the ρ -rule exhibits a different behavior in de Groote’s calculus. In what follows, we show that ρ -reduction can be strongly postponed with respect to $\beta\mu\mu'$ -reduction in the simply typed Parigot calculus, and we demonstrate by giving counterexamples that this is not valid in the de Groote-style syntax.

4.2 The case of ρ -reduction

Now we show that, in the simply typed Parigot-style $\lambda\mu$ -calculus, ρ -reduction can be strongly postponed with respect to $\beta\mu\mu'$ -reduction. The proofs of this subsection can also be found in Py (1998). The presentation there, however, may differ in some places from ours. For the sake of completeness, we outline the proofs below. We prove that ρ -reduction commutes with β -, μ - and μ' -reduction. This results in a reduction sequence starting with β -reduction (or with a μ - or μ' -reduction, respectively) and ending in ρ -reduction (Lemmas 4.12 and 4.14). The subsection culminates in Theorem 4.16, which can be obtained from the above-mentioned results by induction on the length of the reduction sequences.

The following lemma will be used in the proof of Lemma 4.12.

Lemma 4.11. Let $M \rightarrow_{\rho} M'$, then $M[x := N] \rightarrow_{\rho} M'[x := N]$.

Proof. By induction on M . □

The next lemma proves postponement of ρ -reduction with respect to β -reduction.

Lemma 4.12.

1. Let $M \rightarrow_{\rho} P \rightarrow_{\beta} N$. Then there exists Q such that $M \rightarrow_{\beta} Q \rightarrow_{\rho} N$.
2. Let $M \rightarrow_{\rho} P \rightarrow_{\beta} N$. Then there exists Q for which $M \rightarrow_{\beta} Q \rightarrow_{\rho} N$.

Proof.

1. By induction on M .

- The cases $M = \mu\alpha.M_1$ and $M = \lambda x.M_1$ are straightforward.
- Assume $M = [\alpha]M_1$. If $M' = [\alpha]M'_1$ and $M_1 \rightarrow_{\rho} M'_1$, the IH applies. Otherwise $M = [\alpha]M_1 = [\alpha]\mu\beta M_2 \rightarrow_{\rho} M_2[\beta := \alpha] = P \rightarrow_{\beta} N$, which is also straightforward.
- Assume $M = (M_1)M_2$.
 - If $M_1 \rightarrow_{\rho} M'_1$, we argue by induction on M_1 .
 - * $M'_1 \rightarrow_{\beta} M''_1$: we apply IH.
 - * $M_2 \rightarrow_{\beta} M'_2$: straightforward.
 - * $M_1 = \lambda x.M_3$: $M'_1 = \lambda x.M'_3$ for some $M_3 \rightarrow_{\rho} M'_3$ and $N = M'_3[x := M_2]$. Then, by Lemma 4.11, we obtain $M \rightarrow_{\beta} M_3[x := M_2] \rightarrow_{\rho} M'_3[x := M_2] = N$.
 - * $M_1 = [\alpha]M_3$: impossible by the term formation rules.
 - * $M_1 = \mu\alpha.M_3$ or $M_1 = (M_3)M_4$: we apply IH.
 - Assume $M_2 \rightarrow_{\rho} M'_2$.
 - * If $M_1 \rightarrow_{\beta} M'_1$ or $M'_2 \rightarrow_{\beta} M''_2$, we conclude by applying IH.
 - * $M_2 = \lambda x.M_3$ or $M_2 = \mu\alpha.M_3$: IH applies.
 - * $M_2 = [\alpha]M_3$: impossible by the term formation rules.

2. By the previous point of this lemma. □

The following lemma is useful for proving Lemma 4.14.

Lemma 4.13. Let $M \rightarrow_{\rho} M'$ and assume $\alpha \notin \text{fv}(N)$. Then either $M[\alpha :=_r N] \rightarrow_{\rho} M'[\alpha :=_r N]$ (resp. $M[\alpha :=_l N] \rightarrow_{\rho} M'[\alpha :=_l N]$) or $M[\alpha :=_r N] \rightarrow_{\mu} P \rightarrow_{\rho} M'[\alpha :=_r N]$ (resp. $M[\alpha :=_l N] \rightarrow_{\mu'} P \rightarrow_{\rho} M'[\alpha :=_l N]$) for some P .

Proof. By induction on M . The only interesting case is $M = [\gamma]\mu\beta.M_1$.

- If $\gamma \neq \alpha$, then $M[\alpha :=_r N] = [\gamma]\mu\beta.M_1[\alpha :=_r N] \rightarrow_{\rho} M_1[\beta := \gamma][\alpha :=_r N]$.
- If $\gamma = \alpha$, then $M[\alpha :=_r N] = [\alpha](\mu\beta.M_1[\alpha :=_r N])N$
 $\rightarrow_{\mu} [\alpha]\mu\beta.M_1[\alpha :=_r N][\beta :=_r N] \rightarrow_{\rho} M_1[\alpha :=_r N][\beta :=_r N][\beta := \alpha]$
 $= M_1[\beta := \alpha][\alpha :=_r N]$. □

The following lemma proves postponement of ρ -reduction with respect to μ - and μ' -reductions.

Lemma 4.14.

1. Let $M \rightarrow_\rho P \rightarrow_\mu N$. Then there exists Q for which $M \rightarrow_\mu^+ Q \rightarrow_\rho N$.
2. Let $M \rightarrow_\rho P \rightarrow_{\mu'} N$. Then there exists Q for which $M \rightarrow_{\mu'}^+ Q \rightarrow_\rho N$.
3. Let $M \rightarrow_\rho P \rightarrow_\mu N$ (resp. $M \rightarrow_\rho P \rightarrow_{\mu'} N$). Then there exists Q such that $M \rightarrow_\mu^+ Q \rightarrow_\rho N$ (resp. $M \rightarrow_{\mu'}^+ Q \rightarrow_\rho N$).

Proof. We only detail some of the more interesting cases of the first point.

1. By induction on M .
 - The cases $M = \mu\alpha.M_1$ and $M = \lambda x.M_1$ are straightforward.
 - Assume $M = [\alpha]M_1$: if $M_1 \rightarrow_\rho M'_1$, we obtain the result by IH. Otherwise, $M = [\alpha]\mu\beta.M_2 \rightarrow_\rho M_2[\beta := \alpha] \rightarrow_\mu^R N$. Then $M \rightarrow_\mu^{R'} [\alpha]\mu\beta.M_3 \rightarrow_\rho M_3[\beta := \alpha] = N$ for some M_3 , where $R = R'[\beta := \alpha]$.
 - Assume $M = (M_1)M_2$.
 - If $M_1 \rightarrow_\rho M'_1$, we distinguish the following cases.
 - * $M'_1 \rightarrow_\mu M''_1$: we apply IH.
 - * $M_2 \rightarrow_\mu M'_2$: by IH.
 - * $M_1 = \mu\gamma.M_3$ and $M_3 \rightarrow_\rho M'_3$ and $N = \mu\gamma.M'_3[\gamma :=_r M_2]$. Then, by Lemma 4.13, we have $M = (\mu\gamma.M_3)M_2 \rightarrow_\mu \mu\gamma.M_3[\gamma :=_r M_2] \rightarrow_\mu Q \rightarrow_\rho \mu\gamma.M'_3[\gamma :=_r M_2]$ for some Q .
 - * $M_1 = \lambda x.M_3$: the IH applies.
 - * $M_1 = [\alpha]M_3$: impossible by the term formation rules.
 - The case $M = (M_1)M_2$ and $M_2 \rightarrow_\rho M'_2$ is similar to the above one. The only additional point is $M_1 = \mu\alpha.M_3$ and $N = \mu\alpha.M_3[\alpha :=_r M'_2]$. Then, with $Q = \mu\alpha.M_3[\alpha :=_r M_2]$, $M \rightarrow_\mu Q \rightarrow_\rho N$ holds.
2. Analogous to the above case. The proof makes use of Lemma 4.13.
3. By Points 1 and 2 of this lemma.

□

Remark 4.15. We emphasize that Lemmas 4.12 and 4.14 rely heavily on the fact that, in Parigot’s calculus, a μ -application must always be preceded by a μ -abstraction. The lemmas mentioned are not valid in the de Groote-style $\lambda\mu$ -calculus which is shown by the following examples:

- Let $M = ([\alpha]\mu\beta.\lambda x.P)Q$. Then $M \rightarrow_\rho (\lambda x.P[\beta := \alpha])Q \rightarrow_\beta P[\beta := \alpha][x := Q]$, and we are not able to reverse the order of the β - and ρ -reductions.
- Let $M = ([\alpha]\mu\beta.\mu\gamma.P)Q$. Then $M \rightarrow_\rho (\mu\gamma.P[\beta := \alpha])Q \rightarrow_\mu \mu\gamma.P[\beta := \alpha][\gamma :=_r Q]$, and we are not able to reverse the order of the μ - and ρ -reductions.

This phenomenon gives rise to the counterexample falsifying the strong normalization property of the de Groote-style calculus. We will discuss this in Section 5.

Now we can prove the theorem below, which asserts that the ρ -rule can be strongly postponed with respect to $\beta\mu\mu'$ -reduction.

Theorem 4.16. *Let $M \rightarrow_{\rho} P \rightarrow_{\beta\mu\mu'}^{+} N$. Then there is a Q for which $M \rightarrow_{\beta\mu\mu'}^{+} Q \rightarrow_{\rho} N$.*

Proof. A consequence of Lemmas 4.12 and 4.14. □

Theorems 4.9 and 4.16 together yield Theorem 4.2.

4.3 Strong normalization

We are now in a position to state the main assertion of this section, which is Theorem 4.18.

Theorem 4.17. *The $\rho\theta$ -reduction is strongly normalizing.*

Proof. Assume $M \rightarrow_{\rho\theta} M'$. Then it is easy to observe that M' , as a word, consists of fewer symbols than M , which proves our assertion. □

We are now ready to prove our strong normalization result.

Theorem 4.18. *In the simply typed Parigot-style $\lambda\mu$ -calculus, $\beta\mu\mu'\rho\theta$ -reduction is strongly normalizing.*

Proof. It is known that de Groote’s simply typed $\lambda\mu$ -calculus with $\beta\mu\mu'$ -reduction is strongly normalizing (David and Nour 2005). Since de Groote’s calculus is an extension of the original Parigot-style calculus, we immediately have the strong normalization property for the simply typed Parigot calculus with $\beta\mu\mu'$ -reduction. Let $\eta(M)$ denote the longest $\beta\mu\mu'$ -reduction sequence starting from M . Let M be such that there is an infinite $\beta\mu\mu'\rho\theta$ -reduction sequence starting from M and $\eta(M)$ is minimal. By Theorem 4.17, the infinite reduction sequence starting from M must contain a $\beta\mu\mu'$ -reduction, hence there is an M' and M'' such that $M \rightarrow_{\rho\theta} M' \rightarrow_{\beta\mu\mu'} M''$ and M'' has an infinite $\beta\mu\mu'\rho\theta$ -reduction sequence. If we apply Theorem 4.9 successively to the reduction sequence $M \rightarrow_{\rho\theta} M' \rightarrow_{\beta\mu\mu'} M''$, then we obtain an N and P such that $M \rightarrow_{\rho} N \rightarrow_{\beta\mu\mu'}^{+} P \rightarrow_{\theta} M''$. Obviously, P has an infinite $\beta\mu\mu'\rho\theta$ -reduction sequence, since the reduction sequence $P \rightarrow_{\theta} M''$ can be continued with the infinite reduction sequence starting from M'' . We distinguish the following cases:

- $M \rightarrow_{\rho} N$ is empty: since $\eta(P) < \eta(M)$, this contradicts the assumption on M .
- $M \rightarrow_{\rho} N$ is not empty: by Theorem 4.16, there is a Q such that $M \rightarrow_{\beta\mu\mu'}^{+} Q \rightarrow_{\rho} P$, then $\eta(Q) < \eta(M)$. This is a contradiction again. □

5. The μ -Calculus

For the rest of the paper, we return to the de Groote-style syntax. With the creation of the $\lambda\mu$ -calculus, Parigot introduced a formal method to encode proofs in classical logic via terms of an appropriate calculus. The addition of the rules in connection with classical variables made it possible to represent the proof-theoretical machinery called reasoning by absurdity, which is very characteristic of classical reasoning. Parigot posed the question whether the rules corresponding to classical variables, that is, the rules without β , are normalizing. In this spirit, we devote this section to the calculus with rules other than β . The rationale behind our choice is the expectation that examining the proof-theoretical properties can be simpler without the β -rule, since we know that the β -rule in the untyped case is not strongly normalizing and we know, for example, from

Battyányi and Nour (2020) and David and Nour (2005) that the $\mu\mu'$ -calculus has the strong normalization property even in the untyped case. It looks quite tempting to try and prove the strong normalization property for the calculi obtained by adding more simplification rules like ρ -, ε -, and θ -rules to the $\mu\mu'$ -rules. The θ -rule actually does not cause any difficulty, since it is strongly normalizing and, moreover, we can always move a θ -redex to the end of a reduction sequence. However, contradicting our first guess, $\mu\mu'$ -reduction fails to be strongly normalizing with the simplification rules ρ or ε , however, innocent these rules may look. We demonstrate by examples that $\mu\mu'\rho$ - and $\mu\mu'\varepsilon$ -reductions are not strongly normalizing. We are able to prove, however, that weak normalization is preserved: in the second half of this section we show that $\mu\mu'\rho\varepsilon$ -reduction is weakly normalizing. We call the above calculus without β -rule the μ -calculus, that is, we consider $\mu\mu'\rho\varepsilon$ -reduction. In this respect, we talk about μ -normalization and μ -normal form if the reduction is clear from the context. Otherwise, we indicate explicitly the reductions we are dealing with. We could retain the structure of $\lambda\mu$ -terms by encoding a $\lambda\mu$ -term with a μ -term by inductively replacing the subterms of the form $\lambda x.M$ with $((l)x)M$, where l is a constant. This way any $\mu\mu'\rho\varepsilon$ -reduction of a $\lambda\mu$ -term would be simulated by an identical $\mu\mu'\rho\varepsilon$ -reduction of the appropriate μ -term. For the sake of simplicity, we abandon this option. Since only the typed β -reduction is strongly normalizing, in the last section, when we consider β -rule, we exclusively remain in the typed setting. Several results of this section, however, work in the untyped case as well. This is the reason why most of the present section will be concerned with the untyped calculus. Let us first define the μ -terms, where we have deliberately omitted λ -abstractions, since they do not play any role concerning the normalization properties of the calculus without β -rule.

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

5.1 The reductions $\mu\mu'\rho$ and $\mu\mu'\varepsilon$

It is already known that $\mu\mu'$ -reduction is strongly normalizing (Battyányi 2007; Battyányi and Nour 2020; David and Nour 2005). We demonstrate in this section that this property is lost when we add either ρ - or ε -reduction to these rules. This phenomenon is actually a peculiarity of the de Groote-style syntax. In Theorems 5.1 and 5.3, we present examples of μ -terms that are either not strongly or even not weakly normalizing. Both results appeared first in the thesis of the first author (Battyányi 2007).

Theorem 5.1. *The $\mu\mu'\rho$ -reduction in the typed μ -calculus is not strongly normalizing.*

Proof. Let $P = \mu\alpha.[\alpha][\alpha]x$, $Q = \mu\beta.P$ and $M = (Q)P$.

- We observe that the term M can be typed. Let $\alpha : \neg\perp$ and $\beta : \neg(\perp \rightarrow \perp)$, then we have the typing relations $x : \perp \vdash P : \perp$, $x : \perp \vdash Q : \perp \rightarrow \perp$ and $x : \perp \vdash M : \perp$.
- The following reduction steps show that there exists an infinite reduction sequence starting from M .

$$\begin{aligned} M &= (Q)\mu\alpha.[\alpha][\alpha]x \\ &\rightarrow_{\mu'} \mu\alpha.[\alpha](Q)[\alpha](Q)x \\ &\rightarrow_{\mu} \mu\alpha.[\alpha](Q)[\alpha]\mu\beta.P \\ &\rightarrow_{\rho} \mu\alpha.[\alpha](Q)P = \mu\alpha.[\alpha]M \\ &\dots \end{aligned}$$

□

Remark 5.2.

1. Note that the μ -term M cannot be written in Parigot syntax because we will not be able to write two successive $[\alpha]$ s.
2. We remark that, when θ -reduction is allowed, the example above returns the initial μ -term M .
3. We can normalize the μ -term M as follows : $M = (Q)P \rightarrow_{\mu} Q$.
4. We can use this example to find a $\lambda\mu$ -term of integer type which is not strongly normalizable. Just consider the $\lambda\mu$ -term $W = \lambda y. \lambda f. \mu \gamma. M[x := [\gamma]y]$.

Theorem 5.3. *The $\mu\mu'\varepsilon$ -reduction in the typed μ -calculus is not strongly normalizing.*

Proof. Let $K = \mu\alpha.[\alpha][\alpha]x, L = \mu\beta.[\beta](z)[\beta]y$ and $N = \mu\gamma.(L)K$.

- We first observe that the term N can be typed. Let $\alpha : \neg\perp, \beta : \neg(\perp \rightarrow \perp)$ and $\gamma : \neg\perp$. We have the typing relations $x : \perp \vdash K : \perp, y : \perp \rightarrow \perp, z : \perp \rightarrow (\perp \rightarrow \perp) \vdash L : \perp \rightarrow \perp$ and $x : \perp, y : \perp \rightarrow \perp, z : \perp \rightarrow (\perp \rightarrow \perp) \vdash N : \perp$.
- The following reductions show that there exists an infinite reduction sequence starting from N .

$$\begin{aligned}
 N &= \mu\gamma.(\mu\beta.[\beta](z)[\beta]y)K \\
 &\rightarrow_{\mu} \underline{\mu\gamma.\mu\beta.[\beta]((z)[\beta](y)K)K} \\
 &\rightarrow_{\varepsilon} \underline{\mu\gamma.((z)(y)K)K} \\
 &\rightarrow_{\mu'} \underline{\mu\gamma.\mu\alpha.[\alpha]((z)(y)K)[\alpha]((z)(y)K)x} \\
 &\rightarrow_{\varepsilon} \underline{\mu\gamma.((z)(y)K)((z)(y)K)x} \\
 &\rightarrow_{\mu'} \underline{\mu\gamma.((z)(y)K)((z)\mu\alpha.[\alpha](y)[\alpha](y)x)x} \\
 &\rightarrow_{\mu'} \underline{\mu\gamma.((z)(y)K)(\mu\alpha.[\alpha](z)(y)[\alpha](z)(y)x)x} \\
 &\rightarrow_{\mu} \underline{\mu\gamma.((z)(y)K)\mu\alpha.[\alpha]((z)(y)[\alpha]((z)(y)x)x)x} \\
 &\rightarrow_{\mu'} \underline{\mu\gamma.\mu\alpha.[\alpha]((z)(y)K)((z)(y)[\alpha]((z)(y)K)((z)(y)x)x)x} \\
 &\rightarrow_{\varepsilon} \underline{\mu\gamma.((z)(y)K)((z)(y)((z)(y)K)((z)(y)x)x)x} \\
 &\dots
 \end{aligned}$$

To convince ourselves that this reduction can be continued without reaching a normal form, it is enough to observe that there is a certain analogy between lines 5 and 10. The idea of the proof of non-termination is roughly as follows. We have two occurrences of the subterm K in the μ -term $\mu\gamma.((z)(y)K)K$, from which we can start an infinite reduction sequence. In the first stage of each iteration, we reduce exclusively the μ - and μ' -redexes lying around the second occurrence of K until we arrive at the first occurrence of K wrapped in a subterm $(z)(y)K$. Then we distribute that copy of $(z)(y)K$ among all the subterms of the emerging μ' -redex which are named with α , and afterwards we finish the actual iteration round with an ε -reduction. Intuitively, none of the performed reduction steps decreases the total number of μ - and μ' -redexes in the term, which anticipates that the reduction sequence cannot be terminating.

Let us now properly formalize the proof. First of all we need some notations. For every sequence of μ -terms P, Q_0, \dots, Q_n , we define by induction on n , the μ -term $\langle P; (P_0, \dots, P_n) \rangle$ by $\langle P; (P_0) \rangle = ((z)(y)P)P_0$ and $\forall 1 \leq k \leq n, \langle P; (P_0, \dots, P_n) \rangle = \langle ((z)(y)P)P_n; (P_0, \dots, P_{n-1}) \rangle$. Intuitively $\langle P; (P_0, \dots, P_n) \rangle = ((z)(y) \dots ((z)(y)((z)(y)P)P_n)P_{n-1} \dots)P_0$. We define now two sequences of μ -terms $(Q_n)_n$ and $(R_n)_n$ by :

$Q_0 = x, \forall n \in \mathbb{N}, Q_{n+1} = \langle x; (Q_0, \dots, Q_n) \rangle$ and
 $\forall n \in \mathbb{N}, R_n = \mu\gamma.((z)(y)K)\langle K; (Q_0, \dots, Q_n) \rangle$.
 We have $R_0 = \mu\gamma.((z)(y)K)((z)(y)K)x$ and $N \rightarrow_{\mu\mu'\varepsilon}^+ R_0$.

We prove that $\forall n \in \mathbb{N}, R_n \rightarrow_{\mu\mu'\varepsilon}^+ R_{n+1}$.

$$\begin{aligned} R_n &= \mu\gamma.((z)(y)K)\underbrace{((z)(y) \dots ((z)(y)((z)(y)K)Q_n)Q_{n-1} \dots)}_{Q_0} \\ &\rightarrow_{\mu\mu'}^+ \mu\gamma.((z)(y)K)\mu\alpha.[\alpha]\underbrace{((z)(y) \dots ((z)(y)[\alpha]Q_{n+1})Q_n \dots)}_{Q_0} \\ &\rightarrow_{\mu'} \mu\gamma.\mu\alpha.[\alpha]\underbrace{((z)(y)K)((z)(y) \dots ((z)(y)[\alpha]((z)(y)K)Q_{n+1})Q_n \dots)}_{Q_0} \\ &\rightarrow_{\varepsilon} \mu\gamma.((z)(y)K)((z)(y) \dots ((z)(y)((z)(y)K)Q_{n+1})Q_n \dots)Q_0 \\ &= R_{n+1}. \end{aligned}$$

□

Remark 5.4.

1. The variable z is used only to type the μ -term N .
2. We can show that the μ -term N is not $\mu\mu'\varepsilon$ -normalizable. We omit the rather technical proof of this result, since it is not needed in the sequel.

We may guess that one of the reasons for the failure of strong normalization for the above terms could be the presence of two successive μ -applications $[\cdot]$ (M and N contain $[\alpha][\alpha]$). To represent these terms in a typed environment, we need variables of type $\neg\perp$. We have already seen in the beginning of the previous section that terms of this type appear in the first works of Parigot with the aim of encoding classical tautologies with typed terms. Let us give another example to better understand this phenomenon. In Parigot’s syntax, it is not possible to write a $\lambda\mu$ -term with type $\neg\neg X \rightarrow X$ without using a free variable of type $\neg\perp$. The simplest $\lambda\mu$ -term he proposed is: $\lambda y.\mu\alpha.[\varphi](y)\lambda x.\mu\delta.[\alpha]x$. In this term, the variable φ is free and has type $\neg\perp$. In de Groote’s syntax, we can write a term with type $\neg\neg X \rightarrow X$ without free variables and without a variable of type $\neg\perp$. Here is an example: $\lambda y.\mu\alpha.(y)\lambda x.[\alpha]x$. Many researchers have been interested in this kind of variables. For example, Ariola and Herbelin have made use of variables of type $\neg\perp$ in Ariola and Herbelin (2003) (they called them variables for continuation) in order to establish translations between calculi based on classical logic. It is clear that due to the type of these variables, we can pile up two μ -applications $[\alpha][\alpha]$, which causes the loss of the strong normalization property. The first author studied a calculus in his thesis (Battyányi 2007) where variables are not allowed to have type $\neg\perp$. He showed that the obtained calculus enjoys strong normalization.

5.2 Weak normalization of the μ -calculus

In this section we show that $\mu\mu'\rho\varepsilon$ -reduction is weakly normalizing. This result can be surprising in view of the results that we have discussed before, in particular, given that $\mu\mu'\varepsilon$ -reduction is not even normalizing. We will see that the normalization property of $\mu\mu'\rho\varepsilon$ -reduction essentially originates from the structure of normal form terms with respect to all four rules. For the proof, we generalize our method for the weak normalization of $\mu\mu'$ -calculus (Battyányi and Nour 2020). We obtain a very simple normalization algorithm that is discussed in detail at the end of the section. The algorithm will be able to provide us with different normal forms of a given μ -term. The main idea of our approach is to decide what to do when we face an application $(M)N$. We start with normalizing the μ -terms M and N separately to obtain the normal forms M' and N' . The calculation can only continue if either M' or N' starts with μ . This is where the necessity of studying the substitutions $P[\alpha :=_s Q]$, where $s \in \{r, l\}$ and P, Q are in normal form, emerges.

Lemma 5.7 characterizes the redexes that can be created in such substitutions, and Lemma 5.9 demonstrates how to handle them properly.

We present the proofs in more detail in order to help the reader to easier understand the algorithm.

The following lemma depicts the μ -terms obtained by erasing μ -variables from a normal form.

Lemma 5.5. Let M be a μ -term and α a μ -variable. If $M \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$, then $M_\alpha \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ and, if $M \neq \mu\beta.M'$ for some β , then there is no γ such that $M_\alpha = \mu\gamma.M''$.

Proof. By induction on M using Definition 2.5. □

The next lemma is intuitive and has the consequence that we are not able to create a head μ -redex by a μ - or μ' -substitution.

Lemma 5.6. Let M, N be μ -terms and $s \in \{r, l\}$. If $M[\alpha :=_s N] = \mu\gamma.Q$ for some term Q , then $M = \mu\gamma.P$ for some term P .

Proof. By induction on M . □

Now we characterize the redexes that can occur in a term of the form $M[\alpha :=_r N]$ or $M[\alpha :=_l N]$, where M and N are normal forms.

Lemma 5.7. Let M, N be μ -terms such that $M, N \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$, $\sigma = [\alpha :=_r N]$ and $\sigma' = [\alpha :=_l N]$.

1. We have $M\sigma \in \mathcal{NF}_{\mu\rho\varepsilon}$ and the only possible μ' -redexes of $M\sigma$ are named with α and are of the form $(L)N$ provided $N = \mu\gamma.K$ for some term K . In particular, $M\sigma \in \mathcal{WN}_{\mu\mu'\rho\varepsilon}$ if $N \neq \mu\gamma.K$.
2. We have $M\sigma' \in \mathcal{NF}_{\mu'\rho\varepsilon}$ and the only possible μ -redexes of $M\sigma'$ are named with α and are of the form $(N)L$ provided $N = \mu\gamma.K$ for some term K . In particular, $M\sigma \in \mathcal{WN}_{\mu\mu'\rho\varepsilon}$ if $N \neq \mu\gamma.K$.

Proof. We only prove the first point. The proof proceeds by induction on M .

- The result is obvious if $M = x$.
- If $M = \mu\alpha.P$, then $P \neq \mu\beta.P'$ and $M\sigma = \mu\alpha.P\sigma$. By IH, $P\sigma \in \mathcal{NF}_{\mu\rho\varepsilon}$ and, by Lemma 5.6, $P\sigma \neq \mu\beta.Q$, then $M\sigma \in \mathcal{NF}_{\mu\rho\varepsilon}$.
- If $M = [\beta]P$ where $\beta \neq \alpha$, then $P \neq \mu\gamma.P'$ and $M\sigma = [\beta]P\sigma$. By IH, $P\sigma \in \mathcal{NF}_{\mu\rho\varepsilon}$ and, by Lemma 5.6, $P\sigma \neq \mu\gamma.Q$, then $M\sigma \in \mathcal{NF}_{\mu\rho\varepsilon}$.
- If $M = [\alpha]P$, then $M\sigma = [\alpha](P\sigma)N$. By IH, $P\sigma \in \mathcal{NF}_{\mu\rho\varepsilon}$ and, by Lemma 5.6, $P\sigma \neq \mu\beta.Q$, then $M\sigma \in \mathcal{NF}_{\mu\rho\varepsilon}$.
- If $M = (P)Q$, then $M\sigma = (P\sigma)Q\sigma$ and, since M is neither a μ - and nor a μ' -redex, we have $P \neq \mu\beta.P'$ and $Q \neq \mu\beta.Q'$. Using IH, $P\sigma, Q\sigma \in \mathcal{NF}_{\mu\rho\varepsilon}$ and, by Lemma 5.6, $P\sigma \neq \mu\beta.P''$, $Q\sigma \neq \mu\beta.Q''$. Hence, $M\sigma \in \mathcal{NF}_{\mu\rho\varepsilon}$ and the stipulation for the μ' -redexes is valid for $M\sigma$.

The requirement for the μ' -redex can be checked in all the above cases. □

Definition 5.8 (μ - and μ' -good).

1. A μ -term M is said to be μ -good if $M \in \mathcal{NF}_{\mu\rho\varepsilon}$ and its μ' -redexes are named with μ -variables.

2. A μ -term M is said to be μ' -good if $M \in \mathcal{NF}_{\mu'\rho\varepsilon}$ and its μ -redexes are named with μ -variables.

Let us now show how a μ -term that is μ or μ' -good can be normalized.

Lemma 5.9.

1. If M is μ -good, then $\exists M' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $M \rightarrow_{\mu'\rho} M'$ and $M' = \mu\gamma.L$ for some term L iff $M = \mu\gamma.K$ for some term K ; in particular $M \in \mathcal{WN}_{\mu\mu'\rho\varepsilon}$.
2. If M is μ' -good, then $\exists M' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $M \rightarrow_{\mu\rho} M'$ and $M' = \mu\gamma.L$ for some term L iff $M = \mu\gamma.K$ for some term K ; in particular $M \in \mathcal{WN}_{\mu\mu'\rho\varepsilon}$.

Proof. We only prove the first point by induction on M .

- The result is obvious if $M = x$.
- If $M = \mu\alpha.P$, then P is μ -good and $P \neq \mu\beta.K$. By IH, $\exists P' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $P \rightarrow_{\mu'\rho} P'$ and $P' \neq \mu\beta.L$, then $M \rightarrow_{\mu'\rho} \mu\alpha.P' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
- $M = [\alpha]P$ where P is not a μ -redex, then P is μ -good and $P \neq \mu\beta.K$. Because of IH, $\exists P' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $P \rightarrow_{\mu'\rho} P'$ and $P' \neq \mu\beta.L$, then $M \rightarrow_{\mu'\rho} [\alpha]P' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
- If $M = [\alpha](P)\mu\gamma.Q$, then P, Q are μ -good, $P \neq \mu\beta.K$ and $Q \neq \mu\beta.L$. By IH, $\exists P', Q' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $P \rightarrow_{\mu'\rho} P', Q \rightarrow_{\mu'\rho} Q', P' \neq \mu\beta.K'$ and $Q' \neq \mu\beta.L'$. We have $M \rightarrow_{\mu'\rho} [\alpha](P')\mu\gamma.Q' \rightarrow_{\mu'} [\alpha]\mu\gamma.Q'[\gamma :=_l P'] \rightarrow_{\rho} Q'[\gamma := \alpha][\alpha :=_l P']$, and, by point 2 of Lemma 5.7, $Q'[\gamma := \alpha][\alpha :=_l P'] \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
- If $M = (P)Q$, then P, Q are μ -good, $P \neq \mu\beta.K$ and $Q \neq \mu\beta.L$. Applying IH, $\exists P', Q' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $P \rightarrow_{\mu'\rho} P', Q \rightarrow_{\mu'\rho} Q', P' \neq \mu\beta.K'$ and $Q' \neq \mu\beta.L'$, thus $M \rightarrow_{\mu'\rho} (P')Q' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$. □

We can now prove our weak normalization result.

Theorem 5.10. *The μ -calculus is weakly normalizing.*

Proof. Let M be any μ -term. We prove by induction on M that $M \in \mathcal{WN}_{\mu\mu'\rho\varepsilon}$.

- The result is obvious if $M = x$.
- If $M = \mu\alpha.N$, then, by IH, $\exists N' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $N \rightarrow_{\mu\mu'\rho\varepsilon} N'$. We have $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.N'$. We distinguish two cases.
 - If $N' \neq \mu\beta.K$, then $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.N' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
 - If $N' = \mu\beta.K$, then, by Lemma 5.5, $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.\mu\beta.K \rightarrow_{\varepsilon} \mu\alpha.K\beta \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
- If $M = [\alpha]N$, then, using IH, $\exists N' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $N \rightarrow_{\mu\mu'\rho\varepsilon} N'$. We have $M \rightarrow_{\mu\mu'\rho\varepsilon} [\alpha]N'$. We distinguish two cases.
 - If $N' \neq \mu\beta.K$, then $M \rightarrow_{\mu\mu'\rho\varepsilon} [\alpha]N' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
 - If $N' = \mu\beta.K$, then $M \rightarrow_{\mu\mu'\rho\varepsilon} [\alpha]\mu\beta.K \rightarrow_{\rho} K[\beta := \alpha] \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
- If $M = (P)Q$, then, by IH, $\exists P', Q' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $P \rightarrow_{\mu\mu'\rho\varepsilon} P'$ and $Q \rightarrow_{\mu\mu'\rho\varepsilon} Q'$. We have $M \rightarrow_{\mu\mu'\rho\varepsilon} (P')Q'$. We distinguish four cases.
 - If $P' \neq \mu\alpha.K$ and $Q' \neq \mu\beta.L$, then $M \rightarrow_{\mu\mu'\rho\varepsilon} (P')Q' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
 - If $P' = \mu\alpha.K$ and $Q' \neq \mu\beta.L$, then $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.K[\alpha :=_r Q']$. Recall that, by $P' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$, $K \neq \mu\gamma.K'$. Hence, by Lemmas 5.7 and 5.6, $K[\alpha :=_r Q'] \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ and $K[\alpha :=_r Q'] \neq \mu\gamma.O$, then $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.K[\alpha :=_r Q'] \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
 - If $P' \neq \mu\alpha.K$ and $Q' = \mu\beta.L$, then $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\beta.L[\alpha :=_l P']$. By Lemmas 5.7 and 5.6, $L[\alpha :=_l P'] \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ and $L[\alpha :=_l P'] \neq \mu\gamma.O$, then $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.L[\alpha :=_l P'] \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.

- $P' = \mu\alpha.K$ and $Q' = \mu\beta.L$, then we can conclude in two different ways.
 - * We have $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.K[\alpha :=_r \mu\beta.L]$. By Lemmas 5.7 and 5.6, $K[\alpha :=_r \mu\beta.L]$ is μ -good and $K[\alpha :=_r \mu\beta.L] \neq \mu\gamma.O$, then, by Lemma 5.9, $\exists T \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $K[\alpha :=_r \mu\beta.L] \rightarrow_{\mu'\rho} T$ and $T \neq \mu\gamma.O'$, thus $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.T \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$.
 - * We have $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\beta.L[\beta :=_l \mu\alpha.K]$. By Lemmas 5.7 and 5.6, $L[\beta :=_l \mu\alpha.K]$ is μ' -good and $L[\beta :=_l \mu\alpha.K] \neq \mu\gamma.O$, then, by Lemma 5.9, $\exists T \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ such that $L[\beta :=_l \mu\alpha.K] \rightarrow_{\mu\rho} T$ and $T \neq \mu\gamma.O'$, thus $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\beta.T \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$. □

Remark 5.11. Let us summarize the steps that we took to present a normalization algorithm. We proceed by induction on μ -terms.

1. If $M = \mu\alpha.N$, we normalize N and, if we arrive at a normal form that begins with a $\mu\beta$, then we reduce the ε -redex.
2. If $M = [\alpha]N$, we normalize N and, if we arrive at a normal form that begins with a $\mu\beta$, then we reduce the ρ -redex.
3. If $M = (M_1)M_2$, we begin with normalizing M_1 and M_2 , say, to normal forms N_1 and N_2 . At this point, we distinguish four cases.
 - (a) If neither N_1 , nor N_2 begins with a μ , the normalization is finished.
 - (b) If $N_1 = \mu\alpha.N'_1$ and $N_2 = \mu\beta.N'_2$, then we reduce the μ -redex in order to obtain the normal form $\mu\alpha.N'_1[\alpha :=_r N_2]$.
 - (c) If $N_1 \neq \mu\alpha.N'_1$ and $N_2 = \mu\beta.N'_2$, then we reduce the μ' -redex. Hence, we obtain the normal form $\mu\beta.N'_2[\alpha :=_l N_1]$.
 - (d) If $N_1 = \mu\alpha.N'_1$ and $N_2 = \mu\beta.N'_2$, then
 - i. One can reduce the μ -redex so that one obtains a μ -good term $\mu\alpha.N'_1[\alpha :=_r N_2]$. It is now sufficient to reduce the remaining μ' -redexes and then the created ρ - and ε -redexes starting with the innermost ones.
 - ii. Or one can reduce the μ' -redex so that one obtains a μ' -good term $\mu\beta.N'_2[\alpha :=_l N_1]$. It is now sufficient to reduce the remaining μ -redexes and then the created ρ - and ε -redexes starting with the innermost ones.

Observe that in case (d) we can obtain two different normal forms for M .

We now take the μ -terms given in the proofs of Theorems 5.1 and 5.3 and normalize them with our algorithm.

Example 5.12.

1. Following the algorithm, we have two possibilities to normalize $M = (\mu\beta.\mu\alpha.[\alpha][\alpha]x)\mu\alpha.[\alpha][\alpha]x$.

$$\begin{aligned}
 M &= (\mu\beta.\mu\alpha.[\alpha][\alpha]x)\mu\alpha.[\alpha][\alpha]x \\
 &\rightarrow_{\varepsilon} \underline{(\mu\beta.x)\mu\alpha.[\alpha][\alpha]x} \\
 &\rightarrow_{\mu} \mu\beta.x.
 \end{aligned}$$

$$\begin{aligned}
 M &= (\mu\beta.\mu\alpha.[\alpha][\alpha]x)\mu\alpha.[\alpha][\alpha]x \\
 &\rightarrow_{\varepsilon} \underline{(\mu\beta.x)\mu\alpha.[\alpha][\alpha]x} \\
 &\rightarrow_{\mu'} \underline{\mu\alpha.[\alpha](\mu\beta.x)[\alpha](\mu\beta.x)x} \\
 &\rightarrow_{\mu} \underline{\mu\alpha.[\alpha](\mu\beta.x)[\alpha]\mu\beta.x} \\
 &\rightarrow_{\rho} \underline{\mu\alpha.[\alpha](\mu\beta.x)x}
 \end{aligned}$$

$$\begin{aligned} &\rightarrow_{\mu} \mu\alpha.\underline{[\alpha]\mu\beta.x} \\ &\rightarrow_{\rho} \mu\alpha.x. \end{aligned}$$

2. Following the algorithm, we have two possibilities to normalize $N = \mu\gamma.(\mu\beta.[\beta](z)[\beta]y)\mu\alpha.[\alpha][\alpha]x.$

$$\begin{aligned} N &= \mu\gamma.(\mu\beta.[\beta](z)[\beta]y)\mu\alpha.[\alpha][\alpha]x \\ &\rightarrow_{\mu} \mu\gamma.\mu\beta.[\beta]((z)[\beta](y)\mu\alpha.[\alpha][\alpha]x)\mu\alpha.[\alpha][\alpha]x \\ &\rightarrow_{\mu'} \mu\gamma.\mu\beta.[\beta]((z)[\beta]\mu\alpha.[\alpha](y)[\alpha](y)x)\mu\alpha.[\alpha][\alpha]x \\ &\rightarrow_{\rho} \mu\gamma.\mu\beta.[\beta]((z)[\beta](y)[\beta](y)x)\mu\alpha.[\alpha][\alpha]x \\ &\rightarrow_{\mu'} \mu\gamma.\mu\beta.[\beta]\mu\alpha.[\alpha]((z)[\beta](y)[\beta](y)x)[\alpha]((z)[\beta](y)[\beta](y)x)x \\ &\rightarrow_{\rho} \underline{\mu\gamma.\mu\beta.[\beta]((z)[\beta](y)[\beta](y)x)[\beta]((z)[\beta](y)[\beta](y)x)x} \\ &\rightarrow_{\varepsilon} \mu\gamma.((z)(y)(y)x)((z)(y)(y)x)x. \end{aligned}$$

$$\begin{aligned} N &= \mu\gamma.(\mu\beta.[\beta](z)[\beta]y)\mu\alpha.[\alpha][\alpha]x \\ &\rightarrow_{\mu'} \mu\gamma.\mu\alpha.[\alpha](\mu\beta.[\beta](z)[\beta]y)[\alpha](\mu\beta.[\beta](z)[\beta]y)x \\ &\rightarrow_{\mu} \mu\gamma.\mu\alpha.[\alpha](\mu\beta.[\beta](z)[\beta]y)[\alpha]\mu\beta.[\beta]((z)[\beta](y)x)x \\ &\rightarrow_{\rho} \mu\gamma.\mu\alpha.[\alpha](\mu\beta.[\beta](z)[\beta]y)[\alpha]((z)[\alpha](y)x)x \\ &\rightarrow_{\mu} \mu\gamma.\mu\alpha.[\alpha]\mu\beta.[\beta]((z)[\beta](y)[\alpha]((z)[\alpha](y)x)x)[\alpha]((z)[\alpha](y)x)x \\ &\rightarrow_{\rho} \underline{\mu\gamma.\mu\alpha.[\alpha]((z)[\alpha](y)[\alpha]((z)[\alpha](y)x)x)[\alpha]((z)[\alpha](y)x)x} \\ &\rightarrow_{\varepsilon} \mu\gamma.((z)(y)((z)(y)x)x)((z)(y)x)x. \end{aligned}$$

We note that we do not obtain the same normal form.

6. The Case of β -Reduction

In this section, we examine how the weak normalization property is preserved when we extend the calculus with β -reduction. In what follows, since we include β -reduction in our treatment again, instead of the abbreviations μ -reduction and μ -normal form for $\mu\mu'\rho\varepsilon$ -reduction and $\mu\mu'\rho\varepsilon$ -normal form, we explicitly indicate the particular reductions we are talking about. Although the algorithms of the previous section also work in the untyped setting, we now must consider a typed framework, since β -reduction is not weakly normalizing in the untyped form. Hence, all results we state in this section are for typed $\lambda\mu$ -terms. Our algorithm depends on the weak normalization algorithm presented in the previous section, and we also make use of the fact that the typed λ -calculus has the weak normalization property. The idea of the proof is as follows:

- We start from a $\lambda\mu$ -term M_1 in $\mu\mu'\rho\varepsilon$ -normal form, i.e., from a $\lambda\mu$ -term in which there are no μ -, μ' -, ρ - or ε -redexes.
- We eliminate all β -redexes from M_1 by applying an arbitrary weak β -normalization algorithm. Having done this, we arrive at a $\lambda\mu$ -term M_2 in β -normal form.
- Next, we find a $\mu\mu'\rho\varepsilon$ -normal form M_3 of M_2 by our weak normalization algorithm. The $\lambda\mu$ -term M_3 may contain β -redexes.
- We prove, however, that the maximum rank of β -redexes in M_1 is strictly greater than that of M_3 . We will give the necessary definitions in the section.

By this, the process must terminate. Finally, θ -reduction causes no problem since it is strongly normalizing and does not create new redexes when applied to $\mu\mu'\rho\varepsilon$ -normal forms.

We have to consider the evolution of types throughout the reduction. In order to simplify our presentation, we apply the typability relation of Church (Sørensen and Urzyczyn 2006): we explicitly write the types of the variables in the abstractions together with the types of some of the subterms. To begin with, we present the reduction rules in the typed case. Recall that this evolution of types corresponds exactly to the notion of cut elimination in natural deduction proofs. We will not indicate the types of all subterms of a particular $\lambda\mu$ -term, but only the ones that are relevant for our purposes.

$$\begin{array}{ll}
 (\lambda x^A.M^B)N^A & \rightarrow_{\beta} M^B[x^A := N^A] \\
 (\mu\alpha^{-(A \rightarrow B)}.M^{\perp})N^A & \rightarrow_{\mu} \mu\alpha^{-B}.M^{\perp}[\alpha^{-B} :=_r N^A] \\
 (N^{A \rightarrow B})\mu\alpha^{-A}.M^{\perp} & \rightarrow_{\mu'} \mu\alpha^{-B}.M^{\perp}[\alpha^{-B} :=_l N^{A \rightarrow B}] \\
 [\beta^{-A}]\mu\alpha^{-A}.M^{\perp} & \rightarrow_{\rho} M^{\perp}[\alpha^{-A} := \beta^{-A}] \\
 \mu\alpha^{-A}.[\alpha^{-A}]M^A & \rightarrow_{\theta} M^A \\
 \mu\alpha^{-\perp}.\mu\beta^{-\perp}.M^{\perp} & \rightarrow_{\varepsilon} \mu\alpha^{-\perp}.M^{\perp}_{\beta}
 \end{array}$$

Definition 6.1. The length of a type A , which is denoted by $\text{lh}(A)$, is defined as the number of arrows in A .

6.1 The origin of a μ -redex in a β -reduction

The aim of this section is to prove Corollary 6.7, which asserts that the creation of a μ -redex during β -reduction of a $\mu\mu'\rho\varepsilon$ -normal term M forces M to contain a β -redex of a particular type. We start from M and M' , where $M \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ and $M \rightarrow_{\beta} M'$. We suppose that M' contains a μ -redex $(\mu\alpha^{-(A \rightarrow B)}.P^{\perp}_1)Q^A_1$. We intend to say something about the term M , where M is such that the sequence of β -reductions leading to M' has created a new μ -redex in M' . We show that M must contain a β -redex $(\lambda x^C.P^D_2)Q^C_2$ with $\text{lh}(C \rightarrow D) > \text{lh}(A \rightarrow B)$. We consider this β -redex the origin of the μ -redex. (In fact, a β -redex like that can be uniquely assigned to every μ -redex of the above form, but we do not need this statement.) To establish this claim, we have to start from a more general assumption where the property $M \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ is no longer valid.

The assertion of the following lemma explicitly shows the origins of β - and μ -redexes in a β -substitution.

Lemma 6.2.

1. If $(\lambda x.P)Q \leq M[y := N]$, then
 - either $(\lambda x.P)Q \leq N$,
 - or $(\lambda x.P')Q' \leq M$, $P'[y := N] = P$ and $Q'[y := N] = Q$ for some terms P', Q' ,
 - or $(y)M' \leq M$, $N = \lambda x.P$ and $M'[y := N] = Q$ for some term M' .
2. If $(\mu\alpha.P)Q \leq M[y := N]$, then
 - either $(\mu\alpha.P)Q \leq N$,
 - or $(\mu\alpha.P')Q' \leq M$, $P'[y := N] = P$ and $Q'[y := N] = Q$ for some terms P', Q' ,
 - or $(y)M' \leq M$, $N = \mu\alpha.P$ and $M'[y := N] = Q$ for some term M' .

Proof. By induction on M . □

Lemmas 6.3 and 6.4 determine in some sense the origin of a β - or μ -redex in a β -reduction while also stating important relations between the type of the redexes in the original term and in the resulting term, respectively. We can conclude that, during a β -reduction step, a μ -redex can stem from either a β - or a μ -redex. When the origin is a β -redex, then it must be a redex of longer

type. On the contrary, if we consider the origin of a β -redex, then it must be a β -redex again with type not smaller than that of the descendant. This is the statement of Lemma 6.4. Hence, if we examine a sequence of β -reductions starting from a term in μ -normal form, the origin of a μ -redex in the final term must be a β -redex in the original term of longer type (Corollary 6.7).

Lemma 6.3. If $M \rightarrow_{\beta} M'$ and $(\mu\alpha^{-\langle A \rightarrow B \rangle}.P_1^{\perp})Q_1^A \preceq M'$, then either $(\mu\alpha^{-\langle A \rightarrow B \rangle}.P_2^{\perp})Q_2^A \preceq M$ or $(\lambda x^C.P_2^D)Q_2^C \preceq M$ for some terms P_2, Q_2 and some types C, D where $\text{lh}(C \rightarrow D) > \text{lh}(A \rightarrow B)$.

Proof. By induction on M . The only nontrivial cases are

- $M = (M_1)M_2$ is not the reduced β -redex and $M' = (\mu\alpha.P_1)Q_1$,
 - If $M_1 = \mu\alpha.P_1$ and $M_2 \rightarrow_{\beta} Q_1$, then $M = (\mu\alpha^{-\langle A \rightarrow B \rangle}.P_1^{\perp})M_2^A$.
 - If $M_1 \rightarrow_{\beta} \mu\alpha.P_1$ and $M_2 = Q_1$, we only have two possible cases :
 - * $M_1 = \mu\alpha.P_2, P_2 \rightarrow_{\beta} P_1$ and $M = (\mu\alpha^{-\langle A \rightarrow B \rangle}.P_2^{\perp})M_2^A$.
 - * $M_1 = (\lambda y.P_2)Q_2 \rightarrow_{\beta} \mu\alpha.P_1$ and $(\lambda y^C.P_2^{A \rightarrow B})Q_2^C \preceq M$ with $\text{lh}(C \rightarrow (A \rightarrow B)) > \text{lh}(A \rightarrow B)$.
- $M = (\lambda y.M'_1)M_2$ is the reduced β -redex and $M' = M'_1[y := M_2]$. Then, by Lemma 6.2, we have three cases to examine.
 - If $(\mu\alpha.P_1)Q_1 \preceq M_2$, then we have the result.
 - If $(\mu\alpha.P_1)Q_1 = K[y := M_2]$ where $K = (\mu\alpha.P_2)Q_2 \preceq M'_1$, then $(\mu\alpha^{-\langle A \rightarrow B \rangle}.P_2^{\perp})Q_2^A \preceq M$.
 - If $(y)O \preceq M'_1, M_2 = \mu\alpha.P_1$ and $O[y := M_2] = Q_1$, then $M = (\lambda y^{A \rightarrow B}.M_1^C)M_2^{A \rightarrow B}$ and $\text{lh}((A \rightarrow B) \rightarrow C) > \text{lh}(A \rightarrow B)$.

□

Lemma 6.4. If $M \rightarrow_{\beta} M'$ and $(\lambda x^A.P_1^B)Q_1^A \preceq M'$, then $(\lambda x^C.P_2^D)Q_2^C \preceq M$ for some terms P_2, Q_2 and some types C, D where $\text{lh}(C \rightarrow D) \geq \text{lh}(A \rightarrow B)$.

Proof. By induction on M . The only nontrivial cases are

- $M = (M_1)M_2$ is not the reduced β -redex and $M' = (\lambda x.P_1)Q_1$,
 - If $M_1 = \lambda x.P_1$ and $M_2 \rightarrow_{\beta} Q_1$, then $M = (\lambda x^A.P_1^B)M_2^A$.
 - If $M_1 \rightarrow_{\beta} \lambda x.P_1$ and $M_2 = Q_1$, we have only two possible cases :
 - * $M_1 = \lambda x.P_2, P_1 \rightarrow_{\beta} P_2$ and $M = (\lambda x^A.P_2^B)M_2^A$.
 - * $M_1 = (\lambda y.P_2)Q_2 \rightarrow_{\beta} \lambda x.P_1$ and $(\lambda y^C.P_2^{A \rightarrow B})Q_2^C \preceq M$ with $\text{lh}(C \rightarrow (A \rightarrow B)) > \text{lh}(A \rightarrow B)$.
- $M = (\lambda y.M'_1)M_2$ is the reduced β -redex and $M' = M'_1[y := M_2]$. Then, by Lemma 6.2, we have three cases to examine.
 - If $(\lambda x.P_1)Q_1 \preceq M_2$, then we have the result.
 - If $(\lambda x.P_1)Q_1 = K[y := M_2]$ where $K = (\lambda x.P_2)Q_2 \preceq M'_1$, then $(\lambda x^A.P_2^B)Q_2^A \preceq M$.
 - If $(y)O \preceq M'_1, M_2 = \lambda x.P_1$ and $O[y := M_2] = Q_1$, then $M = (\lambda y^{A \rightarrow B}.M_1^C)M_2^{A \rightarrow B}$ and $\text{lh}((A \rightarrow B) \rightarrow C) > \text{lh}(A \rightarrow B)$.

□

We obtain the corollaries below about β -reduction sequences. Intuitively, the origin of a μ -redex in a β -reduction sequence is either a μ -redex of the same type or a β -redex of a type of greater length.

Corollary 6.5. If $M \rightarrow_{\beta} M'$ and $(\lambda x^A.P_1^B)Q_1^A \preceq M'$, then $(\lambda x^C.P_2^D)Q_2^C \preceq M$ for some terms P_2, Q_2 and some types C, D where $\text{lh}(C \rightarrow D) \geq \text{lh}(A \rightarrow B)$.

Proof. By induction on the length of the reduction $M \rightarrow_{\beta} M'$ using Lemma 6.4. □

Corollary 6.6. If $M \rightarrow_{\beta} M'$ and $(\mu\alpha^{-(A \rightarrow B)}.P_1^{\perp})Q_1^A \preceq M'$, then either $(\mu\alpha^{-(A \rightarrow B)}.P_2^{\perp})Q_2^A \preceq M$ or $(\lambda x^C.P_2^D)Q_2^C \preceq M$ for some terms P_2, Q_2 and some types C, D where $\text{lh}(C \rightarrow D) > \text{lh}(A \rightarrow B)$.

Proof. By induction on the length of the reduction $M \rightarrow_{\beta} M'$. If $M = M'$, the result is immediate. If $M \rightarrow_{\beta} N \rightarrow_{\beta} M'$, then, by Lemma 6.3, either $(\mu\alpha^{-(A \rightarrow B)}.P_3^{\perp})Q_3^A \preceq N$ or $(\lambda x^E.P_3^F)Q_3^E \preceq N$ where $\text{lh}(E \rightarrow F) > \text{lh}(A \rightarrow B)$.

- If $(\lambda x^E.P_3^F)Q_3^E \preceq N$ where $\text{lh}(E \rightarrow F) > \text{lh}(A \rightarrow B)$, then by Corollary 6.5, $(\lambda x^C.P_2^D)Q_2^C \preceq M$ where $\text{lh}(C \rightarrow D) \geq \text{lh}(E \rightarrow F) > \text{lh}(A \rightarrow B)$.
- If $(\mu\alpha^{-(A \rightarrow B)}.P_3^{\perp})Q_3^A \preceq N$, then by IH, either $(\mu\alpha^{-(A \rightarrow B)}.P_2^{\perp})Q_2^A \preceq M$ or $(\lambda x^C.P_2^D)Q_2^C \preceq M$ where $\text{lh}(C \rightarrow D) > \text{lh}(A \rightarrow B)$.

□

Corollary 6.7. If $M \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$, $M \rightarrow_{\beta} M'$, and $(\mu\alpha^{-(A \rightarrow B)}.P_1^{\perp})Q_1^A \preceq M'$, then $(\lambda x^C.P_2^D)Q_2^C \preceq M$ for some terms P_2, Q_2 and some types C, D where $\text{lh}(C \rightarrow D) > \text{lh}(A \rightarrow B)$.

Proof. Immediate from Corollary 6.6. □

6.2 The origin of a β -redex in a $\mu\mu'\rho\varepsilon$ -reduction

The aim of this section is to prove Corollary 6.17, which means that the creation of a β -redex during a $\mu\mu'\rho\varepsilon$ -reduction sequence starting from a β -normal term M compels M to contain a $\mu\mu'\rho\varepsilon$ -redex of a particular type. We start from M and N such that $M \in \mathcal{NF}_{\beta}$ and $M \rightarrow_{\mu\mu'\rho\varepsilon} N$. We assume that N contains a β -redex $(\lambda x^A.P_1^B)Q_1^A$. We show that M must then contain a μ -redex $(\mu\alpha^{-(A \rightarrow B)}.P_2^{\perp})Q_2^A$. We consider this μ -redex as the origin of the β -redex. (In fact, a μ -redex like that can be uniquely assigned to every β -redex of the above form, but we do not need this statement.) To establish this claim, we have to start from a more general assumption where $M \in \mathcal{NF}_{\beta}$ is no longer valid. We note that two cases can arise here: the origin of a β -redex with respect to a $\mu\mu'\rho\varepsilon$ -reduction step can either be a β -redex or a μ -redex. When the origin of a β -redex is a μ -redex, then we have to resort to more subtle methods in our investigation. To this end, we introduce two notions: we describe when a term is α -good and when it verifies condition (A, B) for certain types A, B (Definitions 6.9 and 6.12). Intuitively, these notions deal with the situation when the the name of the μ -variable changes because of a ρ -reduction.

The following lemma gives the location of β - and μ -redexes in a μ -substitution.

Lemma 6.8. Let $s \in \{l, r\}$

1. If $(\lambda x.P)Q \preceq M[\beta :=_s N]$, then
 - either $(\lambda x.P)Q \preceq N$,
 - or $(\lambda x.P')Q' \preceq M$, $P'[\beta :=_s N] = P$ and $Q'[\beta :=_s N] = Q$ for some terms P', Q' ,
 - or $s = r$, $[\beta]\lambda x.P' \preceq M$, $N = Q$ and $P'[\beta :=_r N] = P$ for some term P' ,
 - or $s = l$, $[\beta]Q' \preceq M$, $N = \lambda x.P$ and $Q'[\beta :=_l N] = Q$ for some term Q' .
2. If $(\mu\alpha.P)Q \preceq M[\beta :=_s N]$, then
 - either $(\mu\alpha.P)Q \preceq N$,

- or $(\mu\alpha.P')Q' \preceq M$, $U'[\beta :=_s N] = U$ and $Q'[\beta :=_s N] = Q$ for some terms P' , Q' ,
- or $s = r$, $[\beta]\mu\alpha.P' \preceq M$, $N = Q$ and $P'[\beta :=_r N] = P$ for some term P' ,
- or $s = l$, $[\beta]Q' \preceq M$, $N = \mu\alpha.P$ and $Q'[\beta :=_l N] = Q$ for some term Q' .

Proof. By induction on M . □

In what follows, we show that the origin of a μ -redex through a $\mu\mu'\rho\varepsilon$ -reduction sequence is always a μ -redex of the same type (Lemma 6.14). However, we need to add an auxiliary condition on the μ -redex to ensure this property during the search for the origin of a redex (Lemma 6.16). Hence, we give Definitions 6.9 and 6.12.

Definition 6.9. Let M be a $\lambda\mu$ -term, A, B types and $\alpha^{-(A \rightarrow B)} \in \text{fv}(M)$. We say that M is α -good if there exists an occurrence of $[\alpha]$ such that

- either $[\alpha]\lambda y^A.O^B \preceq M$ for some term O , or
- $[\alpha]\mu\beta^{-(A \rightarrow B)}.P^\perp \preceq M$ for some term P which is β -good.

Intuitively a $\lambda\mu$ -term is α -good if it contains a subterm of the following form: $[\alpha]\mu\beta_1 \dots [\beta_1]\mu\beta_2 \dots [\beta_2]\mu\beta_3 \dots [\beta_n]\lambda y.O$.

The next lemma helps understanding the notion of α -good with respect to a substitution and will be used in subsequent proofs.

Lemma 6.10. Let M be a $\lambda\mu$ -term.

1. Assume $M = P[\beta :=_s Q]$, $s \in \{l, r\}$ and $\beta \neq \alpha$.
 - (a) If M is α -good, then either P is α -good or Q is α -good.
 - (b) If P is α -good, then M is α -good.
 - (c) If $\beta \in \text{fv}(P)$ and Q is α -good, then M is α -good.
2. If $\gamma^{-\perp}$ is a μ -variable, then M is α -good iff M_γ is α -good.
3. If $\alpha \notin \text{fv}(N)$ and $s \in \{l, r\}$, then $M[\alpha :=_s N]$ is not α -good.

Proof. By induction on M . □

The following lemma intuitively means that we will not be able to create an α -good term by a $\mu\mu'\rho\varepsilon$ -reduction.

Lemma 6.11. Let $M \rightarrow_{\mu\mu'\rho\varepsilon} M'$ and assume M' is α -good, then M is α -good.

Proof. It is enough to examine the case of one-step reduction. We proceed by induction on M . The cases $M = \lambda x.N$, $M = \mu\beta.N$, $M = (M_1)M_2$, and $M = [\beta]N$, where $\beta \neq \alpha$, are treated by induction and using Lemma 6.10. Assume $M = [\alpha]N$. We examine the form of N .

- If $N = \lambda y.O$, then M is α -good.
- The case $N = [\beta]O$ is impossible, since the type of N differs from \perp by reason of $\alpha : \neg(A \rightarrow B)$.
- Assume $N = \mu\beta.O$.
 - If $M \rightarrow_{\mu\mu'\rho\varepsilon} [\alpha]\mu\beta.O' = M'$ where $O \rightarrow_{\mu\mu'\rho\varepsilon} O'$, then O' is α -good or β -good, thus, by IH, O is α -good or β -good, therefore M is also α -good.
 - If $M = [\alpha]\mu\beta.O \rightarrow_\rho O[\beta := \alpha] = M'$, then O is α -good or β -good, thus M is also α -good.
- Assume $N = (N_1)N_2$.
 - If $M \rightarrow_{\mu\mu'\rho\varepsilon} [\alpha](N'_1)N_2 = M'$ where $N_1 \rightarrow_{\mu\mu'\rho\varepsilon} N'_1$, then N'_1 or N_2 is α -good, thus, by IH, N_1 or N_2 is α -good, therefore M is also α -good.
 - The case $M \rightarrow_{\mu\mu'\rho\varepsilon} [\alpha](N_1)N'_2 = M'$ where $N_2 \rightarrow_{\mu\mu'\rho\varepsilon} N'_2$ is treated in the same way.

- If $M = [\alpha](\mu\beta.N_3)N_2 \rightarrow_{\mu} [\alpha]\mu\beta.N_3[\beta :=_r N_2] = M'$, then, by Lemma 6.10, $N_3[\beta :=_r N_2]$ is not β -good, thus $N_3[\beta :=_r N_2]$ is α -good, and, by Lemma 6.10, N_3 or N_2 is α -good, therefore M is also α -good.
- The case $M = [\alpha](N_1)\mu\beta.N_2 \rightarrow_{\mu'} [\alpha]\mu\beta.N_2[\beta :=_l N_1] = M'$ is treated in the same way. □

In the next definition, we give the condition that should be imposed on the μ -redex to ensure that our typability assertions hold throughout the reduction sequence. For a redex $(\mu\alpha.P)Q$, we demand that there exists a chain, in the sense of Definition 6.9, that starts with $[\alpha]$ and ends with a λ -abstraction. We will see in the proof of Lemma 6.13 how this condition propagates and how it prevents the creation of certain μ -redexes.

Definition 6.12. Let M be a $\lambda\mu$ -term and A, B be types. We say that M verifies the condition (A, B) , or (A, B) holds for M , if M contains a subterm of the form $(\mu\alpha^{-(A \rightarrow B)}.P^{\perp})Q^A$ for some terms P, Q such that P is α -good.

Lemma 6.13. Let M be a $\lambda\mu$ -term and $\gamma^{-\perp}$ be a μ -variable, then M verifies the condition (A, B) iff M_{γ} verifies the condition (A, B) .

Proof. By induction on M . □

Lemma 6.14. Let $M \rightarrow_{\mu\mu'\rho\varepsilon} M'$ and assume condition (A, B) holds for M' . Then condition (A, B) holds for M .

Proof. It is enough to examine the case of one-step reduction. We proceed by induction on M . The cases $M = \lambda x.N$, $M = \mu\beta.N$, and $M = [\beta]N$ are treated by induction and using Lemma 6.13.

- Assume $M = (M_1)M_2$ and M is not the reduced redex.
 - The case $M' = (M_1)M'_2$, $M_2 \rightarrow_{\mu\mu'\rho\varepsilon} M'_2$ and (A, B) holds for M_1 or for M'_2 , then, using IH, (A, B) holds for M_1 or for M_2 , thus (A, B) holds for M .
 - The case $M' = (M_1)M'_2$, $M_2 \rightarrow_{\mu\mu'\rho\varepsilon} M'_2$, $M_1 = \mu\alpha^{-(A \rightarrow B)}.P$ and P is α -good, then (A, B) holds for M .
 - If $M' = (M'_1)M_2$, $M_1 \rightarrow_{\mu\mu'\rho\varepsilon} M'_1$ and (A, B) holds for M'_1 or for M_2 , then, by IH, (A, B) holds for M_1 or for M_2 , thus (A, B) holds for M .
 - If $M' = (M'_1)M_2$, $M_1 \rightarrow_{\mu\mu'\rho\varepsilon} M'_1$, $M'_1 = \mu\alpha^{-(A \rightarrow B)}.P$ and P is α -good, then we examine the possible forms of M_1 .
 - * The cases $M_1 = \lambda x.N$ and $M_1 = [\beta]N$ are impossible.
 - * If $M_1 = \mu\alpha.N$ and $N \rightarrow_{\mu\mu'\rho\varepsilon} P$, then, by Lemma 6.11, N is α -good, thus M verifies the condition (A, B) .
 - * If $M_1 = \mu\alpha.\mu\gamma.N \rightarrow_{\varepsilon} \mu\alpha.N_{\gamma} = \mu\alpha.P$, then, by Lemma 6.10, N is α -good, thus M_1 verifies the condition (A, B) .
 - * If $M_1 = (\mu\alpha.K)L \rightarrow_{\mu} \mu\alpha.K[\alpha :=_r L] = \mu\alpha.P$, then $K[\alpha :=_r L]$ would be α -good which is impossible according to Lemma 6.10.
 - * If $M_1 = (L)\mu\alpha.K \rightarrow_{\mu'} \mu\alpha.K[\alpha :=_l L] = \mu\alpha.P$, then $K[\alpha :=_l L]$ would be α -good which is impossible according to Lemma 6.10.
- If $M = (\mu\beta^{-(C \rightarrow D)}.M_3^{\perp})M_2^C$ is the reduced redex and $M' = \mu\beta.M_3[\beta :=_r M_2]$, then, by Lemma 6.8, we have three subcases to examine.
 - The case where (A, B) holds for M_2 is immediate.
 - If $(\mu\alpha^{-(A \rightarrow B)}.P^{\perp})Q^A = O[\beta :=_r M_2]$ for some $O = (\mu\alpha^{-(A \rightarrow B)}.P_1^{\perp})Q_1^A \leq M_3$ and $P_1[\beta :=_r M_2]$ is α -good, then, since $\alpha \notin \text{fv}(M_2)$, by Lemma 6.10, P_1 is α -good, thus M verifies the condition (A, B) .

- If $[\beta]\mu\alpha.P_1 \leq M_3$, $(\mu\alpha^{-(A \rightarrow B)}.P_1[\beta := M_2]^\perp)M_2^A \leq M'$ and $P_1[\beta := M_2]$ is α -good, then $C \rightarrow D = A \rightarrow B$ and since $\alpha \notin \text{fv}(M_2)$, by Lemma 6.10, P_1 is α -good, thus M verifies the condition (A, B) .
- If $M = (M_1^{C \rightarrow D})\mu\beta^{-C}.M_3^\perp$ is the reduced redex and $M' = \mu\beta.M_3[\beta :=_l M_1]$, then, by Lemma 6.8, we have three subcases to consider.
 - The case where (A, B) holds for M_1 is immediate.
 - If $(\mu\alpha^{-(A \rightarrow B)}.P^\perp)Q^A = O[\beta :=_l M_1]$ for some $O = (\mu\alpha^{-(A \rightarrow B)}.P^\perp)Q_1^A \leq M_3$ and $P_1[\beta :=_l M_1]$ is α -good, then, since $\alpha \notin \text{fv}(M_1)$, by Lemma 6.10, P_1 is α -good, thus M verifies the condition (A, B) .
 - If $M_1 = \mu\alpha.P$, $[\beta]Q_1 \leq M_3$, $(\mu\alpha^{-(A \rightarrow B)}.P^\perp)Q_1[\beta :=_l M_1]^A \leq M'$ and P is α -good, then $C \rightarrow D = A \rightarrow B$ and M verifies the condition (A, B) .

□

The lemma below will be applied in the proof of Lemma 6.16 in the case of ε -reduction.

Lemma 6.15. Let M be a $\lambda\mu$ -term, $\gamma^{-\perp}$ a μ -variable and A, B be two types. Assume $(\lambda x^A.P^B)Q^A \leq M_\gamma$. Then $(\lambda x^A.P_1^B)Q_1^A \leq M$ for some terms P_1, Q_1 .

Proof. By induction on M .

□

The next lemma shows that the origin of a β -redex of type $A \rightarrow B$ through a $\mu\mu'\rho\varepsilon$ -reduction sequence is either a β -redex or a μ -redex verifying the condition (A, B) of the same type.

Lemma 6.16. If $M \rightarrow_{\mu\mu'\rho\varepsilon} M'$ and $(\lambda x^A.P^B)Q^A \leq M'$, then either $(\lambda x^A.P_1^B)Q_1^A \leq M$ for some terms P_1, Q_1 or the condition (A, B) holds for M .

Proof. By Lemma 6.14, it is enough to examine the case of one-step reduction. We proceed by induction on M . The cases $M = \lambda y.N$, $M = \mu\beta.N$, and $M = [\beta]N$ are treated by induction and using Lemma 6.15.

- Assume $M = (M_1)M_2$ and M is not the reduced redex.
 - If $M' = (M'_1)M_2$, $M_1 \rightarrow_{\mu\mu'\rho\varepsilon} M'_1$ and $(\lambda x^A.P^B)Q^A \leq M'_1$ or $(\lambda x^A.P^B)Q^A \leq M_2$, then, in the former case, we have the result by applying the IH. The latter case is trivial.
 - If $M' = (M_1)M'_2$, $M_2 \rightarrow_{\mu\mu'\rho\varepsilon} M'_2$ and $(\lambda x^A.P^B)Q^A \leq M_1$ or $(\lambda x^A.P^B)Q^A \leq M'_2$, then either we have the result or we can apply the IH.
 - If $M' = (\lambda x^A.P^B)M_2^A$ and $M_1 \rightarrow_{\mu\mu'\rho\varepsilon} \lambda x.P$, then the only possible case is $M_1 = \lambda x.O$ and $O \rightarrow_{\mu\mu'\rho\varepsilon} P$, hence $M = (\lambda x^A.O^B)M_2^A$.
 - If $M' = (\lambda x^A.P^B)M_3^A$ and $M_2 \rightarrow_{\mu\mu'\rho\varepsilon} M_3$, then $M = (\lambda x^A.P^B)M_2^A$.
- If $M = (\mu\alpha^{-(A \rightarrow B)}M_3^\perp)M_2^A$ and $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.M_3[\alpha :=_r M_2] = M'$, then, by Lemma 6.8, we also have three subcases to examine.
 - The case where $(\lambda x^A.P^B)Q^A \leq M_2$ is immediate.
 - If $(\lambda x^A.P^B)Q^A = O[\alpha :=_r M_2]$ for some $O = (\lambda x^A.P_1^B)Q_1^A \leq M_3$, then $(\lambda x^A.P_1^B)Q_1^A \leq M$, and we have the result.
 - If $[\alpha]\lambda x^A.P_1^B \leq M_3$ and $(\lambda x^A.P_1[\alpha :=_r M_2]^B)M_2^A \leq M'$, then M_3 is α -good and the condition (A, B) holds for M .
- If $M = (M_1^{A \rightarrow B})\mu\alpha^{-A}.M_3^\perp$, $M \rightarrow_{\mu\mu'\rho\varepsilon} \mu\alpha.M_3[\alpha :=_l M_1] = M'$, then, by Lemma 6.8, we have the subcases below.
 - The case where $(\lambda x^A.P^B)Q^A \leq M_1$ is immediate.
 - If $(\lambda x^A.P^B)Q^A = O[\alpha :=_l M_1]$ for some $O = (\lambda x^A.P_1^B)Q_1^A \leq M_3$, then $(\lambda x^A.P_1^B)Q_1^A \leq M$, and we have the result.

- If $M_1 = \lambda x.P$, $[\alpha]Q_1 \leq M_3$ and $(\lambda x^A.P^B)Q_1[\alpha := M_1]^A \leq M'$, then $M = (\lambda x^A.P^B)\mu\alpha^{-A}.M_3^\perp$ and our assertion follows. □

We obtain the following corollaries about $\mu\mu'\rho\varepsilon$ -reduction sequences. Intuitively, the origin of a β -redex through a $\mu\mu'\rho\varepsilon$ -reduction sequence is a μ -redex of the same type.

Corollary 6.17. If $M \in \mathcal{NF}_\beta$, $M \rightarrow_{\mu\mu'\rho\varepsilon} N$ and $(\lambda x^A.P_1^B)Q_1^A \leq N$, then $(\mu\alpha^{-(A \rightarrow B)}.P_2^\perp)Q_2^A \leq M$ for some terms P_2, Q_2 .

Proof. By Lemma 6.16. □

6.3 Weak normalization

In this subsection, we state our main result (Theorem 6.24), which asserts that $\lambda\mu\mu'\rho\varepsilon\theta$ -reduction in the typed $\lambda\mu$ -calculus is weakly normalizing.

We define the rank of a β -redex first, and then the rank of a $\lambda\mu$ -term. We could also define the rank of all the other redexes (and, especially, that of a μ -redex), but we do not need this for our subsequent proofs.

Definition 6.18. Let M be a $\lambda\mu$ -term.

1. Let $r = (\lambda x^A.P^B)Q^A$ be a β -redex of M . The rank of r in M is defined by $\text{rank}(r, M) = \text{lh}(A \rightarrow B)$.
2. The rank of M is $\text{rank}(M) = \max\{\text{rank}(r, M) \mid r \text{ is a } \beta\text{-redex in } M\}$.

At this point, we can combine all of our previous results on the creation of redexes to prove that $\beta\mu\mu'\rho\varepsilon$ -reduction is weakly normalizing (Theorem 6.21). In the statements below, we implicitly make use of the facts that typed β -reduction is strongly normalizing (Krivine 1993) (in fact, weak normalization suffices here) and that $\mu\mu'\rho\varepsilon$ -reduction is weakly normalizing (Theorem 5.10).

Corollary 6.19. Let $M_1, M_3 \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$ and $M_2 \in \mathcal{NF}_\beta$ such that $M_3 \notin \mathcal{NF}_\beta$ and $M_1 \rightarrow_\beta M_2 \rightarrow_{\mu\mu'\rho\varepsilon} M_3$, then $\text{rank}(M_1) > \text{rank}(M_3)$.

Proof. Let $(\lambda x^A.P_1^B)Q_1^A$ be a β -redex of M_3 having maximal rank. By Corollary 6.17, $(\mu\alpha^{-(A \rightarrow B)}.P_2^\perp)Q_2^A \leq M_2$ and, by Corollary 6.7, $(\lambda y^C.P_1'^D)Q_1'^C \leq M_1$ where $\text{lh}(C \rightarrow D) > \text{lh}(A \rightarrow B)$. Hence $\text{rank}(M_1) > \text{rank}(M_3)$. □

Corollary 6.20. If $M \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$, then $M \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$.

Proof. By induction on $\text{rank}(M)$. If $M \notin \mathcal{NF}_\beta$, then $M \rightarrow_\beta M'$ and $M' \in \mathcal{NF}_\beta$ for some M' . If $M' \notin \mathcal{NF}_{\mu\mu'\rho\varepsilon}$, then $M' \rightarrow_{\mu\mu'\rho\varepsilon} M''$ and $M'' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$. Finally, if $M'' \notin \mathcal{NF}_\beta$, by Corollary 6.19, $\text{rank}(M) > \text{rank}(M'')$ and, by IH, $M'' \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$, then $M \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$. □

We are now in a position to prove our weak normalization result.

Theorem 6.21. In the typed $\lambda\mu$ -calculus, $\beta\mu\mu'\rho\varepsilon$ -reduction is weakly normalizing.

Proof. Let M be a $\lambda\mu$ -term and $M \rightarrow_{\mu\mu'\rho\varepsilon} M'$ where $M' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$. By Corollary 6.20, $M' \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$, hence $M \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$. \square

We can also extend our weak normalization algorithm for the θ -rule. First, we prove that θ -reduction creates no $\beta\mu\mu'\rho\varepsilon$ -redex when the starting $\lambda\mu$ -term is in $\beta\mu\mu'\rho\varepsilon$ -normal form. Hence, θ -reduction can be postponed.

Lemma 6.22. Let $M \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$. If $M \rightarrow_{\theta} M'$, then $M' \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$.

Proof. It is enough to examine the case of one-step reduction. Let $\mu\alpha.[\alpha]N$ be the θ -redex reduced in M . We observe that if $\mu\alpha.[\alpha]N < M$, then $\lambda x.\mu\alpha.[\alpha]N \preceq M$, and if $\lambda x.\mu\alpha.[\alpha]N < M$, then $[\beta]\lambda x.\mu\alpha.[\alpha]N \preceq M$ or $(Q)\lambda x.\mu\alpha.[\alpha]N \preceq M$, where $Q \neq \lambda y.Q'$ and $Q \neq \mu\gamma.Q'$. Thus, θ -reduction in M cannot create other redexes. \square

Lemma 6.23. Both in the typed and in the untyped $\lambda\mu$ -calculus, θ -reduction strongly normalizes.

Proof. We observe that θ -reduction decreases the size of the terms. \square

Theorem 6.24. In the typed $\lambda\mu$ -calculus, $\lambda\mu\mu'\rho\varepsilon\theta$ -reduction is weakly normalizing.

Proof. Follows from Theorem 6.21 and Lemmas 6.22 and 6.23. \square

7. Future Work

1. Apart from the fact that the μ' -rule destroys confluence, it has another flaw. It is the only rule (among the rules we have presented in this paper) that does not preserve second-order types (the classical counterparts of Girard's system \mathcal{F} and Krivine's system \mathcal{AF}_2). In Py (1998), Py has presented an example of a $\lambda\mu$ -term discovered by Raffalli which produces a $\lambda\mu$ -term of an arbitrary type starting from another $\lambda\mu$ -term of a different type. The problem is caused exclusively by the μ' -rule. It would be interesting to find conditions under which, on the one hand, the μ' -rule would preserve types, and, on the other hand, the uniqueness of the representation of integers and strong normalization would be retained.
2. In Nour and Ziadeh (2017), the second author and Ziadeh have presented a realizability semantics for the $\lambda\mu$ -calculus equipped only with β - and μ -reduction rules. They have obtained a completeness result for this semantics. This type of semantics is interesting since, on the one hand, it characterizes the algorithmic behavior of typed $\lambda\mu$ -terms and, on the other hand, it can be a tool for proving weak or strong normalization. Adding other reduction rules (even the simplest ones like ρ - or θ -rules) considerably complicates the study of these semantics. It would be interesting to find a good realizability semantics for the $\lambda\mu$ -calculus with all rules discussed in this paper.
3. The second author has published (Nour 2002) a calculus that encodes second order, classical natural deduction. This calculus also contains rules that permit the encoding of a parallel *or* operator. It would be interesting to study the μ' -rule in this calculus and prove its weak normalization.

Acknowledgements. We are grateful to T. Hirschowitz and F. Kamareddine, who helped us make the manuscript better, and we thank the referees for their thorough work and numerous comments which have greatly contributed to clarifying and improving the paper.

References

- Ariola, Z. M. and Herbelin, H. (2003). Minimal classical logic and control operators. In: Baeten, J. C. M., Lenstra, J. K., Parrow, J. and Woeginger, G. J. (eds.) *Proceedings of the 30th International Conference on Automata, Languages and Programming, ICALP'03*, 871–885.
- Barbanera, F. and Berardi, S. (1994). A symmetric lambda calculus for classical program extraction. In: Hagiya, M. and Mitchell, J. C. (eds.) *Proceedings of Theoretical Aspects of Computer Software, TACS'94*, Lecture Notes in Computer Science, vol. 789, Springer Verlag, 495–515.
- Battyányi, P. (2007). *Normalization Properties of Symmetric Logical Calculi*. Phd thesis, University of Chambéry.
- Battyányi, P. and Nour, K. (2020). Normalization proofs for the un-typed $\mu\mu'$ -calculus. Special Issue: *LICMA'19 Lebanese International Conference on Mathematics and Applications*, AIMS Mathematics, vol. 5(4), 3702–3713.
- Curien, P.-L. and Herbelin, H. (2000). The duality of computation. In: Odersky, M. and Wadler, P. (eds.) *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming, ICFP'00*, ACM Press, 233–243.
- David, R. and Nour, K. (2005). Arithmetical proofs of strong normalization results for the symmetric $\lambda\mu\mu'$ -calculus. In: Urzyczyn, P. (ed.) *Typed Lambda Calculi and Applications, TLCA'05*, Lecture Notes in Computer Science, vol. 3461, Springer Verlag, 162–178.
- David, R. and Nour, K. (2007). Arithmetical proofs of strong normalization results for symmetric lambda calculi. *Fundamenta Informaticae* 77 (4) 489–510.
- de Groote, P. (1998). An environment machine for the $\lambda\mu$ -calculus. *Mathematical Structures in Computer Science* 8 637–669.
- de Groote, P. (1994). On the relation between the $\lambda\mu$ -calculus and the syntactic theory of sequential control. In: Pfenning, F. (ed.) *5th International Conference on Logic Programming and Automated Reasoning, LPAR'94*, Lecture Notes in Artificial Intelligence, vol. 822, Springer Verlag, 31–43.
- Girard, J.-Y., Lafont, Y. and Taylor, P. (1989). *Proofs and Types*, Cambridge University Press, Cambridge, United Kingdom.
- Griffin, T. (1990). A formulae-as-type notion of control. In: Allen, F. E. (ed.) *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, POPL'90*, ACM Press.
- Krivine, J.-L. (1993). *Lambda-Calculus Types and Models*, Ellis Horwood Ltd., Harlow, United Kingdom.
- Krivine, J.-L. (1994). Classical logic, storage operators and 2nd order lambda-calculus. *Annals of Pure and Applied Logic* 68 53–78.
- Leivant, D. (1983). Reasoning about functional programs and complexity classes associated with type disciplines. In: *24th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, 460–469.
- Munch-Maccagnoni, G. (2013). *Syntax and Models of a Non-associative Composition of Programs and Proofs*. Phd thesis, Université Paris Diderot.
- Murthy, C. R. (1991). An evaluation semantics for classical proofs. In: *Proceedings of the Sixth Annual IEEE Symposium on Logic in Computer Science*, 96–107.
- Nour, K. (1997). La valeur d'un entier classique en $\lambda\mu$ -calcul. *Archive for Mathematical Logic* 36 461–473.
- Nour, K. (2002). A non-deterministic classical logic (the $\lambda\mu^{++}$ -calculus). *Mathematical Logic Quarterly* 48 357–366.
- Nour, K. and Saber, K. (2009). A completeness result for the simply typed lambda-mu calculus. *Annals of Pure and Applied Logic* 161 109–118.
- Nour, K. and Ziadeh, M. (2017). A revised completeness result for the simply typed $\lambda\mu$ -calculus using realizability semantics. *Logical Methods in Computer Science* 13(3:13) 1–13.
- Parigot, M. (1992). $\lambda\mu$ -calculus: an algorithmic interpretation of classical natural deduction. In: Voronkov, A. (ed.) *Logic Programming and Automated Reasoning*, Lecture Notes in Computer Science, vol. 624, Berlin, Springer Verlag, 190–201.
- Parigot, M. (1993). Classical proofs as programs. In: Gottlob, G., Leitsch, A. and Mundici, D. (eds.) *Proceedings of 3rd Kurt Godel Colloquium, KGC'93*, Lecture Notes in Computer Science, vol. 713, Springer-Verlag, 263–276.
- Parigot, M. (1997). Proofs of strong normalization for second order classical natural deduction. *Journal of Symbolic Logic* 62 1461–1479.
- Py, W. (1998). *Confluence en $\lambda\mu$ -calcul*. Phd thesis, University of Chambéry.
- Rehof, N. J. and Sørensen, M. H. (1994). The λ_{Δ} -calculus. In: Hagiya, M. and Mitchell, J. C. (eds.) *Theoretical Aspects of Computer Software*, Lecture Notes in Computer Science, vol. 789, Springer Verlag, 516–542.
- Saurin, A. (2008). On the relations between the syntactic theories of $\lambda\mu$ -calculi. In: Kaminski, M. and Martini, S. (eds.) *17th EACSL Annual Conference on Computer Science Logic*, Lecture Notes in Computer Science, vol. 5213, Springer Verlag, 154–168.
- Saurin, A. (2012). Böhm theorem and Böhm trees for the Lambda-mu-calculus. *Theoretical Computer Science* 435 106–138.
- Sørensen, M. H. and Urzyczyn, P. (2006). *Lectures on the Curry-Howard Isomorphism*, Elsevier Science & Technology, Oxford, United Kingdom.
- Wadler, P. (2003). Call-by-value is dual to call-by-name. In: Runciman, C. and Shivers, O. (eds.) *Proceedings of the Eighth ACM SIGPLAN International Conference on Functional Programming, ICFP'03*, 189–201.
- Wadler, P. (2005). Call-by-value is dual to call-by-name - reloaded. In: Giesl, J. (ed.) *Term Rewriting and Applications*, Lecture Notes in Computer Science, vol. 3467, Springer Verlag, 185–203.

Cite this article: Battyányi P. and Nour K (2023). Normalization in the simply typed $\lambda\mu\mu'\rho\theta\varepsilon$ -calculus. *Mathematical Structures in Computer Science*. <https://doi.org/10.1017/S096012952200041X>