



HAL
open science

Normalization properties of $\lambda\mu$ -calculus using realizability semantics

Péter Battyányi, Karim Nour

► **To cite this version:**

Péter Battyányi, Karim Nour. Normalization properties of $\lambda\mu$ -calculus using realizability semantics. 2023. hal-04740130

HAL Id: hal-04740130

<https://hal.science/hal-04740130v1>

Preprint submitted on 16 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.

Normalization properties of $\lambda\mu$ -calculus using realizability semantics

Péter Battyányi* and Karim Nour†

November 7, 2023

Abstract

In this paper, we present a general realizability semantics for the simply typed $\lambda\mu$ -calculus. Then, based on this semantics, we derive both weak and strong normalization results for two versions of the $\lambda\mu$ -calculus equipped with specific simplification rules. The novelty in our method, in addition to its more systematic approach, lies in its applicability to a broader set of reduction rules. Furthermore, we have obtained a surprising characterization of the typed $\lambda\mu$ -terms with the help of our saturation properties.

Keywords and phrases: $\lambda\mu$ -calculus ; classical logic ; strong normalization ; weak normalization ; realizability semantics ; saturated set.

Subject code classifications : 03B40 ; 03B70 ; 03F05 ; 68Q42.

1 Introduction

The Curry-Howard isomorphism [13] has revealed the connection between the simplification of abstract terms and the normalization of proofs in intuitionistic logical theories. This insight has proven to be one of the most fruitful ideas in logic and proof theory. It was understood in the 1990s that this correspondence can also be extended to the realm of classical logic [10, 15]. Several systems have been introduced to account for this correspondence, among which the most well-known are probably Parigot's $\lambda\mu$ -calculus [18], Curien and Herbelin's $\bar{\lambda}\mu\tilde{\mu}$ -calculus [5], Barbanera and Berardi's λ^{Sym} -calculus [1] and Rehof and Sørensen's $\lambda\Delta$ -calculus [23]. All of these systems enjoy the strong normalization property [20, 21, 1, 23]. All of the mentioned proofs rely on the well-known reducibility technique or, in other words, the method of realizability semantics credited to Girard and Tait [8]. Later on, several other proofs making use of "combinatorial" arguments emerged for the propositional parts of these calculi [6, 7, 2, 3].

Initially, Parigot's system consisted of the β - and μ -reductions, along with some simplifications rules called ρ and θ . The calculus is confluent [19, 22], and possesses the subject reduction and strong normalization properties [19]. However, from a practical standpoint, it has a major drawback, namely, the uniqueness of data representation is not valid. There are normal terms of integer type that are not

*Department of Computer Science, Faculty of Informatics, University of Debrecen, Kassai út 26, 4028 Debrecen, Hungary, battyanyi.peter@inf.unideb.hu

†Université Savoie Mont Blanc, CNRS, LAMA, LIMD, 73000 Chambéry, France, karim.nour@univ-smb.fr

Church numerals. To address this issue, in this paper, we adopt one of the solutions proposed by Parigot [19]: we introduce a new rule μ' , which is the symmetric counterpart of μ and allows us to further simplify a Church numeral, and thus getting closer to the form of a λ -calculus Church numeral. To obtain the value of an integer-typed term in the $\lambda\mu$ -calculus, i.e., the usual form of a Church numeral in the λ -calculus, we require additional simplification rules, namely, the ρ , ε and θ -rules [4].

The calculus equipped with the μ' -rule has the subject reduction property in the simply typed version, but it lacks the Church-Rosser property. However, the μ' -rule plays a crucial role in the presentation of the call-by-value $\lambda\mu$ -calculus. It allows a calculated value, acting as the functional part, to find its argument. De Groote [12] successfully integrated μ' -reduction with the $\lambda\mu$ -calculus while preserving confluence. Furthermore, Py studied a call-by-value μ' -reduction in his thesis [22]. In particular, he demonstrated confluence and strong normalization properties for both μ - and μ' -reductions. He also enriched the syntax with new operators, ensuring that the subject reduction property was maintained, and providing computational content to the quantifiers.

Nevertheless, μ' -reduction does not preserve types in a type system based on second-order logic. Raffalli gave an example of a term of type A that can be reduced (using the rule μ') to any term of type B . His example was published in Py's thesis [22]. A solution to this problem, as proposed by Py, can involve giving algorithmic content to the rules associated with the quantifier \forall [22]. As part of our future work, we plan to develop a realizability semantics that will enable us to establish normalization results within this framework.

The calculus augmented with the μ' -rule does not possess the Church-Rosser property. However, the μ' -rule plays a crucial role in the presentation of the call-by-value $\lambda\mu$ -calculus. It allows a calculated value, acting as the functional part, to find its argument. De Groote [12] successfully integrated this rule with the $\lambda\mu$ -calculus while maintaining confluence. Py studied the call-by-value μ' -reduction in his thesis [22]. In particular, he demonstrated confluence and strong normalization properties for both μ - and μ' -reductions. He also enriched the syntax with new operators, ensuring that the subject reduction property was maintained, and providing computational content to the quantifiers.

The Parigot-style $\lambda\mu$ -calculus, as originally defined, imposes severe restrictions on the term formation rules. It introduces two sets of variables, the so-called intuitionistic, or β -variables, and the classical, or μ -variables. In addition to the usual λ -abstraction and application, two new constructs appear: μ -application, where the left hand side of the application is a μ -variable, and μ -abstraction, where a μ -prefix is formed with a μ -variable. Originally, these two are interconnected by Parigot: a μ -abstraction must always be followed by a μ -application in the term and, conversely, a μ -application must always be preceded by a μ -abstraction. However, it turned out, that this syntax did not allow encoding a proof of the classical tautology $\neg\neg A \rightarrow A$ without free variables of type $\neg\perp$ (similarly for some other classical tautologies). Furthermore, another drawback emerged: in this syntax with many reduction rules, Böhm's theorem does not hold. In his thesis, Py [22] found 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 $\lambda x.\lambda y.y$. To remedy this situation, De Groote proposed a more flexible syntax in which the restriction concerning μ -abstractions and μ -applications is resolved: he split the rules for negation into two, one standing for the introduction and one for the elimination of negation without imposing further restrictions on their applicability. The proposed version allowed him to construct an abstract machine for the $\lambda\mu$ -calculus [11]. Additionally, Saurin [25] proved that De Groote's version enjoys the separation property.

Regarding the normalization properties, it was shown by the authors that the Parigot-style simply typed $\lambda\mu$ -calculus augmented with the rules μ' , ρ , ε , θ preserves the strong normalization property [4]. However, the situation is completely different when we turn to De Groote's syntax. Although, it turns out that the simply typed $\lambda\mu$ -calculus in De Groote's formalism preserves strong normalization if we only consider $\beta\mu\mu'$ -reduction [6, 4]- and this is even true for the reduction $\mu\mu'$ both in the untyped and typed cases [6, 3]- if we add more rules to the existing ones, like ρ , ε , this is no longer true. Specifically, the $\mu\mu'\rho$ -reduction is not strongly normalizing and the $\mu\mu'\varepsilon$ -reduction is not weakly normalizing even in the typed case. It can be shown that weak normalization is retained both for the untyped $\mu\mu'\rho\varepsilon$ -reduction and for the typed $\beta\mu\mu'\rho\varepsilon$ -reduction [4].

However, the proofs of the results in [4] involve elaborate arguments and detailed analyses of the behaviour of residuals of redexes. In this paper, we present a semantical approach to these results. Proving normalization properties through semantical interpretations has a long tradition dating back to the works of Tait and Girard [27, 8]. In [20], Parigot proved the strong normalization property for the second-order $\lambda\mu$ -calculus equipped with the ρ - and θ -rules and without the μ' -rule. He followed a reasoning that was a generalization of the Tait-Girard reducibility method. Additionally, he formulated another proof for the strong normalization of second-order classical propositional logic by extending the Gödel-Kolmogorov translation to second order $\lambda\mu$ -calculus. In his paper [12], de Groote proved that the strong normalization property holds for his version of the second-order $\lambda\mu$ -calculus together with rules ρ , θ and ε . He applied the Tait-Girard reducibility method for the case of $\lambda\mu$ -reduction and then he demonstrated that the ρ -, θ and ε -rules can be postponed. Furthermore, Berardi and Barbanera also applied the reducibility method for their calculus; however, their proof is even more complicated [1]. Due to the presence of an involutive negation, i.e., an operator for which $A = A^{\perp\perp}$ holds, the authors resorted to a very technical fixed-point operator to obtain the candidates of reducibility for the terms.

In contrast to the papers mentioned above, our proof of the weak and strong normalization properties is more centered around finding appropriate definitions of reducibility candidates, by which the strong normalization of $\beta\mu\rho\theta\varepsilon$ -reduction and the weak normalization of $\beta\mu\mu'\rho\theta\varepsilon$ -reduction follow in a relatively straightforward manner. We emphasize that $\beta\mu\mu'\rho\theta\varepsilon$ -reduction is not strongly normalizing [4] so the strongest result we can achieve in this context is the weak normalization property.

The paper is structured as follows: the next section is devoted to the relevant notions and already known results concerning the simply-typed $\lambda\mu$ -calculus. In Section 3, we develop the tools needed for the subsequent sections: we define a notion of saturation with the help of which we introduce the reducibility candidates for the typed calculus. We conclude this section with the correctness theorem, which asserts that our interpretation adheres to typability rules. Moreover, in contrast to most of the realizability notions that have appeared in the literature thus far, our definition of saturation yields the surprising result that the set of typable terms is not only contained within the collection of saturated sets, but the two sets coincide. Therefore, when we intend to prove a certain property for the typable terms, it is enough to prove that the set of typable terms possessing that property is saturated. By our result, it then follows that every typable term enjoys that property.

The novelty in our approach is that, instead of trying to prove that the auxiliary rules can be postponed, which would not even be the case in the presence of μ' , we modify the notion of saturation so that the verification of the necessary properties can be guaranteed with respect to all of the rules of $\beta\mu\mu'\rho\theta\varepsilon$ -reduction in a single process. We do not need permutation lemmas to achieve our goal when we consider the $\lambda\mu$ -calculus with all of the mentioned rules.

In Section 4, we present a proof of the strong normalization of the simply-typed

$\lambda\mu$ -calculus equipped with $\beta\mu\rho\theta\varepsilon$ -reduction. In Section 5, we take a step further and augment the calculus under consideration with the μ' -rule. We demonstrate that the set of terms having a normal form is saturated, which implies that the $\beta\mu\mu'\rho\theta\varepsilon$ -reduction is weakly normalizing. We then conclude with a discussion on future work.

2 The $\lambda\mu$ -calculus

The $\lambda\mu$ -calculus was introduced by Parigot [18] as a tool for encoding classical natural deduction with terms. In our paper, we restrict our attention to the simply typed calculus, specifically, we are concerned with the representation of the implicational fragment of classical propositional logic using natural deduction style proofs.

The $\lambda\mu$ -calculus involves two sets of variables: one set consists of the original λ -variables, often referred to as intuitionistic variables, while the other set comprises μ -variables, known as classical variables. Parigot's calculus imposes certain constraints on the rules for forming terms. Notably, a μ -abstraction must be immediately followed by an application involving a μ -variable, and vice versa. However, de Groote [11] introduced a modified version of the $\lambda\mu$ -calculus, which offers more flexibility in its syntax. In de Groote's syntax, a μ -abstraction can be followed by an arbitrary term (in the untyped version), not exclusively by a μ -application. For the purpose of this paper, we exclusively focus on the de Groote-style calculus, and the following definitions pertain only to this version.

Definition 2.1 ($\lambda\mu$ -terms)

1. Let $\mathcal{V}_\lambda = \{x, y, z, \dots\}$ denote the set of λ -variables and $\mathcal{V}_\mu = \{\alpha, \beta, \gamma, \dots\}$ denote the 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 $(M N)$ if we apply M to N .

2. In a $\lambda\mu$ -term the λ and μ operators bind the variables. 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 $\text{fv}(\mu\alpha.M) = \text{fv}(M) \setminus \{\alpha\}$.

Definition 2.2 (Type system) The types are built from a set \mathcal{V}_τ of atomic types and the constant \perp with the connective \rightarrow . The type formation rules are the following.

$$\mathfrak{T} := \mathcal{V}_\tau \cup \{\perp\} \mid \mathfrak{T} \rightarrow \mathfrak{T}$$

In the definition below, Γ (resp. Δ) denotes a (possibly empty) context, that is, a finite set of declarations of the form $x : A$ (resp. $\alpha : B$) for a λ -variable x (resp. a μ -variable α) and types A, B such that a λ -variable x (resp. a μ -variable α) occurs at most once in an expression $x : A$ (resp. $\alpha : A$) of Γ (resp. of Δ). The typing rules are as follows.

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

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

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

We say that the $\lambda\mu$ -term M is typable with type A , if there is Γ, Δ and a derivation tree such that the uppermost nodes of the tree are axioms and the bottom node is $\Gamma \vdash M : A, \Delta$. The presentation with two contexts (one on the left for the λ -variables and one on the right for the μ -variables) allows us to avoid using negation in types. However, when we correlate typability proofs with proofs in classical natural deduction, the context standing on the right corresponds to a set of hypothesis consisting of negative formulas. Alternatively, we could have applied two contexts on the left: one for the λ -variables and one for the μ -variables, represented by negated types.

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 the realm of $\lambda\mu$ -terms, various reduction rules exist, with some being fundamental and corresponding to the elimination of logical cuts, while others are introduced to eliminate specific subterms and are often referred to as simplification rules. In this article, we focus on the rules essential for achieving a reasonable representation of data, similar to the case of the λ -calculus. For a more comprehensive understanding of these rules, readers are encouraged to consult [4], where we demonstrate that Church numerals are the only normal $\lambda\mu$ -terms that have the type of integers provided we consider the additional rules μ', ρ, ε and θ .

Throughout this paper, we employ various types of substitutions. These substitutions include necessary variable renaming to prevent the capture of free variables. Below, we present the concept of μ -substitution, which plays a crucial role in defining μ -redex, as introduced in Definition 2.5.

Definition 2.3 (μ -substitution)

1. A μ -substitution σ is an expression of the form $[\alpha :=_s N]$ where $s \in \{l, r\}$, $\alpha \in \mathcal{V}_\mu$ and $N \in \mathcal{T}$.
2. Let σ be the μ -substitution $[\alpha :=_s N]$ and $M \in \mathcal{T}$. 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 2.4 (α -translation) *Let $M \in \mathcal{T}$ and $\alpha \in \mathcal{V}_\mu$. 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 .

We proceed by defining the specific redexes that are the focus of this paper, along with the reductions they induce. Additionally, we provide a brief overview of some of the important results in relation to the $\lambda\mu$ -calculus. For a more in-depth understanding of the requisite concepts and definitions, interested readers are referred to the standard textbooks, such as [9] and [14].

Definition 2.5 (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 outlined in Definition 2.5 find their justification in the cut elimination rules of natural deduction within classical propositional logic. As a consequence of these reductions, the calculus exhibits a type preservation property, as demonstrated in Theorem 2.7, which is also known as the subject reduction property.

Definition 2.6 (Reduction and normalization) *Let $\mathcal{R} \subseteq \{\beta, \mu, \mu', \rho, \theta, \varepsilon\}$.*

1. *Let $M, M' \in \mathcal{T}$. 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).*

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

2. We denote by $\rightarrow_{\mathcal{R}}$ the reflexive, transitive closure of $\rightarrow_{\mathcal{R}}$. I.e., $M \rightarrow_{\mathcal{R}} M'$ iff $M \rightarrow_{\mathcal{R}} M_1 \rightarrow_{\mathcal{R}} M_2 \rightarrow \dots \rightarrow_{\mathcal{R}} M_k = M'$.
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 $M \rightarrow_{\mathcal{R}} M'$. 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.

We list here the most important results that we will use in our paper. For the missing proofs, the reader is referred to [3, 4, 6, 22].

Theorem 2.7 *The $\beta\mu\mu'\rho\theta\varepsilon$ -reduction preserves types, that is, if $\Gamma \vdash M : A; \Delta$ and $M \rightarrow_{\beta\mu\mu'\rho\theta\varepsilon} M'$, then $\Gamma \vdash M' : A; \Delta$.*

Theorem 2.8 *The $\mu\mu'$ -reduction is strongly normalizing for the untyped $\lambda\mu$ -terms.*

Theorem 2.9 *The $\beta\mu\mu'$ -reduction is strongly normalizing for the typed $\lambda\mu$ -terms.*

Theorem 2.10 *The $\mu\mu'\rho\theta\varepsilon$ -reduction is weakly normalizing for the untyped $\lambda\mu$ -terms.*

Theorem 2.11 *The $\beta\mu\mu'\rho\theta\varepsilon$ -reduction is weakly normalizing for the typed $\lambda\mu$ -terms.*

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

3 The semantics of the system

In this section, we define reducibility candidates for the $\lambda\mu$ -calculus and prove its correctness. First and foremost, we clarify what we mean by a saturated set. We then identify a particular set, denoted as \mathbb{T} , which we choose to be saturated. Additionally, we define the properties required for a set to be saturated with respect to \mathbb{T} , referred to as being \mathbb{T} -saturated.

Following this, we construct a model consisting of \mathbb{T} -saturated sets and interpret the types within this model. In previous works related to similar topics, a specific set of terms was typically fixed from the outset, such as the set of strongly normalizable terms. Subsequently, it was shown that this set satisfied the necessary properties. In contrast, our approach involves identifying the properties of the set \mathbb{T} that are sufficient to establish a correctness theorem. Once these properties are defined, we demonstrate that certain sets indeed possess these properties.

The conditions we impose on the set \mathbb{T} are primarily derived from the typing rules. Our aim is to ensure the fulfillment of the correctness theorem, as indicated by Theorem 3.15. To achieve this, we establish three groups of conditions related to saturation.

- The first group consists of three straightforward conditions that ensure saturation concerning the constructors λ , μ , and $[\cdot]$. These conditions contain essential assumptions to maintain the typability of terms. While the first condition does not pose complications in subsequent sections, the other two are more delicate and require special attention.
- The second group consists of a single condition that is crucial for ensuring that all variables are within the set \mathbb{T} . This condition is essential for managing non-closed terms (terms containing free variables) and preserving specific properties related to the constructor \rightsquigarrow , which interprets the logical connector \rightarrow .
- The final two conditions are relatively standard in the literature and are used in virtually all realizability semantics. They are crucial for managing the typing rules associated with logical introduction rules. It will become apparent later that these conditions, in particular, require that reductions must contain at least the head reduction rules corresponding to β -reduction and μ -reduction.

Due to certain technical difficulties, we made the choice to demand that the set \mathbb{T} exclusively contains typed terms. This choice was found to be justified, as we were going to derive results that only made sense within a typed environment. Consequently, we exercised caution in all our definitions to include only typed terms within the set \mathbb{T} .

The following definition introduces essential concepts, and Definition 3.2 outlines the conditions under which a set is deemed saturated.

Definition 3.1 (Preliminary notions)

1. We denote the set of typable $\lambda\mu$ -terms \mathcal{T}_p , i.e.
 $\mathcal{T}_p = \{M \in \mathcal{T} / \exists \Gamma, \Delta, A, \text{ such that } \Gamma \vdash M : A; \Delta\}$.
2. If $\mathcal{L} \subseteq \mathcal{T}$, we denote by $\mathcal{L}^{<\omega}$ the set of finite (possibly empty, denoted by \emptyset) sequences of $\lambda\mu$ -terms of \mathcal{L} .
3. Let $\mathcal{L} \subseteq \mathcal{T}$. If $\bar{N} = N_1 \dots N_n \in \mathcal{L}^{<\omega}$, we write $\bar{P} \sqsubseteq \bar{N}$ if $\bar{P} = N_1 \dots N_k$ where $0 \leq k \leq n$. Intuitively, \bar{P} is an initial subsequence of the sequence \bar{N} .
4. Let $M \in \mathcal{T}$ and $\bar{P} \in \mathcal{T}^{<\omega}$, we define by induction the $\lambda\mu$ -term $(M)\bar{P}$: $(M)\emptyset = M$ and, if $\bar{P} = N\bar{Q}$, that is, \bar{P} is the sequence consisting of $N \in \mathcal{T}$ followed by $\bar{Q} \in \mathcal{T}^{<\omega}$, then $(M)\bar{P} = ((M)N)\bar{Q}$.

Definition 3.2 (Saturated sets) Let $\mathbb{T} \subseteq \mathcal{T}_p$. We say that \mathbb{T} is saturated if

- (C1) : $\forall M \in \mathbb{T}, \forall x \in \mathcal{V}_\lambda, \lambda x.M \in \mathbb{T}$.
- (C2) : $\forall M \in \mathbb{T}, \forall \alpha \in \mathcal{V}_\mu, \text{ if } \mu\alpha.M \in \mathcal{T}_p, \text{ then } \mu\alpha.M \in \mathbb{T}$.
- (C3) : $\forall M \in \mathbb{T}, \forall \alpha \in \mathcal{V}_\mu, \text{ if } [\alpha]M \in \mathcal{T}_p, \text{ then } [\alpha]M \in \mathbb{T}$.
- (C4) : $\forall n \geq 0, \forall N_1, \dots, N_n \in \mathbb{T}, \forall x \in \mathcal{V}_\lambda, \text{ if } (x)N_1 \dots N_n \in \mathcal{T}_p, \text{ then } (x)N_1 \dots N_n \in \mathbb{T}$.

(C5) : $\forall M, N \in \mathcal{T}_p, \forall \bar{P} \in \mathcal{T}_p^{<\omega}, \text{ if } N \in \mathbb{T}, (\lambda x.M)N\bar{P} \in \mathcal{T}_p \text{ and } (M[x := N])\bar{P} \in \mathbb{T},$
then $(\lambda x.M)N\bar{P} \in \mathbb{T}$.

(C6) : $\forall M \in \mathcal{T}_p, N \in \mathcal{T}_p^{<\omega}, \text{ if } \bar{N} \in \mathbb{T}^{<\omega}, (\mu\alpha.M)\bar{N} \in \mathcal{T}_p \text{ and } \mu\alpha.M[\alpha :=_r \bar{N}] \in \mathbb{T},$
then $(\mu\alpha.M)\bar{N} \in \mathbb{T}$.

Remark 3.3

1. The conditions (C1), (C2) and (C3) describe the requirements for saturation concerning λ -, μ -abstraction and μ -application. We note that we must pay special attention to ensuring that we stay within the realm of typable terms. Consequently, additional conditions are needed to guarantee typability.
2. Condition (C4) implies that \mathbb{T} contains λ -variables, which will prove to be essential for demonstrating certain closure properties of \mathbb{T} .
3. Condition (C5) states that \mathbb{T} is saturated with respect to head β -reduction when the original term is an application.
4. Condition (C6) is basically the same but concerns head μ -reduction. It is worth noting that a somewhat different condition can be formulated in this case since μ does not disappear during the reduction.
5. As we proceed, we will see that the stability properties of \mathbb{T} outlined in the above definition are indispensable to the correctness theorem.

Lemma 3.4 *The set \mathcal{T}_p is saturated.*

Proof We can observe that the conditions of Definition 3.2 are trivially verified. \square

In this section, we are going to prove a surprising result which is the following:

The set \mathcal{T}_p is the unique saturated set.

We will see in section 4 and 5 some applications of this powerful result.

Now, our goal is to define type interpretations by means of particular subsets of \mathbb{T} . To achieve this, it is essential to define our base sets that will serve as the images of type constants, or at the very least, to establish their desired properties. Additionally, we need to introduce an operation, denoted as \rightsquigarrow , for interpreting the arrow \rightarrow in types. However, we must exercise caution in our constructions to ensure that we remain within the domain of typed terms.

The definition of \rightsquigarrow provided below is quite standard and aligns with a common concept in realizability semantics: the terms in $\mathcal{K} \rightsquigarrow \mathcal{L}$ are those which map the terms of the set \mathcal{K} into the set \mathcal{L} . Furthermore, we need to define an implication operator for subsets of finite sequences of terms and sets of terms. As we delve into the proof of Theorem 3.15, it will become apparent that the second interpretation of \rightsquigarrow is essential for the correctness of the reducibility semantics within classical logic.

Following the definition of the \rightsquigarrow operator, we outline the conditions imposed on subsets of \mathbb{T} in Definition 3.6. These conditions are necessary for interpreting types and ensuring that the underlying subsets of \mathbb{T} remain saturated sets.

Definition 3.5 (The \rightsquigarrow operator) *Let \mathbb{T} be a saturated set and consider $\mathcal{K}, \mathcal{L} \subseteq \mathbb{T}$, and $\mathcal{X} \subseteq \mathbb{T}^{<\omega}$. We define two new subsets of \mathbb{T} relative to \mathbb{T} :*

$$\mathcal{K} \rightsquigarrow_{\mathbb{T}} \mathcal{L} = \{M \in \mathbb{T} / \forall N \in \mathcal{K}, \text{ if } (M)N \in \mathcal{T}_p, \text{ then } (M)N \in \mathcal{L}\},$$

$$\mathcal{X} \rightsquigarrow_{\mathbb{T}} \mathbb{T} = \{M \in \mathbb{T} / \forall \bar{N} \in \mathcal{X}, \forall \bar{P} \sqsubseteq \bar{N}, \text{ if } (M)\bar{P} \in \mathcal{T}_p, \text{ then } (M)\bar{P} \in \mathbb{T}\}.$$

Definition 3.6 (\mathbb{T} -saturated sets) *Let \mathbb{T} be saturated and $\mathcal{S} \subseteq \mathbb{T}$. We call \mathcal{S} a \mathbb{T} -saturated set or, in short, \mathbb{T} -saturated if*

(D1) : $\forall M, N \in \mathcal{T}_p, \forall \bar{P} \in \mathcal{T}_p^{<\omega}$, if $N \in \mathbb{T}$, $(\lambda x.M)N\bar{P} \in \mathcal{T}_p$ and $(M[x := N])\bar{P} \in \mathcal{S}$, then $(\lambda x.M)N\bar{P} \in \mathcal{S}$.

(D2) : $\forall n \geq 0, \forall N_1, \dots, N_n \in \mathbb{T}, \forall x \in \mathcal{V}_\lambda$, if $(x)N_1 \dots N_n \in \mathcal{T}_p$, then $(x)N_1 \dots N_n \in \mathcal{S}$.

(D3) : $\exists \mathcal{X}_\mathcal{S} \subseteq \mathbb{T}^{<\omega}, \mathcal{S} = \mathcal{X}_\mathcal{S} \rightsquigarrow_{\mathbb{T}} \mathbb{T}$.

Remark 3.7

1. Condition (D1) indicates that \mathcal{S} must be saturated concerning head β -reduction when the original term is an application. The requirement that $N \in \mathbb{T}$ is essential for this condition. However, we do not need a similar property for saturation with respect to head μ -reduction.
2. Condition (D2) implies that, specifically, every λ -variable must belong to \mathcal{S} .
3. Condition (D3) plays a crucial role in allowing us to formulate the correctness theorem. It enables us to transition from an interpretation of a proof of contradiction based on the assumption $\neg A$ to an interpretation of a proof of A .
4. Finally, if \mathbb{T} is a saturated set, it is also considered \mathbb{T} -saturated since $\mathbb{T} = \emptyset \rightsquigarrow_{\mathbb{T}} \mathbb{T}$.

The following remark can be seen as an easy exercise and will not be used in the rest of the paper.

Remark 3.8 Let \mathcal{K}, \mathcal{L} be \mathbb{T} -saturated, $M \in \mathcal{L}$ and $x \in \mathcal{V}_\lambda$ such that x is not free in M , then, by (C1), $\lambda x.M \in \mathbb{T}$. We have $\forall N \in \mathcal{K}$, if $(\lambda x.M)N \in \mathcal{T}_p$, then $M[x := N] = M \in \mathcal{L}$, thus, by (D1), $(\lambda x.M)N \in \mathcal{L}$. Therefore $\lambda x.M \in \mathcal{K} \rightsquigarrow_{\mathbb{T}} \mathcal{L}$. This shows that we can create an infinite number of terms in $\mathcal{K} \rightsquigarrow_{\mathbb{T}} \mathcal{L}$. We will see in Lemma 3.10 that $\mathcal{K} \rightsquigarrow_{\mathbb{T}} \mathcal{L}$ contains every λ -variable.

In the sequel, we omit the subscript \mathbb{T} and we simply write $\mathcal{K} \rightsquigarrow \mathcal{L}$ or $\mathcal{X} \rightsquigarrow \mathbb{T}$, respectively.

We can now introduce the concept of a \mathbb{T} -model, which comprises the sets used for interpreting types. Lemma 3.10 is of paramount importance as it guarantees that the sets comprising a model maintain their property of being \mathbb{T} -saturated.

Definition 3.9 (\mathbb{T} -model) *Let \mathbb{T} be saturated. A \mathbb{T} -model \mathcal{M} is the smallest set defined by the following inductive steps.*

1. $\mathbb{T} \in \mathcal{M}$.
2. Let $(\mathcal{S}_i)_{i \in I}$ be \mathbb{T} -saturated sets. Then $\mathcal{S}_i \in \mathcal{M}$ ($i \in I$).
3. Let $\mathcal{U}, \mathcal{V} \in \mathcal{M}$. Then $\mathcal{U} \rightsquigarrow \mathcal{V} \in \mathcal{M}$.

In other words, let \mathcal{S}_i ($i \in I$) be \mathbb{T} -saturated sets. Then \mathcal{M} is a \mathbb{T} -model if \mathcal{M} is the smallest set containing \mathcal{S}_i ($i \in I$), \mathbb{T} and is closed under the constructor \rightsquigarrow .

It will turn out (notably in Corollary 3.16) that, for the correctness theorem to hold, it would be enough to consider models containing only the set \mathbb{T} . At this point, however, we do not know yet that \mathcal{T}_p is the unique saturated set.

Lemma 3.10 *Let \mathcal{M} be a \mathbb{T} -model. If $\mathcal{S} \in \mathcal{M}$, then \mathcal{S} is \mathbb{T} -saturated.*

Proof Let $\mathcal{S} \in \mathcal{M}$, where \mathcal{M} is a \mathbb{T} -model. We verify the three points of Definition 3.6 one by one by induction on the minimal number of steps \rightsquigarrow in construction of \mathcal{S} . If $\mathcal{S} = \mathbb{T}$ or $\mathcal{S} = \mathcal{S}_i$, then the statement clearly holds. Assume $\mathcal{S} = \mathcal{U} \rightsquigarrow \mathcal{V}$ for some $\mathcal{U}, \mathcal{V} \in \mathcal{M}$. First of all, by Definition 3.5, we have $\mathcal{S} \subseteq \mathbb{T}$.

- (D1) Let $M, N \in \mathcal{T}_p$ and $\bar{P} \in \mathcal{T}_p^{<\omega}$ such that $N \in \mathbb{T}$, $(\lambda x.M)N\bar{P} \in \mathcal{T}_p$ and $(M[x := N])\bar{P} \in \mathcal{S}$. We will prove that $(\lambda x.M)N\bar{P} \in \mathcal{U} \rightsquigarrow \mathcal{V}$. First, we have $(M[x := N])\bar{P} \in \mathcal{S} \subseteq \mathbb{T}$ (by virtue of Definition 3.5), then, by (C5), $(\lambda x.M)N\bar{P} \in \mathbb{T}$. Let $Q \in \mathcal{U}$ such that $(\lambda x.M)N\bar{P}Q \in \mathcal{T}_p$, then $(M[x := N])\bar{P}Q \in \mathcal{T}_p$. Since we have $(M[x := N])\bar{P} \in \mathbb{T}$ and $(M[x := N])\bar{P} \in \mathcal{U} \rightsquigarrow \mathcal{V}$, thus $(M[x := N])\bar{P}Q \in \mathcal{V}$ and, by IH, $(\lambda x.M)N\bar{P}Q \in \mathcal{V}$. Therefore $(\lambda x.M)N\bar{P} \in \mathcal{S}$.
- (D2) Let $N_1, \dots, N_n \in \mathbb{T}$ and $x \in \mathcal{V}_\lambda$ such that $(x)N_1 \dots N_n \in \mathcal{T}_p$. We will prove that $(x)N_1 \dots N_n \in \mathcal{U} \rightsquigarrow \mathcal{V}$. First, by (C4), $(x)N_1 \dots N_n \in \mathbb{T}$. Assume $N \in \mathcal{U}$ and $(x)N_1 \dots N_n N \in \mathcal{T}_p$, then, by IH, $N \in \mathbb{T}$ and $(x)N_1 \dots N_n N \in \mathcal{V}$. Therefore $(x)N_1 \dots N_n \in \mathcal{S}$.
- (D3) Let $\mathcal{X}_\mathcal{V} \subseteq \mathbb{T}^{<\omega}$ such that $\mathcal{V} = \mathcal{X}_\mathcal{V} \rightsquigarrow \mathbb{T}$. By IH such an $\mathcal{X}_\mathcal{V}$ exists. Let $\mathcal{UX}_\mathcal{V} = \{N\bar{N} / N \in \mathcal{U} \text{ and } \bar{N} \in \mathcal{X}_\mathcal{V}\}$. We will verify $\mathcal{S} = \mathcal{UX}_\mathcal{V} \rightsquigarrow \mathbb{T}$. First, we have $\mathcal{UX}_\mathcal{V} \subseteq \mathbb{T}^{<\omega}$.

- Let $M \in \mathcal{S}$ and $\bar{P} = N\bar{Q}$ where $N \in \mathcal{U}$ and $\bar{Q} \in \mathcal{X}_\mathcal{V}$. Let $\bar{R} \sqsubseteq \bar{P}$ such that $(M)\bar{R} \in \mathcal{T}_p$. If $\bar{R} = \emptyset$, then $(M)\bar{R} = M \in \mathbb{T}$. If not, $\bar{R} = N\bar{S}$ where $\bar{S} \sqsubseteq \bar{Q}$. Since $(M)N \in \mathcal{T}_p$, $(M)N \in \mathcal{V} \subseteq \mathbb{T}$ and since $(M)N\bar{S} \in \mathcal{T}_p$ and $\bar{S} \sqsubseteq \bar{Q} \in \mathcal{X}_\mathcal{V}$, $(M)N\bar{S} \in \mathbb{T}$. Therefore $\mathcal{S} \subseteq \mathcal{UX}_\mathcal{V} \rightsquigarrow \mathbb{T}$.
- Let $M \in \mathcal{UX}_\mathcal{V} \rightsquigarrow \mathbb{T}$ and $N \in \mathcal{U}$ such that $(M)N \in \mathcal{T}_p$, then $(M)N \in \mathbb{T}$, since $N \sqsubseteq N\bar{Q} \in \mathcal{UX}_\mathcal{V}$. Let $\bar{P} \sqsubseteq \bar{N} \in \mathcal{X}_\mathcal{V}$ such that $(M)N\bar{P} \in \mathcal{T}_p$, then $N\bar{P} \sqsubseteq N\bar{N}$, thus $(M)N\bar{P} \in \mathbb{T}$. Hence, $(M)N \in \mathcal{X}_\mathcal{V} \rightsquigarrow \mathbb{T} = \mathcal{V}$. Therefore $\mathcal{UX}_\mathcal{V} \rightsquigarrow \mathbb{T} \subseteq \mathcal{S}$.

□

We have now reached a pivotal concept in grasping the essence of this kind of reducibility candidates. When attempting to model the equation $A = \neg\neg A$, rather than relying on fixed-point operators, we embrace the realizability semantics proposed by Parigot. Specifically, he addressed the challenge of deriving the interpretation of A from that of $\neg\neg A$ by ensuring that every element \mathcal{S} within \mathcal{M} can be represented as $\mathcal{S}^\perp \rightsquigarrow \mathbb{T}$ for some set \mathcal{S}^\perp . Intuitively, we can consider \mathcal{S}^\perp , referred to as the orthogonal of \mathcal{S} , as the interpretation of the negation of a type. This concept enables us to effectively incorporate classical logic without the need for fixed-point operators.

For the properties we aim to establish, it is vital that the empty list be included in the orthogonal set of the interpretation of each type. Therefore, we incorporate it into \mathcal{S}^\perp in the subsequent definition.

Definition 3.11 (Weakest precondition) *Let \mathcal{M} be a \mathbb{T} -model, and $\mathcal{S} \in \mathcal{M}$. We write*

$$\mathcal{S}^\perp = \left(\bigcup \{ \mathcal{X} \subseteq \mathbb{T}^{<\omega} / \mathcal{S} = \mathcal{X} \rightsquigarrow \mathbb{T} \} \right) \cup \{ \emptyset \}.$$

The next lemma demonstrates that $\mathcal{S}^\perp \rightsquigarrow \mathbb{T}$ coincides with \mathcal{S} , hence, \mathcal{S}^\perp does indeed describe the weakest condition under which can ensure this property.

Lemma 3.12 *Let \mathcal{M} be a \mathbb{T} -model, and $\mathcal{S} \in \mathcal{M}$. We have $\mathcal{S} = \mathcal{S}^\perp \rightsquigarrow \mathbb{T}$.*

Proof

- Assume $M \in \mathcal{S}$ and $\bar{P} \sqsubseteq \bar{N} \in \mathcal{S}^\perp$ such that $(M)\bar{P} \in \mathcal{T}_p$.
 - If $\bar{N} = \emptyset$, then $\bar{P} = \emptyset$ and $(M)\bar{P} = M \in \mathcal{S} \subseteq \mathbb{T}$.
 - If $\bar{N} \in \mathcal{X}_0$ for some \mathcal{X}_0 with $\mathcal{S} = \mathcal{X}_0 \rightsquigarrow \mathbb{T}$, we have, by definition, $(M)\bar{P} \in \mathbb{T}$.
- Let $\mathcal{X}_0 \subseteq \mathbb{T}^{<\omega}$ such that $\mathcal{S} = \mathcal{X}_0 \rightsquigarrow \mathbb{T}$. Since $\mathcal{X}_0 \subseteq \mathcal{S}^\perp$, then $\mathcal{S}^\perp \rightsquigarrow \mathbb{T} \subseteq \mathcal{X}_0 \rightsquigarrow \mathbb{T} = \mathcal{S}$. □

We can now interpret the types in a \mathbb{T} -model. Note that the interpretation of the type \perp will be the whole set \mathbb{T} .

Definition 3.13 (Interpretation of types) *Let \mathcal{M} be a \mathbb{T} -model. An \mathcal{M} -interpretation \mathcal{I} is a function $X \mapsto \mathcal{I}(X)$ from the set of atomic types \mathcal{V}_T to \mathcal{M} which we extend for any type formula as follows: $\mathcal{I}(\perp) = \mathbb{T}$ and $\mathcal{I}(A \rightarrow B) = \mathcal{I}(A) \rightsquigarrow \mathcal{I}(B)$.*

We are now prepared to state and prove the generalized correctness theorem. This theorem establishes a connection between the concept of typability and interpretation within a model. In simple terms, it asserts that any $\lambda\mu$ -term M of type A belongs to the interpretation of A in any model. Furthermore, given that the interpretation of any type is contained in \mathbb{T} , it follows that the $\lambda\mu$ -term M will also be in \mathbb{T} . Consequently, a well-chosen set \mathbb{T} enables us to uncover properties of typable terms in relation to a specific reduction.

Definition 3.14 (Simultaneous substitution) *Let $M, M_1, \dots, M_n \in \mathcal{T}$, $\bar{N}_1, \dots, \bar{N}_m \in \mathcal{T}^{<\omega}$, and σ be the simultaneous substitution $[(x_i := M_i)_{1 \leq i \leq n}; (\alpha_j :=_r \bar{N}_j)_{1 \leq j \leq m}]$, which is not an object of the syntax. Then $M\sigma$ is obtained from the $\lambda\mu$ -term M by replacing each x_i by M_i and replacing inductively each subterm of the form $[\alpha_j]U$ in M by $[\alpha_j](U)\bar{N}_j$. Here is a formal definition of $M\sigma$*

- If $M = x$ and $x \neq x_i$, then, $M\sigma = x$.
- If $M = x_i$, then $M\sigma = M_i$.
- If $M = \lambda x.U$, then we can assume that x is a new λ -variable and $M\sigma = \lambda x.U\sigma$.
- If $M = (U)V$, then $M\sigma = (U\sigma)V\sigma$.
- If $M = \mu\alpha.U$, then we can assume that α is a new μ -variable and $M\sigma = \mu\alpha.U\sigma$.
- If $M = [\alpha]U$ and $\alpha \neq \alpha_j$, then $M\sigma = [\alpha]U\sigma$.
- If $M = [\alpha_j]U$, then $M\sigma = [\alpha_j](U\sigma)\bar{N}_j$.

It is worth noting that the statement of the following theorem comes with a set of substantial assumptions. Notably, we incorporate initial subsequences of sequences into $\mathcal{I}(B_j)^\perp$, and we also require that $M\sigma$ be typable. A thorough examination of the proof of the theorem reveals that these assumptions are indispensable for the current approach.

Theorem 3.15 (General correctness theorem)

Let \mathcal{M} be a \mathbb{T} -model, \mathcal{I} an \mathcal{M} -interpretation,

$\Gamma = \{x_i : A_i\}_{1 \leq i \leq n}$, $\Delta = \{\alpha_j : B_j\}_{1 \leq j \leq m}$,

$M_i \in \mathcal{I}(A_i)$ for all $1 \leq i \leq n$, $\bar{P}_j \sqsubseteq \bar{N}_j \in (\mathcal{I}(B_j))^\perp$ for all $1 \leq j \leq m$ and

$\sigma = [(x_i := M_i)_{1 \leq i \leq n}; (\alpha_j :=_r \bar{P}_j)_{1 \leq j \leq m}]$.

If $\Gamma \vdash M : A$; Δ and $M\sigma \in \mathcal{T}_p$, then $M\sigma \in \mathcal{I}(A)$.

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

ax : In this case, $M = x_i$, $A = A_i$, $\Gamma \vdash x_i : A_i$; Δ and $M\sigma = \sigma(x_i) = M_i \in \mathcal{T}_p$.
By hypothesis, $M = M_i \in \mathcal{I}(A_i) = \mathcal{I}(A)$.

\rightarrow_i : In this case, $M = \lambda x.N$ and $A = B \rightarrow C$, $\Gamma, x : B \vdash N : C$; Δ and $M\sigma = \lambda x.N\sigma \in \mathcal{T}_p$. Since $x \in \mathcal{I}(B)$ and $N\sigma = N\sigma'$ where $\sigma' = \sigma + [x := x]$, we obtain, by IH, $N\sigma \in \mathcal{I}(C) \subseteq \mathbb{T}$, then (C1) involves $M\sigma \in \mathbb{T}$. Let $P \in \mathcal{I}(B)$ such that $(M\sigma)P \in \mathcal{T}_p$. We have $(M\sigma)P \rightarrow_\beta N\sigma''$ where $\sigma'' = \sigma + [x := P]$, then $N\sigma'' \in \mathcal{T}_p$. By IH, $N\sigma'' \in \mathcal{I}(C)$ and $P \in \mathbb{T}$, then, by (D1), $(M\sigma)P \in \mathcal{I}(C)$. Therefore $M\sigma \in \mathcal{I}(B) \rightsquigarrow \mathcal{I}(C) = \mathcal{I}(A)$.

\rightarrow_e : In this case, $M = (N)P$, $\Gamma \vdash P : A$; Δ , $\Gamma \vdash N : B \rightarrow A$; Δ , and $M\sigma = (N\sigma)P\sigma \in \mathcal{T}_p$. Then $N\sigma, P\sigma \in \mathcal{T}_p$ and, by IH, $N\sigma \in \mathcal{I}(B) \rightsquigarrow \mathcal{I}(A)$ and $P\sigma \in \mathcal{I}(B)$. This means $M\sigma = (N\sigma)P\sigma \in \mathcal{I}(A)$.

\perp_i : In this case, $M = \mu\alpha.N$, $\Gamma \vdash N : \perp$; $\alpha : A, \Delta'$ and $M\sigma = \mu\alpha.N\sigma \in \mathcal{T}_p$. Since $\emptyset \in \mathcal{I}(A)^\perp$ and $N\sigma = N\sigma'$ where $\sigma' = \sigma + [\alpha :=_r \emptyset]$, we obtain, by IH, $N\sigma \in \mathcal{I}(\perp) = \mathbb{T}$. Let $\emptyset \neq \bar{P} \sqsubseteq \bar{N} \in \mathcal{I}(A)^\perp$ such that $(M\sigma)\bar{P} \in \mathcal{T}_p$. We have $(M\sigma)\bar{P} \rightarrow_\mu \mu\alpha.N\sigma''$ where $\sigma'' = \sigma + [\alpha :=_r \bar{P}]$, then $\mu\alpha.N\sigma'', N\sigma'' \in \mathcal{T}_p$. By IH, $N\sigma'' \in \mathbb{T} = \mathcal{I}(\perp)$ and, by (C2), $\mu\alpha.N\sigma'' = \mu\alpha.N\sigma[\alpha :=_r \bar{P}] \in \mathbb{T}$, hence, by (C6), $(M\sigma)\bar{P} \in \mathbb{T}$. Therefore $M\sigma = \mu\alpha.N\sigma \in \mathcal{I}(A)$.

\perp_e : In this case, $M = [\alpha_j]N$, $A = \perp$, $\Gamma \vdash N : B_j$; Δ and let $\bar{P}_j \sqsubseteq \bar{N}_j \in \mathcal{I}(B_j)^\perp$ be such that $M\sigma = [\alpha_j](N\sigma)\bar{P}_j \in \mathcal{T}_p$. Then $N\sigma, (N\sigma)\bar{P}_j \in \mathcal{T}_p$ and, by IH, $N\sigma \in \mathcal{I}(B_j)$ and, since $\bar{P}_j \sqsubseteq \bar{N}_j \in \mathcal{I}(B_j)^\perp$, then $(N\sigma)\bar{P}_j \in \mathbb{T}$. Therefore, by (C3), $M\sigma = [\alpha_j](N\sigma)\bar{P}_j \in \mathbb{T} = \mathcal{I}(\perp)$. □

We can now state and prove the main result of this section.

Corollary 3.16 *For every saturated set \mathbb{T} we have $\mathbb{T} = \mathcal{T}_p$, i.e., \mathcal{T}_p is the unique saturated set.*

Proof It suffices to check that if $\Gamma \vdash M : A, \Delta$ and \mathbb{T} is a saturated set, then $M \in \mathbb{T}$. Assume $\Gamma = \{x_i : A_i\}_{1 \leq i \leq n}$ and $\Delta = \{\alpha_j : B_j\}_{1 \leq j \leq m}$. Let \mathcal{M} be the \mathbb{T} -model containing only the set \mathbb{T} and \mathcal{I} the \mathcal{M} -interpretation defined by $\mathcal{I}(\mathcal{X}) = \mathbb{T}$ for all $\mathcal{X} \in \mathcal{P}$. Since $x_i \in \mathcal{I}(A_i)$ for all $1 \leq i \leq n$ and $\emptyset \in \mathcal{I}(B_j)^\perp$ for all $1 \leq j \leq m$, then, by the general correctness lemma and since $M \in \mathcal{T}_p$, $M = M[(x_i := x_i)_{1 \leq i \leq n}; (\alpha_j :=_r \emptyset)_{1 \leq j \leq m}] \in \mathcal{I}(A) \subseteq \mathbb{T}$. □

It is important to note that we rely on Theorem 3.15 to derive the preceding result. In other words, arriving at this result directly, without first establishing the interpretations of types and then applying the multiple hypotheses of this theorem, appears to be unfeasible.

This result underscores its significance. Essentially, it implies that the set of typable $\lambda\mu$ -terms is the only set of terms meeting the requirements of a saturated set. In other words, when aiming to demonstrate that typable $\lambda\mu$ -terms possess a particular property, it suffices to consider the set of $\lambda\mu$ -terms with that property and then establish that this set is indeed saturated.

In Sections 4 and 5, we will explore applications of this result.

4 Strong normalization property of $\beta\mu\rho\varepsilon\theta$ -reduction

In this section, we demonstrate that the $\beta\mu$ -reduction enjoys the strong normalization property when augmented with the rules ρ, ε , and θ i.e., all the mentioned rules

except for μ' . This result is not novel; Parigot proved it for his original calculus and de Groote subsequently confirmed it for his version. What distinguishes our approach here is the methodology employed. Instead of verifying strong normalization for a smaller set of rules and subsequently applying commutation rules, we use the result of the previous section. Specifically, we consider all of our reduction rules and choose our base set as the set of typable terms strongly normalizable with respect to all of the considered rules. We then verify that this set possesses all of the properties required for it to be considered saturated.

Hence, we define $\mathcal{R} = \{\beta, \mu, \rho, \theta, \varepsilon\}$ and choose $\mathbb{T} = \mathcal{SN}_{\mathcal{R}} \cap \mathcal{T}_p$. Our aim is to verify that \mathbb{T} is saturated. By Corollary 3.16, this establishes that every typable term is strongly normalizable with respect to $\beta\mu\rho\theta\varepsilon$ -reduction. In the subsequent discussions, we use the abbreviation $M[\alpha :=_r N]$ for $M[\alpha := N]$, as we do not consider μ' -reduction in this case. We then verify the fulfillment of the conditions outlined in Definition 3.2.

First of all, we state a few easy lemmas.

Lemma 4.1 *Let $M, N \in \mathcal{T}$ and $\alpha, \beta, \gamma \in \mathcal{V}_\mu$ such that $\alpha \neq \beta, \gamma$. Then $(M[\beta := \gamma])_\alpha = M_\alpha[\beta := \gamma]$, $(M[x := N])_\alpha = M_\alpha[x := N_\alpha]$, $(M[\beta := N])_\alpha = M_\alpha[\beta := N_\alpha]$, and $(M_\alpha)_\beta = (M_\beta)_\alpha$.*

Proof All the cases follow by a straightforward induction on M . □

Lemma 4.2 *Let $M \in \mathcal{T}$ and $\alpha, \beta \in \mathcal{V}_\mu$. Then $M \rightarrow_{\mathcal{R}} N$ iff $M[\alpha := \beta] \rightarrow_{\mathcal{R}} N[\alpha := \beta]$.*

Proof By induction on M . □

Lemma 4.3 *Let $M \in \mathcal{T}$ and $\alpha, \beta \in \mathcal{V}_\mu$. Then $M \in \mathcal{NF}_{\mathcal{R}}$ iff $M[\alpha := \beta] \in \mathcal{NF}_{\mathcal{R}}$.*

Proof Follows from the previous lemma. □

Now we turn to verifying the properties listed in Definition 3.2.

Lemma 4.4 (Condition (C1)) *Let $M \in \mathbb{T}$ and $x \in \mathcal{V}_\lambda$. Then $\lambda x.M \in \mathbb{T}$.*

Proof Straightforward. □

Before proving the condition (C2), we point out that it is inevitable that we work in the typed framework. Indeed, it is easy to find an example of a $\lambda\mu$ -term $M \in \mathcal{SN}_{\mathcal{R}}$ but $\mu\alpha.M \notin \mathcal{SN}_{\mathcal{R}}$. Namely, let $M = \mu\beta.(\lambda x.(x)x)[\beta]\lambda x.(x)x$, then we have $M \in \mathcal{NF}_{\mathcal{R}}$ but $\mu\alpha.M \rightarrow_\varepsilon \mu\alpha.M_\beta = \mu\alpha.(\lambda x.(x)x)\lambda x.(x)x \notin \mathcal{WN}_{\mathcal{R}}$. More generally, the following remark will be valid throughout the section.

If we choose \mathbb{T} as above, we can observe that none of the conditions of Lemma 3.2 will necessitate a typed framework, except for the condition (C2). However, in adherence to the formulation of Definition 3.2, we include typability assumptions in the respective lemmas. Only the verification of condition (C2) for \mathbb{T} will depend on typability assumptions.

Lemma 4.5 (Condition (C2)) *Let $M \in \mathbb{T}$ and $\alpha \in \mathcal{V}_\mu$ such that $\mu\alpha.M \in \mathcal{T}_p$. Then $\mu\alpha.M \in \mathbb{T}$.*

Proof We prove the lemma with the help of Lemmas 4.6 and 4.8. □

Lemma 4.6 *Let $M \in \mathcal{T}_p$ and assume $M_\alpha \rightarrow_{\mathcal{R}} U$ for some $\alpha \in \mathcal{V}_\mu$. Then $\exists V$ such that $M \rightarrow_{\mathcal{R}} V$ and $V_\alpha = U$.*

Proof Let $M \in \mathcal{T}_p$ and assume $M_\alpha \rightarrow_{\mathcal{R}} U$.

- If M is a variable, there is nothing to prove.
- If $M = \lambda x.N$, we can apply IH.
- Assume $M = [\beta]N$.
 - If $\beta \neq \alpha$, then $M_\alpha = [\beta]N_\alpha$. If $U = [\beta]W$ with $N_\alpha \rightarrow_{\mathcal{R}} W$, then we can apply IH. Otherwise, $N = \mu\gamma.P$, and, applying Lemma 4.1, $[\beta]N_\alpha = [\beta]\mu\gamma.P_\alpha \rightarrow_\rho P_\alpha[\gamma := \beta] = (P[\gamma := \beta])_\alpha$. We let $M \rightarrow_\rho P[\gamma := \beta] = V$.
 - If $\beta = \alpha$, then $M_\alpha = ([\alpha]N)_\alpha = N_\alpha$. In this case we use IH.
- Assume $M = \mu\gamma.N$.
 - If $U = \mu\gamma.W$ such that $N_\alpha \rightarrow_{\mathcal{R}} W$, then we can apply IH.
 - If $N = \mu\delta.P$, then, by Lemma 4.1, $M_\alpha = \mu\gamma\mu\delta.P_\alpha \rightarrow_\varepsilon \mu\gamma.(P_\alpha)_\delta = \mu\gamma.(P_\delta)_\alpha$ and we let $M \rightarrow_\varepsilon \mu\gamma.P_\delta = V$.
 - If $N = [\gamma]P$, $\gamma \notin \text{fv}(P)$, $M_\alpha = \mu\gamma.[\gamma]P_\alpha \rightarrow_\theta P_\alpha$ and we have $M \rightarrow_\theta P = V$.
- If $M = (N)P$, then $M_\alpha = (N_\alpha)P_\alpha$. If $U = (Q)R$ such that $N_\alpha \rightarrow_{\mathcal{R}} Q$ or $P_\alpha \rightarrow_{\mathcal{R}} R$, then we can apply IH. Hence, we may assume that M_α is the redex reduced. We distinguish the various cases according to the structure of N .
 - N cannot be a variable, since then M_α is not a redex.
 - If $N = [\gamma]N'$, then N has type \perp , hence this case is impossible.
 - If $N = \lambda x.N'$, then $M \rightarrow_\beta N'[x := P]$, and, by applying Lemma 4.1, $M_\alpha = (\lambda x.N'_\alpha)P_\alpha \rightarrow_\beta N'_\alpha[x := P_\alpha] = (N'[x := P])_\alpha$, and the result follows.
 - If $N = \mu\gamma.N'$, then $M \rightarrow_\mu \mu\gamma.N'[\gamma := P]$, and, by Lemma 4.1, $M_\alpha = (\mu\gamma.N'_\alpha)P_\alpha \rightarrow_\mu \mu\gamma.N'_\alpha[\gamma := P_\alpha] = (\mu\gamma.N'[\gamma := P])_\alpha$, and the result follows.
 - The case $N = (N_1)N_2$ is impossible, since then M_α is not a redex.

□

Definition 4.7 (Strong normalization) *We say that a term M is strongly normalizable, in notation $M \in \mathcal{SN}_{\mathcal{R}}$, if every reduction sequence starting from M terminates. Since the reduction tree is locally finite, then, by König's lemma, this is equivalent to asserting that the lengths of the reduction sequences starting from M is bounded from above. Let us denote the length of the longest reduction sequence of M by $\eta(M)$. Observe that, if $M \rightarrow_{\mathcal{R}} M'$, then $\eta(M) > \eta(M')$. Then $M \in \mathcal{SN}_{\mathcal{R}}$ iff, for every $M \rightarrow_{\mathcal{R}} N$, we have $N \in \mathcal{SN}_{\mathcal{R}}$. Moreover, $\eta(M) = \max\{\eta(N) \mid M \rightarrow_{\mathcal{R}} N\} + 1$.*

Lemma 4.8 *Let $M \in \mathcal{T}_p$ and $\alpha \in \mathcal{V}_\mu$. If $M \in \mathcal{SN}_{\mathcal{R}}$, then $M_\alpha \in \mathcal{SN}_{\mathcal{R}}$ and $\eta(M_\alpha) \leq \eta(M)$.*

Proof Let $M \in \mathcal{SN}_{\mathcal{R}}$ and assume $M_\alpha \rightarrow_{\mathcal{R}} U$. We reason by induction on $\eta(M)$. By the previous lemma, $\exists V$ such that $M \rightarrow_{\mathcal{R}} V$ and $V_\alpha = U$. Then $\eta(V) < \eta(M)$, hence, we can apply IH to V . We obtain the bound for $\eta(M_\alpha)$ if we take into account the inequality $\eta(U) + 1 \leq \eta(V) + 1 \leq \eta(M)$, which follows from the previous fact and IH. □

We can now turn to the proof of Lemma 4.5.

Proof of Lemma 4.2.

Let $M \in \mathbb{T}$ and $\alpha \in \mathcal{V}_\mu$ such that $\mu\alpha.M \in \mathcal{T}_p$. We assume by lexicographic induction with respect to $(\eta(M), \text{cxt}_y(M))$ that $M \in \mathcal{SN}_{\mathcal{R}}$ implies $\mu\alpha.M \in \mathcal{SN}_{\mathcal{R}}$. Let $\mu\alpha.M \rightarrow_{\mathcal{R}} N$. We claim that $N \in \mathcal{SN}_{\mathcal{R}}$. We examine the three cases possible.

- If $N = \mu\alpha.M'$ with $M \rightarrow_{\mathcal{R}} M'$, then $\eta(M') < \eta(M)$ and, by IH, $N \in \mathcal{SN}_{\mathcal{R}}$.
- If $M = \mu\beta.P$ and $\mu\alpha.M \rightarrow_{\varepsilon} \mu\alpha.P_\beta = N$, then $P \in \mathcal{SN}_{\mathcal{R}}$, $\eta(P_\beta) \leq \eta(P) \leq \eta(M)$ and $\text{cxt}_y(P) < \text{cxt}_y(M)$, thus, by IH, $N \in \mathcal{SN}_{\mathcal{R}}$.
- If $M = [\alpha]N$, $\alpha \notin \text{fv}(N)$ and $\mu\alpha.M \rightarrow_{\theta} N$, then $N \in \mathcal{SN}_{\mathcal{R}}$.

□

Lemma 4.9 (Condition (C3)) *Let $M \in \mathbb{T}$ and $\alpha \in \mathcal{V}_\mu$ such that $[\alpha]M \in \mathcal{T}_p$. Then $[\alpha]M \in \mathbb{T}$.*

Proof Let $M \in \mathbb{T}$ and $\alpha \in \mathcal{V}_\mu$ such that $[\alpha]M \in \mathcal{T}_p$. We will prove, by induction on $\eta(M)$, that $[\alpha]M \in \mathcal{SN}_{\mathcal{R}}$. Let $[\alpha]M \rightarrow_{\mathcal{R}} N$. It suffices to check that $N \in \mathcal{SN}_{\mathcal{R}}$. We examine two cases.

- If $N = [\alpha]M'$ with $M \rightarrow_{\mathcal{R}} M'$, then $\eta(M') < \eta(M)$ and, by IH, $N \in \mathcal{SN}_{\mathcal{R}}$.
- If $M = \mu\beta.P$ and $\mu\alpha.M \rightarrow_{\rho} P[\beta := \alpha] = N$, then $P \in \mathcal{SN}_{\mathcal{R}}$, thus, by Lemma 4.3, $N \in \mathcal{SN}_{\mathcal{R}}$.

□

Condition (C4) proves to be trivial due to the fact that we have omitted the μ' -reduction, that is, the symmetric counterpart of μ .

Lemma 4.10 (Condition (C4)) *Let $n \geq 0$, $N_1, \dots, N_n \in \mathbb{T}$ and $x \in \mathcal{V}_\lambda$. Then $(x)N_1 \dots N_n \in \mathcal{T}_p$ involves $(x)N_1 \dots N_n \in \mathbb{T}$.*

Proof We argue by induction on $\sum_{k=1}^n \eta(N_k)$. Indeed, $(x)N_1 \dots N_n \rightarrow_{\mathcal{R}} U$ implies $U = (x)U_1 \dots U_n$, where $N_i \rightarrow_{\mathcal{R}} U_i$ for one index $1 \leq i \leq n$ and $N_j = U_j$ if $1 \leq j \leq n$ and $i \neq j$. By this, the result follows. □

The lemma below contains an easy property of the reduction.

Lemma 4.11 *Let M, U, N be given such that $M \rightarrow_{\mathcal{R}} U$. Then $M[x := N] \rightarrow_{\mathcal{R}} U[x := N]$.*

Proof By induction on M .

□

We reveal two observations in the next lemma before verifying the condition (C5).

Lemma 4.12

1. *Let $M[x := N] \in \mathcal{SN}_{\mathcal{R}}$ for some terms M, N and variable $x \in \mathcal{V}_\lambda$. Then $M \in \mathcal{SN}_{\mathcal{R}}$.*
2. *Let $M[\alpha := N] \in \mathcal{SN}_{\mathcal{R}}$ for some terms M, N and variable $\alpha \in \mathcal{V}_\mu$. Then $M \in \mathcal{SN}_{\mathcal{R}}$.*

Proof We deal only with Point 1, the other case being similar. Let $M \rightarrow_{\mathcal{R}} U$. We prove that $U \in \mathcal{SN}_{\mathcal{R}}$. We argue by induction on $\eta(M[x := N])$. From the assumption $M \rightarrow_{\mathcal{R}} U$, we deduce $M[x := N] \rightarrow_{\mathcal{R}} U[x := N]$ by the previous lemma. Since $M[x := N] \in \mathcal{SN}_{\mathcal{R}}$, we have $U[x := N] \in \mathcal{SN}_{\mathcal{R}}$ and $\eta(U[x := N]) < \eta(M[x := N])$, which, by IH, gives the result. □

Lemma 4.13 (Condition (C5)) *Let $(M[x := N])\bar{P} \in \mathbb{T}$, $N \in \mathbb{T}$ and $(\lambda x.M)N\bar{P} \in \mathcal{T}_p$. Then $(\lambda x.M)N\bar{P} \in \mathbb{T}$.*

Proof Firstly, we observe that $(M[x := N])\bar{P} \in \mathcal{SN}_{\mathcal{R}}$ implies $M, \bar{P} \in \mathcal{SN}_{\mathcal{R}}$. We proceed by induction on $\eta(M) + \eta(N) + \eta(\bar{P})$ where $\eta(\bar{P})$ is the sum of the η s of the terms of the sequence \bar{P} . Let $(\lambda x.M)N\bar{P} \rightarrow_{\mathcal{R}} Q$. We examine two cases.

- If $Q = (\lambda x.M')N'\bar{P}'$ with $M \rightarrow_{\mathcal{R}} M'$ or $N \rightarrow_{\mathcal{R}} N'$ or $\bar{P} \rightarrow_{\mathcal{R}} \bar{P}'$, then we can apply IH.
- If $Q = (M[x := N])\bar{P}$, we have the result by assumption. □

Lemma 4.14 (Condition (C6)) *Let $\mu\alpha.M[\alpha := \bar{N}] \in \mathbb{T}$, $\bar{N} \in \mathbb{T}^{<\omega}$ and $(\mu\alpha.M)\bar{N} \in \mathcal{T}_p$. Then $(\mu\alpha.M)\bar{N} \in \mathbb{T}$.*

Proof We observe that, by Lemma 4.12, $\mu\alpha.M[\alpha := \bar{N}] \in \mathcal{SN}_{\mathcal{R}}$ implies $M \in \mathcal{SN}_{\mathcal{R}}$. We note that, if $\bar{N} = N_1N_2 \dots N_n$, then $M[\alpha := \bar{N}] = M[\alpha := N_1][\alpha := N_2] \dots [\alpha := N_n]$. Therefore, we need to prove that, if $(\mu\alpha.M[\alpha := N])\bar{P} \in \mathcal{SN}_{\mathcal{R}}$ with $N \in \mathcal{SN}_{\mathcal{R}}$ and $\bar{P} \in \mathcal{SN}_{\mathcal{R}}^{<\omega}$, then $(\mu\alpha.M)N\bar{P} \in \mathcal{SN}_{\mathcal{R}}$. We proceed by induction on $\eta(M) + \eta(N) + \eta(\bar{P})$ where $\eta(\bar{P})$ is the sum of the η s of the terms of the sequence \bar{P} . Let $(\mu\alpha.M)N\bar{P} \rightarrow_{\mathcal{R}} Q$. We examine two cases.

- If $Q = (\mu\alpha.M')N'\bar{P}'$ with $M \rightarrow_{\mathcal{R}} M'$ or $N \rightarrow_{\mathcal{R}} N'$ or $\bar{P} \rightarrow_{\mathcal{R}} \bar{P}'$, then we can apply IH.
- If $Q = (\mu\alpha.M[\alpha := N])\bar{P}$, we have the result. □

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

Theorem 4.15 *\mathbb{T} is a saturated set.*

Proof We put together Lemmas 4.4, 4.5, 4.9, 4.10, 4.13 and 4.14. □

Corollary 4.16 *If $M \in \mathcal{T}_p$, then $M \in \mathcal{SN}_{\beta\mu\rho\theta\varepsilon}$.*

Proof We apply Corollary 3.16 and Theorem 4.15. □

5 Weak normalization property of $\beta\mu\mu'\rho\varepsilon\theta$ -reduction

In this section, we prove that $\beta\mu\rho\varepsilon\theta$ -reduction augmented with the μ' -rule has the weak normalization property. In [4], we provided a syntactic proof of this result. However, that proof was notably complex, involving the strong normalization of β -reduction within a typed framework and the weak normalization of the other reductions within an untyped framework. The most challenging part was devising an algorithm to combine the two normalization results to establish the weak normalization of the set of all reductions. The proof presented here takes a different approach. It relies on the results of Section 3 by choosing a suitable set that satisfies all the properties required for the saturation. We observe that condition (C4) is the most difficult one to establish, and, interestingly, it will be formulated and verified within the untyped context. While this proof for the weak normalization property is elegant and concise, it lacks the constructive aspect found in the previous proof [4], and as a result, we are unable to extract a concrete normalization algorithm.

Additionally, it is worth noting that, except for verifying condition (C2) for \mathbb{T}' , none of the other properties require typability assumptions. While the formulations

of the lemmas mention typability assumptions to align with Definition 3.2, we will present proofs that do not rely on types for conditions with the exception of (C2).

We first show in the next two lemmas that, for the proof of the weak normalization property of $\beta\mu\mu'\rho\varepsilon\theta$ -reduction, it is sufficient to verify that $\beta\mu\mu'\rho\varepsilon$ -reduction enjoys weak normalization. This is because we can add θ -reduction to the set of reductions without compromising the weak normalization property.

Lemma 5.1 *Let us suppose $M \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$ and $M \rightarrow_\theta N$ for some $M, N \in \mathcal{T}$. Then $N \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$ and, if N starts with λ (with μ , resp.), then M also starts with λ (with μ , resp.).*

Proof By induction on M . It suffices to check the property for one-step of reduction.

- If $M = \lambda x.M'$, the result is trivial.
- If $M = \mu\alpha.M'$ and $N = \mu\alpha.N'$ where $M' \rightarrow_\theta N'$, then, by IH, $N' \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$ and N' does not start with μ . Thus $N \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$ and the second property is obviously verified.
- If $M = \mu\alpha.[\alpha]M'$, $\alpha \notin \text{fv}(M')$ and $N = M'$, then $M' \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$ and the second property is obviously verified.
- If $M = [\alpha]M'$, then $N = [\alpha]N'$ where $M' \rightarrow_\theta N'$. Then, by IH, $N' \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$ and N' does not start with μ . Hence, $N \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$.
- If $M = (M_1)M_2$, $N = (M'_1)M_2$ and $M_1 \rightarrow_\theta M'_1$, then $M'_1, M_2 \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$ and M'_1 does not start with μ or λ . Hence, $N \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$.
- If $M = (M_1)M_2$, $N = (M_1)M'_2$ and $M_2 \rightarrow_\theta M'_2$, then $M_1, M'_2 \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$ and M'_2 does not start with μ . Hence, $N \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$. □

Lemma 5.2 *The θ -reduction strongly normalizes.*

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

Theorem 5.3 $\mathcal{WN}_{\beta\mu\mu'\rho\varepsilon} \subseteq \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon\theta}$ holds true.

Proof If $M \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$, then $\exists M'$ such that $M \rightarrow_{\beta\mu\mu'\rho\varepsilon} M'$ and $M' \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon}$. Thus, by Lemmas 5.1 and 5.2, $\exists N \in \mathcal{NF}_{\beta\mu\mu'\rho\varepsilon\theta}$ such that $M' \rightarrow_\theta N$. □

Let us now recall the example of our paper [4] which shows that with the reduction μ' we lose the strong normalization property. Let $M = (\mu\beta.U)U$ where $U = \mu\alpha.[\alpha][\alpha]x$, then $x : \perp \vdash M : \perp$ and there are M_1, M_2, M_3 such that $M \rightarrow_{\mu'} M_1 \rightarrow_{\mu} M_2 \rightarrow_{\rho} M_3 \rightarrow_\theta M$, which means $M \notin \mathcal{SN}_{\beta\mu\mu'\rho\varepsilon\theta}$. Hence, we cannot hope to prove the strong normalizability of $\{\beta, \mu, \mu', \rho, \varepsilon\}$ -reduction. Instead, we intend to show that $\{\beta, \mu, \mu', \rho, \varepsilon\}$ -reduction enjoys the weak normalization property.

In what follows, we let $\mathcal{R}' = \{\beta, \mu, \mu', \rho, \varepsilon\}$ and $\mathbb{T}' = \mathcal{WN}_{\mathcal{R}'} \cap \mathcal{T}_p$. Our aim is to verify that \mathbb{T}' is saturated, from which, by Corollary 3.16, it follows that every typable term is weakly normalizable. We verify the conditions of Definition 3.2 one by one below. Beforehand, we deal with some lemmas that will be needed in the proofs. Lemma 5.5 is similar to Lemma 4.3.

Lemma 5.4 *Let $M \in \mathcal{T}$ and $\alpha, \beta \in \mathcal{V}_\mu$. Then $M \rightarrow_{\mathcal{R}'} N$ iff $M[\alpha := \beta] \rightarrow_{\mathcal{R}'} N[\alpha := \beta]$.*

Proof By induction on M . □

Lemma 5.5 *Let $M \in \mathcal{T}$ and $\alpha, \beta \in \mathcal{V}_\mu$. Then $M \in \mathcal{NF}_{\mathcal{R}'}$ iff $M[\alpha := \beta] \in \mathcal{NF}_{\mathcal{R}'}$.*

Proof Follows from the previous lemma. \square

Lemma 5.6 *Let $M \in \mathcal{T}_p$ and $\alpha \in \mathcal{V}_\mu$. If $M \in \mathcal{NF}_{\mathcal{R}'}$, then $M_\alpha \in \mathcal{NF}_{\mathcal{R}'}$. Moreover, the following statements hold.*

1. *If M_α starts with μ , then M starts with μ .*
2. *If M_α starts with λ , then M starts with λ or $[\cdot]$.*

Proof By induction on M . We detail only the more interesting cases.

- If $M = [\beta]M'$ and $\beta \neq \alpha$, then M' does not start with μ , $M_\alpha = [\beta]M'_\alpha$ and, by IH, $M'_\alpha \in \mathcal{NF}_{\mathcal{R}'}$ and M'_α does not start with μ . Hence, we have the result.
- If $M = [\alpha]M'$, then M' does not start with μ , and $M_\alpha = M'_\alpha$. By IH, $M'_\alpha \in \mathcal{NF}_{\mathcal{R}'}$ and M'_α does not start with μ . Obviously, if M_α starts with λ , then M starts with $[\cdot]$.
- If $M = (P)Q$, then P, Q do not start with μ , P does not start with λ and $M_\alpha = (P_\alpha)Q_\alpha$. By IH, $P_\alpha, Q_\alpha \in \mathcal{NF}_{\mathcal{R}'}$ and P_α, Q_α do not start with μ . If P_α starts with λ , then P starts either with λ or with $[\cdot]$. Since $M \in \mathcal{NF}_{\mathcal{R}'}$, P cannot start with λ and, since $M \in \mathcal{T}_p$, P cannot start with $[\cdot]$, either. Hence, $M_\alpha \in \mathcal{NF}_{\mathcal{R}'}$ \square

Remark 5.7 *In the lemma above we obviously need the assumption that M be typable. Otherwise, let $M = ([\alpha]\lambda x.P)Q$. Then $M \in \mathcal{NF}_{\mathcal{R}'}$, but $M_\alpha = (\lambda x.P)Q \notin \mathcal{NF}_{\mathcal{R}'}$.*

The next lemma is intuitive and has the consequence that we are not able to create a μ , λ or $[\cdot]$ by a μ - or μ' -substitution or by replacing a μ -variable with another one.

Lemma 5.8 *Let $M, N \in \mathcal{T}$, $\alpha \in \mathcal{V}_\mu$ and $s \in \{r, l\}$.*

1. *If $M[\alpha :=_s N]$ starts with λ (resp. μ , $[\cdot]$), then M also starts with λ (resp. μ , $[\cdot]$).*
2. *If $M[\alpha := \beta]$ starts with λ (resp. μ , $[\cdot]$), then M also starts with λ (resp. μ , $[\cdot]$).*

Proof By induction on M . \square

Lemma 5.9 (Condition (C1)) *Let us suppose $M \in \mathbb{T}'$. Then $\lambda x.M \in \mathbb{T}'$.*

Proof Obvious. \square

Lemma 5.10 (Condition (C2)) *Let $M \in \mathbb{T}'$ and $\alpha \in \mathcal{V}_\mu$. If $\mu\alpha.M \in \mathcal{T}_p$, then $\mu\alpha.M \in \mathbb{T}'$.*

Proof We consider a normalization of M i.e. $M \rightarrow N \in \mathcal{NF}_{\mathcal{R}'}$. We take the reduction sequence $\mu\alpha.M \rightarrow \mu\alpha.N$ obtained thereof. We distinguish the different cases.

- If N does not start with μ , then $\mu\alpha.N \in \mathcal{NF}_{\mathcal{R}'}$ (note that $\theta \notin \mathcal{R}'$), thus $\mu\alpha.M \in \mathcal{WN}_{\mathcal{R}'}$.
- If $N = \mu\beta.P$, then P does not start with μ . By Lemma 5.6, we have $\mu\alpha.M \rightarrow \mu\alpha.\mu\beta.P \rightarrow_\varepsilon \mu\alpha.P_\beta \in \mathcal{NF}_{\mathcal{R}'}$. Hence, $\mu\alpha.M \in \mathcal{WN}_{\mathcal{R}'}$. \square

We demonstrate the fact that the assumption $\mu\alpha.M \in \mathcal{T}_p$ is needed in the lemma above by giving a simple example. Let $M = \mu\beta.([\beta]\lambda y.(y)\delta)\delta$, where $\delta = \lambda x.(x)x$. Then $M \in \mathcal{NF}_{\mathcal{R}'}$ and $\mu\alpha.M \rightarrow_\varepsilon \mu\alpha.M_\beta = \mu\alpha.(\lambda y.(y)\delta)\delta \rightarrow_\beta \mu\alpha.(\delta)\delta$. Hence, $\mu\alpha.M \notin \mathcal{WN}_{\mathcal{R}'}$.

Lemma 5.11 (Condition (C3)) *Let $M \in \mathbb{T}'$ and $\alpha \in \mathcal{V}_\mu$. If $[\alpha]M \in \mathcal{T}_p$, then $[\alpha]M \in \mathbb{T}'$.*

Proof We consider a normalization of M i.e. $M \rightarrow N \in \mathcal{NF}_{\mathcal{R}'}$. Then we take the reduction sequence $[\alpha]M \rightarrow [\alpha]N$ obtained thereof. We distinguish the different cases.

- If N does not start with μ , then $[\alpha]N \in \mathcal{NF}_{\mathcal{R}'}$. Thus $[\alpha]M \in \mathcal{WN}_{\mathcal{R}'}$.
- If $N = \mu\beta.P$, then, by Lemma 5.5, we have $[\alpha]M \rightarrow [\alpha]\mu\beta.P \rightarrow_\rho P[\beta := \alpha] \in \mathcal{NF}_{\mathcal{R}'}$, then $[\alpha]M \in \mathcal{WN}_{\mathcal{R}'}$. □

Next, we are going to prove the condition (C4) for \mathbb{T}' .

Lemma 5.12 (Condition (C4)) *Let $n \geq 0$, $N_1, \dots, N_n \in \mathbb{T}'$ and $x \in \mathcal{V}_\lambda$, then $(x)N_1 \dots N_n \in \mathbb{T}'$.*

We prove the lemma with the help of several auxiliary lemmas. Our intuition is that, when we are given a term $(x)N_1 \dots N_n$ with $N_1, \dots, N_n \in \mathcal{WN}$, then we move from left to right until we find the first $1 \leq i \leq n$ such that $N_i = \mu\gamma.N'_i$ for some N'_i . We normalize the term $(x)N_1 \dots N_i$, and, after obtaining the normal form $\mu\gamma.N''_i$, we proceed with normalizing $(\mu\gamma.N''_i)N_{i+1} \dots N_n$. The normalization strategy is a little tricky at that point, however. When the next term N_{i+1} does not start with a μ , then we perform a μ -reduction for the redex $(\mu\gamma.N''_i)N_{i+1}$. On the other hand, if $N_{i+1} = \mu\gamma_{i+1}.N'_{i+1}$, then we continue with a μ' -reduction concerning the μ' -redex $(\mu\gamma.N''_i)\mu\gamma_{i+1}.N'_{i+1}$. We continue in this way for the remaining components of the application. First of all, we introduce some necessary notions.

Definition 5.13 (α -clean property) *Let $M \in \mathcal{T}_p$ and $\alpha \in \mathcal{V}_\mu$. We say that M is α -clean if, for every subterm $[\alpha]U$ of M , U does not start with λ .*

Intuitively, α -clean terms do not create new β -redexes when a μ -substitution is considered with respect to α .

To establish condition (C4), we rely on some technical lemmas (Lemmas 5.14, 5.15, and 5.16) to attain a weak normalization result. As we have seen in the above explanation, we need to normalize the result of a μ -substitution of a normal $\lambda\mu$ -term into another normal $\lambda\mu$ -term. The difficulty in such normalization arises from the possibility of encountering β -redexes. The α -clean condition aims to prevent these occurrences. We achieve this normalization result in two steps. First, we normalize without the μ' -reduction (without the μ -reduction, resp.) and we describe precisely the remaining μ' -redexes (μ -redexes, resp.). Then, we handle the final normalization by eliminating these redexes.

Lemma 5.14

1. *Let $M, N \in \mathcal{NF}_{\mathcal{R}'}$ such that M is α -clean. Then $M[\alpha :=_r N] \in \mathcal{NF}_{\beta\mu\rho\varepsilon}$ is α -clean and the μ' -redexes of $M[\alpha :=_r N]$ are of the form $[\alpha](U)N$ if $N = \mu\beta.N'$. In particular, if $N \neq \mu\beta.N'$, then $M[\alpha :=_r N] \in \mathcal{NF}_{\mathcal{R}'}$ as well.*
2. *Let $M, N \in \mathcal{NF}_{\mathcal{R}'}$ such that $N \neq \lambda x.N'$ for some N' . Then $M[\alpha :=_l N] \in \mathcal{NF}_{\beta\mu'\rho\varepsilon}$ and the μ -redexes of $M[\alpha :=_l N]$ are of the form $[\alpha](N)U$ if $N = \mu\beta.N'$. In particular, if $N \neq \mu\beta.N'$, then $M[\alpha :=_l N] \in \mathcal{NF}_{\mathcal{R}'}$ as well.*

Proof Both points can be proved by induction on M .

1.
 - In case of $M = \lambda x.M'$, we apply the induction hypothesis.
 - If $M = \mu\beta.M'$, then M' does not start with μ and $M[\alpha :=_r N] = \mu\beta.M'[\alpha :=_r N]$. By Lemma 5.8, $M'[\alpha :=_r N]$ does not start with μ and we apply IH on M' to obtain the result.
 - If $M = (M_1)M_2$, then M_1 does not start with μ or λ , M_2 does not start with μ and $M[\alpha :=_r N] = (M_1[\alpha :=_r N])M_2[\alpha :=_r N]$. By Lemma 5.8, $M_1[\alpha :=_r N]$ does not start with μ or λ , $M_2[\alpha :=_r N]$ does not start with μ and, applying IH on M_1, M_2 , we obtain the result.
 - If $M = [\beta]M'$ and $\beta \neq \alpha$, then M' does not start with μ and $M[\alpha :=_r N] = [\beta]M'[\alpha :=_r N]$. By Lemma 5.8, $M'[\alpha :=_r N]$ does not start with μ and the result follows from IH.
 - If $M = [\alpha]M'$, then M' does not start with μ and $M[\alpha :=_r N] = [\alpha](M'[\alpha :=_r N])N$. Since M is α -clean, M' does not start with λ and, by Lemma 5.8, $M'[\alpha :=_r N]$ does not start with μ or λ . By applying IH, we see that $[\alpha](M_1[\alpha :=_r N])N \in \mathcal{NF}_{\beta\mu\rho\varepsilon}$ is α -clean and the μ' -redexes are of the desired form.
2.
 - In case of $M = \lambda x.M', M = \mu\beta.M', M = (M_1)M_2$ and $M = [\beta]M'$ ($\alpha \neq \beta$), we apply, as in Point 1, IH.
 - If $M = [\alpha]M'$, then M' does not start with μ and $M[\alpha :=_l N] = [\alpha](N)M'[\alpha :=_l N]$. By Lemma 5.8, $M'[\alpha :=_r N]$ does not start with μ . By virtue of IH and since N does not start with λ , we see that $[\alpha](N)M_1[\alpha :=_l N] \in \mathcal{NF}_{\beta\mu'\rho\varepsilon}$ and the μ -redexes are of the desired form.

□

Lemma 5.15 *Let $M, N \in \mathcal{NF}_{\mathcal{R}'}$ such that $M \neq \lambda x.M'$ and, if $M = \mu\alpha.M'$, then M is α -clean. Then $N[\gamma :=_l M] \twoheadrightarrow_{\mu\rho} P \in \mathcal{NF}_{\mathcal{R}'}$ and P is γ -clean. Moreover, if N does not start with μ , then the same is true for P .*

Proof The proof proceeds by induction on N .

- If $N = \lambda x.N'$, then the assertion is straightforward.
- If $N = [\beta]N'$ ($\beta \neq \gamma$). Then N' does not start with μ . By IH, $N'[\gamma :=_l M] \twoheadrightarrow_{\mu\rho} P' \in \mathcal{NF}_{\mathcal{R}'}$ and P' does not start with μ . Hence, $[\beta]P' \in \mathcal{NF}_{\mathcal{R}'}$.
- If $N = [\gamma]N'$, then $N[\gamma :=_l M] = [\gamma](M)N'[\gamma :=_l M]$ and N' does not start with μ . Hence by Lemma 5.8, $N'[\gamma :=_l M]$ does not start with μ either. By IH, $N'[\gamma :=_l M] \twoheadrightarrow_{\mu\rho} P' \in \mathcal{NF}_{\mathcal{R}'}$ and P' does not start with μ . If M does not start with μ , then we are ready. Otherwise, assume $M = \mu\alpha.M'$. Then M' does not start with μ and M is α -clean. In this case, $N[\gamma :=_l M] = [\gamma](\mu\alpha.M')N'[\gamma :=_l M] \rightarrow_{\mu} [\gamma]\mu\alpha.M'[\alpha :=_r N'[\gamma :=_l M]] \rightarrow_{\rho} M'[\alpha :=_r N'[\gamma :=_l M]][\alpha := \gamma] \twoheadrightarrow_{\mu\rho} P \in \mathcal{NF}_{\mathcal{R}'}$ by applying Lemmas 5.14 and 5.5 and IH. From Lemma 5.8, it follows that P does not start with μ and, by IH, P is γ -clean.
- $N = \mu\beta.N'$. Then we can apply IH.
- $N = (N_1)N_2$. Then N_1 does not start with λ or μ and N_2 does not start with μ . By using IH, we obtain the result.

□

Lemma 5.16 *Let $P, Q \in \mathcal{NF}_{\mathcal{R}'}$ and $\alpha \in \mathcal{V}_\mu$. Then the following statements hold.*

1. *Let $P = \mu\alpha.P'$ such that P is α -clean. Assume Q does not start with μ . Then $\exists R$ for which $(P)Q \rightarrow_\mu \mu\alpha.R \in \mathcal{NF}_{\mathcal{R}'}$ and R is α -clean.*
2. *Let $Q = \mu\gamma.Q'$. Assume $P \neq \lambda x.P'$ and, if $P = \mu\alpha.P'$, then P is α -clean. Then $\exists R$ for which $(P)Q \rightarrow_{\mu\mu'\rho} \mu\gamma.R \in \mathcal{NF}_{\mathcal{R}'}$ and R is γ -clean.*

Proof Let P, Q be as in the lemma. We verify the two statements of the lemma.

1. Assume $P = \mu\alpha.P'$ such that P is α -clean and Q does not start with μ . Then we apply Lemma 5.14 to $(\mu\alpha.P')Q \rightarrow_\mu \mu\alpha.P'[\alpha :=_r Q]$ to obtain the result.
2. We apply the previous lemma to $(P)Q \rightarrow_{\mu'} \mu\gamma.Q'[\gamma :=_l P]$. □

Proof of Lemma 5.12 Let us assume we are given terms $N_1, \dots, N_n \in \mathcal{WN}_{\mathcal{R}'}$, where $x \in \mathcal{V}_\lambda$. We shall assume $n > 0$, as, otherwise, the statement is trivial. Let $1 \leq i \leq n$ be the first index such that $N_i = \mu\gamma.N'_i$. We apply μ' -reductions to $(x)N_1 \dots N_{i-1} \mu\gamma_i.N'_i$ to obtain $\mu\gamma_i.N'_i[\gamma_i :=_l (x)N_1 \dots N_{i-1}] = \mu\gamma_i.N'$. Then, according to the lemma above, $\mu\gamma_i.N'$ is γ_i -clean and, by Lemma 5.14, $\mu\gamma_i.N' \in \mathcal{NF}_{\mathcal{R}'}$. The remaining term to normalize is $(\mu\gamma_i.N')N_{i+1} \dots N_n$. We prove by induction on i that $(\mu\gamma_i.N')N_{i+1} \dots N_n \in \mathcal{WN}_{\mathcal{R}'}$. Assume N_{i+1} does not start with μ . Then Point 1 of Lemma 5.16 applies, and we obtain $(\mu\gamma_i.N')N_{i+1} \rightarrow_\mu \mu\gamma_i.N''$ such that $\mu\gamma_i.N''$ is γ_i -clean and $\mu\gamma_i.N'' \in \mathcal{NF}_{\mathcal{R}'}$. By IH, $(\mu\gamma_i.N'')N_{i+2} \dots N_n \in \mathcal{WN}_{\mathcal{R}'}$. On the other hand, suppose $N_{i+1} = \mu\gamma_{i+1}.N'_{i+1}$. Applying Point 2 of Lemma 5.16 to $(\mu\gamma_i.N')\mu\gamma_{i+1}.N'_{i+1}$, we obtain $\mu\gamma_{i+1}.R \in \mathcal{NF}_{\mathcal{R}'}$ such that R is γ_{i+1} -clean and we conclude that $(\mu\gamma_{i+1}.R)N_{i+2} \dots N_n \in \mathcal{WN}_{\mathcal{R}'}$. □

We can turn to the conditions (C5) and (C6).

Lemma 5.17 (Condition (C5)) *Let $M, N \in \mathcal{T}_p$, $N \in \mathbb{T}'$ and $\bar{P} \in \mathcal{T}_p^{<\omega}$. If $(M[x := N])\bar{P} \in \mathbb{T}'$ and $(\lambda x.M)N\bar{P} \in \mathcal{T}_p$, then $(\lambda x.M)N\bar{P} \in \mathbb{T}'$.*

Proof Indeed, we have $(\lambda x.M)N\bar{P} \rightarrow_\beta (M[x := N])\bar{P} \in \mathcal{WN}_{\mathcal{R}'}$. □

Lemma 5.18 (Condition (C6)) *Let $M, N \in \mathcal{T}_p$ and $\bar{N} \in (\mathbb{T}')^{<\omega}$. If $\mu\alpha.M[\alpha :=_r \bar{N}] \in \mathbb{T}'$ and $(\mu\alpha.M)\bar{N} \in \mathcal{T}_p$, then $(\mu\alpha.M)\bar{N} \in \mathbb{T}'$.*

Proof Indeed, we have $(\mu\alpha.M)\bar{N} \rightarrow_\mu \mu\alpha.M[\alpha :=_r \bar{N}] \in \mathcal{WN}_{\mathcal{R}'}$. □

We are now in a position to state and prove the main theorem of this section.

Theorem 5.19 *\mathbb{T}' is a saturated set.*

Proof We use Lemmas 5.9, 5.10, 5.11, 5.12, 5.17 and 5.18. □

Theorem 5.20 *If $M \in \mathcal{T}_p$, then $M \in \mathcal{WN}_{\beta\mu\rho\epsilon\theta}$.*

Proof We apply Corollary 3.16 and Theorems 5.3 and 5.19. □

Remark 5.21 As a final remark, we would emphasize that the weak normalization property follows for the typable terms without any additional effort. Notably, verifying that the set $\mathcal{WN}_{\beta\mu\rho\epsilon\theta} \cap \mathcal{T}_p$ is saturated is easy, as condition (C3) becomes trivial with the absence of the μ' -rule. Thus, if M is a typable $\lambda\mu$ -term, then $M \in \mathcal{WN}_{\beta\mu\rho\epsilon\theta}$.

6 Future work

- We have proven normalization results (both strong and weak) for sets of reduction rules in the simply typed $\lambda\mu$ -calculus. We believe that this method is sufficiently general to allow us to establish analogous results when we introduce additional reductions. For instance, we can consider the following new typing rule:

$$[\alpha][\beta]M \rightarrow_{\delta} [\beta]M_{\alpha} \quad (\beta \neq \alpha)$$

and

$$[\alpha][\alpha]M \rightarrow_{\delta} M_{\alpha}$$

We can investigate the normalization properties (both weak and strong) of a set of rules that includes this new reduction rule δ . We are confident that our method is well-suited for such investigations.

- We have already mentioned that the reduction rule μ' does not preserve types during reduction in a typed system based on second-order logic, and, that Raffalli provided an example of a term of type A that can be reduced (using the rule μ') to a term of type B , where B can be any type. This result appeared in Py's thesis [22]. To address this problem, the main idea is to give algorithmic content to the rules associated with the quantifier \forall . Py has already accomplished this in the previously cited work. In light of this, our aim is to expand the method of reducibility candidates in a way that allows us to establish normalization results within a typed framework for second-order logic.

References

- [1] F. Barbanera and S. Berardi. *A symmetric lambda calculus for classical program extraction*, In: M. Hagiya and J.C. Mitchell (editors), Proceedings of theoretical aspects of computer software, TACS '94., Lecture Notes in Computer Science (789), pp. 495-515, Springer Verlag, 1994.
- [2] Battyányi, P., and Nour, K. *Strong normalization of λ_{Prop}^{Sym} - and $\bar{\lambda}\mu\tilde{\mu}^*$ -calculi*, Logical Methods in Computer Science, vol. 13 (3:34), pp. 1-22, 2017, [https://doi.org/10.23638/LMCS-13\(3:34\)2017](https://doi.org/10.23638/LMCS-13(3:34)2017).
- [3] Battyányi, P., and Nour, K. *Normalization proofs for the un-typed $\mu\mu'$ -calculus*, Special Issue: LICMA'19 Lebanese International Conference on Mathematics and Applications., AIMS Mathematics, 5(4), pp. 3702-3713, 2020, <https://doi.org/10.3934/math.2020239>.
- [4] Battyányi, P., and Nour, K. *Normalization in the simply typed $\lambda\mu\mu'\rho\theta\varepsilon$ -calculus*. Mathematical Structures in Computer Science, 32(8), pp. 1066-1098, 2022, <https://doi.org/10.1017/S096012952200041X>.
- [5] Curien, P.-L., and Herbelin, H. *The duality of computation*, In: M. Odersky, P. Wadler (editors), Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming, ICFP '00, pp. 233-243, ACM Press, 2000.
- [6] David, R., and Nour, K. *Arithmetical proofs of strong normalization results for the symmetric $\lambda\mu\mu'$ -calculus*, In: P. Urzyczyn (editor), Typed Lambda Calculi and Applications, TLCA '05, Lecture Notes in Computer Science (3461), pp. 162-178, Springer Verlag, 2005.

- [7] David, R., and Nour, K. *Arithmetical proofs of strong normalization results for symmetric lambda calculi*, *Fundamenta Informaticae*, 77(4), pp. 489-510, 2007.
- [8] Girard, J.-Y. *Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur*, PhD thesis (in French), Université Paris Diderot, Paris 7, 1972.
- [9] Girard, J.-Y., Lafont, Y., and Taylor, P. *Proofs and Types*, Cambridge University Press, 1989.
- [10] Griffin, T. *A formulae-as-type notion of control*, In: F. E. Allen (editor), *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, POPL '90*, ACM Press, 1990.
- [11] de Groote, P. *An environment machine for the $\lambda\mu$ -calculus*, *Mathematical Structures in Computer Science* 8, pp. 637-669, 1998.
- [12] de Groote, P. *On the relation between the $\lambda\mu$ -calculus and the syntactic theory of sequential control*, In: F. Pfenning (editor), *5th International Conference on Logic Programming and Automated Reasoning, LPAR '94*, *Lecture Notes in Artificial Intelligence* (822), pp. 31-43, Springer Verlag, 1994.
- [13] Howard, W. A. *The formulae-as-types notion of construction*, In: Curry, H., Hindley, J. R., and Seldin, J. P. (eds.), *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, pp. 479-490, Academic Press, 1980.
- [14] Krivine, J.-L. *Lambda-calculus types and models*, Ellis Horwood, 1993.
- [15] Murthy, C. R. *An evaluation semantics for classical proofs*, In: *Proceedings of the sixth annual IEEE symposium on logic in computer science*, pp. 96-107, 1991.
- [16] Nour, K. and Saber, K. *A completeness result for the simply typed lambda-mu calculus*, *Annals of Pure and Applied Logic*, 161, pp. 109-118, 2009.
- [17] Nour, K., and Ziadeh, M. *A revised completeness result for the simply typed $\lambda\mu$ -calculus using realizability semantics*, *Logical Methods in Computer Science*, 13(3:13), pp. 1-13, 2017, [https://doi.org/10.23638/LMCS-13\(3:13\)2017](https://doi.org/10.23638/LMCS-13(3:13)2017).
- [18] Parigot, M. *$\lambda\mu$ -calculus: an algorithmic interpretation of classical natural deduction*, In: A. Voronkov (editor), *Logic Programming and Automated Reasoning*, *Lecture Notes in Computer Science* (624), Springer Verlag, Berlin, pp. 190-201, 1992.
- [19] Parigot, M. *Classical proofs as programs*, In: G. Gottlob, A. Leitsch, and D. Mundici (eds.), *Proc. of 3rd Kurt Godel Colloquium, KGC'93*, *Lecture Notes in Computer Science* (713), pp. 263-276, Springer-Verlag, 1993.
- [20] Parigot, M. *Proofs of strong normalization for second order classical natural deduction*, *Journal of Symbolic Logic* (62), pp. 1461-1479, 1997.
- [21] Polonovski, E. *Strong normalization of $\bar{\lambda}\mu\bar{\iota}$ -calculus with explicit substitutions*, In: Walukiewicz, I. (ed.), *Foundations of Software Science and Computation Structures*, 7th International Conference, FOSSACS 2004, *Lecture Notes in Computer Science* (2987), pp. 423-437, Springer Verlag, 2004.
- [22] Py, W. *Confluence en $\lambda\mu$ -calcul*, PhD thesis, University of Chambéry, 1998.

- [23] Rehof, N. J., and Sørensen, M. H. *The λ_{Δ} -calculus*, In: M. Hagiya, J. C. Mitchell (editors), *Theoretical Aspects of Computer Software*, Lecture Notes in Computer Science (789), pp. 516-542, Springer Verlag, 1994.
- [24] Saurin, A. *On the Relations between the Syntactic Theories of $\lambda\mu$ -calculus*, In: M. Kaminski, S. Martini (editors), *17th EACSL Annual Conference on Computer Science Logic*, Lecture Notes in Computer Science (5213), pp. 154-168, Springer Verlag, 2008.
- [25] A. Saurin. *Böhm theorem and Böhm trees for the Lambda-mu-calculus*, *Theoretical Computer Science* 435, pp. 106-138, 2012, <https://doi.org/10.1016/j.tcs.2012.02.027>.
- [26] Sørensen, M. H., and Urzyczyn, P. *Lectures on the Curry-Howard Isomorphism*, Elsevier Science, 2006.
- [27] Tait, W.W., *Intensional interpretation of functionals of finite type I*, *J. Symbolic Logic* 32, pp. 198-212, 1967.