HAL
open science

Strong normalization of lambda-Sym-Prop and lambda bare-mu-mu tilde*-calculi

Péter Battyányi, Karim Nour

To cite this version:

Péter Battyányi, Karim Nour. Strong normalization of lambda-Sym-Prop and lambda bare-mu-mu tilde*-calculi . 2016. hal-01421201

HAL Id: hal-01421201

https://hal.science/hal-01421201v1

Preprint submitted on 21 Dec 2016

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.

Strong normalization of λ^{Sym}_{Prop}- and $\overline{\lambda}\mu\tilde{\mu}^*$-calculi

Péter Battyányi
Department of Computer Science, Faculty of Informatics,
University of Debrecen,
Kassai út 26, 4028 Debrecen, Hungary
email : `battyanyi.peter@inf.unideb.hu`

Karim Nour
LAMA - Équipe LIMD
Université Savoie Mont Blanc
73376 Le Bourget du Lac
email : `karim.nour@univ-smb.fr`

Abstract

In this paper we give an arithmetical proof of the strong normalization of λ^{Sym}_{Prop} of Berardi and Barbanera [1], which can be considered as a formulae-as-types translation of classical propositional logic in natural deduction style. Then we give a translation between the λ^{Sym}_{Prop}-calculus and the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, which is the implicational part of the $\overline{\lambda}\mu\tilde{\mu}$-calculus invented by Curien and Herbelin [3] extended with negation. In this paper we adapt the method of [4] for proving strong normalization. The novelty in our proof is the notion of zoom-in sequences of redexes, which leads us directly to the proofs of the main theorems.

1 Introduction

It was revealed by the works of Murthy [8] and Griffin [6] that the Curry-Howard isomorphism, which establishes a correspondence between natural deduction style proofs in intuitionistic logic and terms of the typed λ-calculus, can be extended to the case of classical logic, as well. Since their discovery many calculi appeared aiming to give an encoding of proofs formulated either in classical natural deduction or in classical sequent calculus.

The $\lambda\mu$-calculus presented by Parigot in [15] finds its origin in the so called Free Deduction (FD) (cf. [15]). Parigot resolves the deterministic nature of intuitionistic natural deduction: unlike in the case of intuitionistic natural deduction, when eliminating an instance of a cut in FD, there can be several choices for picking out the subdeductions to be transformed. By introducing variables of a new kind, the so called μ-variables, Parigot distinguishes formulas that are not active at the moment but the current continuation can be passed over to them. Besides the usual β-reduction, Parigot introduces a new reduction rule called the μ-rule corresponding to structural cut eliminations made necessary by the occurrence of new forms of cuts due to the rule in connection with the μ-variables. The result is a calculus, the $\lambda\mu$-calculus (cf. [13]), which is in relation with classical natural deduction. The μ'-rule is the symmetric counterpart of the μ-rule. It was introduced by Parigot in [14] with the intention of keeping the unicity of representation of data (cf. [11]), the price was, however, that confluence had been lost. In the presence of other simplification rules besides μ and μ', even the strong normalization property is lost (cf. [2]).

Historically, the first calculus reflecting the symmetry of classical propositional logic was the λ^{Sym}_{Prop}-calculus of Berardi and Barbanera [1] establishing a formulae-

as-types connection with natural deduction in classical logic. The calculus λ_{Prop}^{Sym} uses an involutive negation which is not defined as $A \to \bot$. There are negated and non-negated atomic types, and the main connective is not the arrow but the classical $\wedge$ and $\vee$. Berardi and Barbanera make use of the natural symmetry of classical logic expressed by the de Morgan laws in defining negated types. In their paper, Berardi and Barbanera proved that λ_{Prop}^{Sym} is strongly normalizable with a symmetric version of the Tait-Girard reducibility method (cf. [20]). In this paper, leaning on the combinatorial proof applied by David and Nour in [5], we prove that λ_{Prop}^{Sym} is strongly normalizing. The novelty in our proof is the application of so-called zoom-in sequences of redexes, which was inspired by the work in [19]. We prove strong normalizability by verifying that it is closed under substitution. From the assumption that $U[x := V]$ is strongly normalizing and U, V are strongly normalizing, we can identify a subterm U' of a reduct of U such that $U'[x := V]$ also is strongly normalizing. The reduction sequence leading to U' is a so-called zoom-in sequence of redexes: each subsequent element is a subterm of the one-step reduct of the preceding one. We prove that zoom-in sequences have useful invariant properties, which makes it relatively easy for us to set the stage for the main theorem. Due to its intrinsic symmetry in dealing with the typing relation, the λ_{Prop}^{Sym}-calculus also proves to be very close to the calculus named by Nour as classical combinatory logic (CCL). In [12], a calculus of combinators is defined which is equivalent to the full classical propositional logic in natural deduction style. Then a translation is given in both directions between λ_{Prop}^{Sym} and CCL.

Curien and Herbelin introduced the $\overline{\lambda}\mu\tilde{\mu}$-calculus (cf. [3]), which established a correspondence, via the Curry-Howard isomorphism, between classical Gentzen-style sequent calculus and a logical calculus. The $\overline{\lambda}\mu\tilde{\mu}$-calculus possesses a rather strong symmetry: it has right-hand side and left-hand side terms (also referred to as environments). The strong normalization of the calculus was proved by Polonovski [16], and a proof formalizable in first order Peano arithmetic was found by David and Nour (cf. [4]).

As to the connection between the $\lambda\mu$ and the $\overline{\lambda}\mu\tilde{\mu}$-calculus, Curien and Herbelin (cf. [3]) defined a translation both for the call-by-value and the call-by-name part of the $\lambda\mu$-calculus into the $\overline{\lambda}\mu\tilde{\mu}$-calculus. Rocheteau [17] finished this work by defining simulations between the two calculi in both directions. In this paper we define the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, which is the $\overline{\lambda}\mu\tilde{\mu}$-calculus extended with negation, and we describe translations between the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus and the λ_{Prop}^{Sym}-calculus. As a consequence, we obtain that, if one of the calculi is strongly normalizable, then the other one necessary admits this property.

The proof applied in the paper is an adaptation of that in David and Nour [4]. In [4] arithmetic proofs, that is, proofs formalizable in first-order Peano arithmetic, are given for the strong normalizability of the $\overline{\lambda}\mu\tilde{\mu}$- and Parigot's symmetric $\lambda\mu$-calculus. It is demonstrated that the set of strongly normalizable terms are closed under substitution. The goal is achieved by applying implicitly an alternating substitution to find out which part of the substitution would be responsible for being not strongly normalizable provided the basis of the substitution and the terms written in are strongly normalizable. In this paper we reach the same goal by identifying a minimal non strongly normalizing sequence of redexes provided an infinite reduction sequence is given. We call this sequence of redexes a minimal zoom-in reduction sequence. The idea of zoom-in sequence was inspired by [19], where perpetual reduction strategies are defined in order to locate the minimal non strongly normalizing subterms of the elements of an infinite reduction sequence. Again, alternating substitutions are defined inductively starting from two sets of terms, and it is proven that zoom-in reduction sequences do not lead out of these substitutions. With this in hand, the method in [4] can be applied.

We prove the strong normalization of the λ^{Sym}_{Prop}-calculus, though our proof works with some slight modification in the case of the $\overline{\lambda}\mu\tilde{\mu}$-calculus, as well (cf.[2]). However, instead of repeating the proof here, we give a translation of the λ^{Sym}_{Prop}-calculus into the $\overline{\lambda}\mu\tilde{\mu}$-calculus, and vice versa. In fact, to make the connection more visible we define the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, which is the $\overline{\lambda}\mu\tilde{\mu}$-calculus extended with terms expressing negated types. Hence, we also obtain a new proof of strong normalization of the $\overline{\lambda}\mu\tilde{\mu}$-calculus.

The paper is organized as follows. In the first section we introduce the λ^{Sym}_{Prop}-calculus of Berardi and Barbanera, and, as the first step towards strong normalization, prove that the permutation rules can be postponed. In the nest section we show that the β, $\beta^{\perp}$, π and $\pi^{\perp}$ rules together are strongly normalizing. Section 4 introduces the $\overline{\lambda}\mu\tilde{\mu}$-calculus defined by Curien and Herbelin, and we augment the calculus with negation in order to make the comparison of the λ^{Sym}_{Prop}- and the $\overline{\lambda}\mu\tilde{\mu}$-calculi simpler. Section 5 provides translations between the λ^{Sym}_{Prop}- and the $\overline{\lambda}\mu\tilde{\mu}^*$-calculi such that the strong normalization of one of the calculi implies that of the other. The last section contains conclusions with regard to the results of the paper.

2 The λ^{Sym}_{Prop}-calculus

The λ^{Sym}-calculus was introduced by Berardi and Barbanera in [1]. It is organized entirely around the duality in classical logic. It has a negation "built-in": the negation of A is not defined as $A \to \perp$. Each type is rather related to its natural negated type by the notion of duality introduced by negation in classical logic. In fact, Berardi and Barbanera defined a calculus equivalent to first order Peano arithmetic. However, we only consider here its propositional part, denoted by λ^{Sym}_{Prop}, since all the other calculi treated by us in this work are concerned with propositional logic.

Definition 2.1 *The set of types are built from two sets of base types $\mathcal{A} = \{a, b, \ldots\}$ (atomic types) and $\mathcal{A}^{\perp} = \{a^{\perp}, b^{\perp}, \ldots\}$ (negated atomic types).*

1. *The set of m-types is defined by the following grammar*

$$A := \alpha \mid \alpha^{\perp} \mid A \wedge A \mid A \vee A$$

   where α ranges over $\mathcal{A}$ and $\alpha^{\perp}$ over $\mathcal{A}^{\perp}$.

2. *The set of types is defined by the following grammar*

$$C := A \mid \perp.$$

3. *We define the negation of an m-type as follows*

$$(\alpha)^{\perp} = \alpha^{\perp} \qquad (\alpha^{\perp})^{\perp} = \alpha \qquad (A \wedge B)^{\perp} = A^{\perp} \vee B^{\perp} \qquad (A \vee B)^{\perp} = A^{\perp} \wedge B^{\perp}.$$

   In this way we get an involutive negation, i.e. for every m-type A, $(A^{\perp})^{\perp} = A$.

4. *The complexity of a type is defined inductively as follows.*

$$cxty(A) = 0, \text{ if } A \in \mathcal{A} \cup \mathcal{A}^{\perp} \cup \{\perp\}.$$
$$cxty(A_1 \wedge A_2) = cxty(A_1 \vee A_2) = cxty(A_1) + cxty(A_2) + 1.$$

   Then, for every m-type A, $cxty(A) = cxty(A^{\perp})$.

Definition 2.2 *1. We denote by Var the set of term-variables. The set of terms $\mathcal{T}$ of the λ^{Sym}_{Prop}-calculus together with their typing rules are defined as follows. In the definition below the type of a variable must be an m-type and Γ denotes a context (the set of declarations of variables).*

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

$$\langle\,,\rangle \quad \frac{\Gamma \vdash P_1 : A_1 \quad \Gamma \vdash P_2 : A_2}{\Gamma \vdash \langle P_1, P_2\rangle : A_1 \wedge A_2} \qquad \sigma_i \quad \frac{\Gamma \vdash P_i : A_i}{\Gamma \vdash \sigma_i\,(P_i) : A_1 \vee A_2} \; i \in \{1,2\}$$

$$\lambda \quad \frac{\Gamma, x : A \vdash P : \bot}{\Gamma \vdash \lambda x P : A^{\bot}} \qquad \star \quad \frac{\Gamma \vdash P_1 : A^{\bot} \quad \Gamma \vdash P_2 : A}{\Gamma \vdash (P_1 \star P_2) : \bot}$$

2. As usuel, we denote by $Fv(M)$, the set of the free vraribles of the term M.

3. The complexity of a term of $\mathcal{T}$ is defined as follows.

$$cxty(x) = 0,$$
$$cxty(\langle P_1, P_2\rangle) = cxty((P_1 \star P_2)) = cxty(P_1) + cxty(P_2),$$
$$cxty(\lambda x P) = cxty(\sigma_i(P)) = cxty(P) + 1, \textit{ for } i \in \{1,2\}.$$

Definition 2.3 *1. The reduction rules are enumerated below.*

(β)	$(\lambda x P \star Q)$	$\to_{\beta}$	$P[x := Q]$	
$(\beta^{\bot})$	$(Q \star \lambda x P)$	$\to_{\beta^{\bot}}$	$P[x := Q]$	
(η)	$\lambda x(P \star x)$	$\to_{\eta}$	P	*if* $x \notin Fv(P)$
$(\eta^{\bot})$	$\lambda x(x \star P)$	$\to_{\eta^{\bot}}$	P	*if* $x \notin Fv(P)$
(π)	$(\langle P_1, P_2\rangle \star \sigma_i(Q_i))$	$\to_{\pi}$	$(P_i \star Q_i)$	$i \in \{1,2\}$
$(\pi^{\bot})$	$(\sigma_i(Q_i) \star \langle P_1, P_2\rangle)$	$\to_{\pi^{\bot}}$	$(Q_i \star P_i)$	$i \in \{1,2\}$
$(Triv)$	$E[P]$	$\to_{Triv}$	P	$(*)$

(*) *If $E[-]$ is a context with type $\bot$ and $E[-] \neq [-]$, P has type $\bot$ and $E[-]$ does not bind any free variables in P.*

2. Let us take the union of the above rules. Let $\to$ stand for the compatible closure of this union and, as usual, $\to^*$ denote the reflexive, symmetric and transitive closure of $\to$. The notions of reduction sequence, normal form and normalization are defined with respect to $\to$.

3. Let M, N be terms. The length (i.e. the number of steps) of the reduction $M \to^* N$ is denoted by $lg(M \twoheadrightarrow N)$.

We enumerate below some theoretical properties of the λ^{Sym}_{Prop}-calculus (cf. [1] and [9]).

Proposition 2.4 (Type-preservation property) *If $\Gamma \vdash P : A$ and $P \to^* Q$, then $\Gamma \vdash Q : A$.*

Proposition 2.5 (Subformula property) *If Π is a derivation of $\Gamma \vdash P : A$ and P is in normal form, then every type occurring in Π is a subformula of a type occurring in Γ, or a subformula of A.*

Theorem 2.6 (Strong normalization) *If $\Gamma \vdash P : A$, then P is strongly normalizable, i.e. every reduction sequence starting from P is finite.*

Berardi and Barbanera proved Theorem 2.6 for the extension of the λ^{Sym}_{Prop}-calculus equivalent to first-order Peano-arithmetic. The proof of this result by Berardi-Barbanera [1] is based on reducibility candidates, but the definition of the interpretation of a type relies on non-arithmetical fixed-point constructions.

We present a syntactical and arithmetical proof of the strong normalization of the λ^{Sym}_{Prop}-calculus in Section 3. The proof was inspired by [4]. First we establish that the permutation rules η, $\eta^{\perp}$ and $Triv$ can be postponed so that we can restrict our attention uniquely to the rules β, $\beta^{\perp}$, π, and $\pi^{\perp}$.

2.1 Permutation rules

First of all, we prove that the η- and $\eta^{\perp}$-reductions can be postponed w.r.t. β, $\beta^{\perp}$, π, and $\pi^{\perp}$.

Definition 2.7 *1. Let $\lambda_{\beta\pi}$-calculus denote the calculus with only the reduction rules $\rightarrow_{\beta}$, $\rightarrow_{\beta^{\perp}}$, $\rightarrow_{\pi}$, and $\rightarrow_{\pi^{\perp}}$.*

2. Let $\rightarrow_{\beta\pi}$ stand for the union of $\rightarrow_{\beta}, \rightarrow_{\beta^{\perp}}, \rightarrow_{\pi}, \rightarrow_{\pi^{\perp}}$ and let $M \rightarrow_e N$ denote the fact that $M \rightarrow_{\eta} N$ or $M \rightarrow_{\eta^{\perp}} N$.

3. We denote by $\rightarrow_{\beta_0}$ (resp. by $\rightarrow_{\beta_0^{\perp}}$) the β-reduction $(\lambda x M \star N) \rightarrow_{\beta} M[x := N]$ (resp. the $\beta^{\perp}$-reduction $(N \star \lambda x M) \rightarrow_{\beta^{\perp}} M[x := N]$), where x occurs at most once in M.

4. We use the standard notation $\rightarrow^{+}$ and $\rightarrow^{*}$ for the transitive and reflexive, transitive closure of a reduction, respectively.

We examine the behaviour of a $\rightarrow_e$ rule followed by a $\rightarrow_{\beta}$ and a $\rightarrow_{\beta_0}$ rule in Lemmas 2.8 and 2.9.

Lemma 2.8 *If $U \rightarrow_e V \rightarrow_{\beta} W$, then $U \rightarrow_{\beta} V' \rightarrow_e^{*} W$ or $U \rightarrow_{\beta_0} V' \rightarrow_{\beta} W$ for some V'.*

Proof We assume $\rightarrow_e$ is an η-reduction, the proof of the case of $\rightarrow_{\beta}$ is similar. The proof is by induction on $cxty(U)$. The only interesting case is $U = (U_1 \star U_2)$. We consider only some of the subcases.

1. $U_1 = \lambda x(U_3 \star x)$, with $x \notin Fv(U_3)$, and $V = (U_3 \star U_2) \rightarrow_{\beta} U_4[y := U_2] = W$, where $U_3 = \lambda y U_4$. In this case $U = (\lambda x(U_3 \star x) \star U_2) \rightarrow_{\beta_0} (U_3 \star U_2) \rightarrow_{\beta} U_4[y := U_2] = W$, so $\rightarrow_{\eta}\rightarrow_{\beta}$ turns into $\rightarrow_{\beta_0}\rightarrow_{\beta}$.

2. $U_1 = \lambda x U_3$, $U_3 \rightarrow_{\eta} U_4$ and $V = (\lambda x U_4 \star U_2) \rightarrow_{\beta} U_4[x := U_2] = W$. Then $U \rightarrow_{\beta} V' = U_3[x := U_2] \rightarrow_{\eta} U_4[x := U_2] = W$.

3. $U_1 = \lambda x U_3$, $U_2 \rightarrow_{\eta} U_4$ and $V = (\lambda x U_3 \star U_4) \rightarrow_{\beta} U_3[x := U_4] = W$. Then $U \rightarrow_{\beta} V' = U_3[x := U_2] \rightarrow_{\eta}^{*} U_3[x := U_4]$.

□

Lemma 2.9 *If $U \rightarrow_e V \rightarrow_{\beta_0} W$, then $U \rightarrow_{\beta_0} W$ or $U \rightarrow_{\beta_0} V' \rightarrow_e W$ or $U \rightarrow_{\beta_0} V' \rightarrow_{\beta_0} W$ for some V'.*

Proof By induction on $cxty(U)$. We assume $U = (U_1 \star U_2)$ and we consider some of the more interesting cases.

1. $U_1 = \lambda x(U_3 \star x)$, with $x \notin Fv(U_3)$, and $V = (U_3 \star U_2) \rightarrow_{\beta_0} U_4[y := U_2] = W$, where $U_3 = \lambda y U_4$. In this case $U = (\lambda x(U_3 \star x) \star U_2) \rightarrow_{\beta_0} (U_3 \star U_2) \rightarrow_{\beta_0} U_4[y := U_2] = W$, thus $\rightarrow_{\eta}\rightarrow_{\beta_0}$ turns into $\rightarrow_{\beta_0}\rightarrow_{\beta_0}$.

2. $U_1 = \lambda x U_3$, $U_3 \to_\eta U_4$ and $V = (\lambda x U_4 \star U_2) \to_{\beta_0} U_4[x := U_2] = W$. Then $U \to_{\beta_0} V' = U_3[x := U_2] \to_\eta U_4[x := U_2] = W$.

3. $U_1 = \lambda x U_3$, $U_2 \to_\eta U_4$ and $V = (\lambda x U_3 \star U_4) \to_{\beta_0} U_3[x := U_4] = W$. Then $U \to_{\beta_0} V' = U_3[x := U_2] \to_\eta U_3[x := U_4]$ provided x occurs in U_3. Otherwise $U \to_{\beta_0} U_3 = W$.

□

We obtain easily the following lemma on the behaviour of several $\to_e$ rules followed by a $\to_\beta$ or a $\to_{\beta_0}$ rule.

Lemma 2.10 *If $U \to_e^* V \to_{\beta_0} W$, then $U \to_{\beta_0}{}^+ V' \to_e^* W$ for some V', and $lg(U \to_{\beta_0}{}^+ V' \to_e^* W) \leq lg(U \to_e^* V \to_{\beta_0} W)$.*

Proof By induction on $lg(U \to_e^* V \to_{\beta_0} W)$, using Lemma 2.9. □

Lemma 2.11 *If $U \to_e^* V \to_\beta W$, then $U \to_\beta{}^+ V' \to_e^* W$ for some V'.*

Proof By induction on $lg(U \to_e^* V \to_\beta W)$. Use Lemmas 2.8, 2.10. □

Lemma 2.12 *If $U \to_e^* V \to_{\beta^\perp} W$, then $U \to_{\beta^\perp}{}^+ V' \to_e^* W$ for some V'.*

Proof Similar to that of the previous lemma. □

We investigate now how a $\to_e$ rule behaves when followed by a $\to_\pi$ or $\to_{\pi^\perp}$ rule.

Lemma 2.13 *If $U \to_e V \to_\pi W$ (resp. $U \to_e V \to_{\pi^\perp} W$), then $U \to_\pi V' \to_e W$ or $U \to_\pi W$ (resp. $U \to_{\pi^\perp} V' \to_e W$ or $U \to_{\pi^\perp} W$) for some V'.*

Proof Observe that in case of $U \to_e V \to_\pi W$ the following possibilities can occur: either $U = \lambda x(V \star x)$ and $V \to_\pi W$ or $U = (\langle P_1, P_2 \rangle \star \sigma(Q))$ and $V = (\langle P_1', P_2' \rangle \star \sigma(Q'))$, where exactly one of $P_i \to_e P_i'$, $Q \to_e Q'$ holds, the other two terms are left unchanged. From this, the statement easily follows. □

Lemma 2.14 *If $U \to_e^* V \to_{\beta\pi} W$, then $U \to_{\beta\pi}^+ V' \to_e W$ for some V'.*

Proof By Lemmas 2.11, 2.12 and 2.13. □

Lemma 2.15 *If $U \to_e^* V \to_{\beta\pi}^* W$, then $U \to_{\beta\pi}^+ V' \to_e^* W$ for some V'.*

Proof Follows from the previous lemma. □

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

Lemma 2.16 *The η- and the $\eta_\perp$-reductions are strongly normalizing.*

Proof The η- and $\eta_\perp$-reductions on M reduce the complexity of M. □

Definition 2.17 *1. Let $\lambda_{\beta\pi\eta}$-calculus denote the calculus obtained from the $\lambda_{\beta\pi}$-calculus by adding the η- and $\eta^\perp$-reductions to it.*

2. Let $\to_{\beta\pi\eta}$ denote the union of $\to_\beta$, $\to_{\beta^\perp}$, $\to_\pi$, $\to_{\pi^\perp}$, $\to_\eta$ and $\to_{\eta^\perp}$.

3. Assume M is a term strongly normalizable in the $\lambda_{\beta\pi}$-calculus. Then we denote by $\eta_{\beta\pi}(M)$ the length of the longest reduction sequence $\to_{\beta\pi}^*$ starting from M.

Corollary 2.18 *If the $\lambda_{\beta\pi}$-calculus is strongly normalizing, then the $\lambda_{\beta\pi\eta}$-calculus is also strongly normalizing.*

Proof Let M be a term, we prove by induction on $\eta_{\beta\pi}(M)$ that M is strongly normalizable in the $\lambda_{\beta\pi\eta}$-calculus. Assume S is an infinite $\beta\pi\eta$-reduction sequence starting from M. If S begins with a $\to_{\beta\pi}$, then the induction hypothesis applies. In the case when S contains only $\to_e$-reductions, we are done by Lemma 2.16. Otherwise there is an initial subsequent $M \to_e^+ M' \to_{\beta\pi} N$. By Lemma 2.14, we have $M \to_{\beta\pi}^+ M'' \to_e^* N$. Thus, we can apply the induction hypothesis to M''. □

What has remained is to augment the calculus treated so far with the rule $Triv$. For strong normalization, it is enough to show that $\rightarrow_{Triv}$ can be postponed w.r.t. $\rightarrow_{\beta\pi\eta}$.

Lemma 2.19 *If $U \rightarrow^*_{Triv} V \rightarrow_{\beta\pi\eta} W$, then $U \rightarrow^+_{\beta\pi\eta} V' \rightarrow^*_{Triv} W$ for some V'.*

Proof It is enough to prove that if $U \rightarrow_{Triv} V \rightarrow_{\beta\pi\eta} W$, then $U \rightarrow_{\beta\pi\eta} V' \rightarrow_{Triv} W$ for some V'. Observe that if $U = E[V] \rightarrow_{Triv} V \rightarrow_{\beta\pi\eta} W$, then $V : \bot$ and $W : \bot$, from which the statement follows. □

Lemma 2.20 *The reduction $\rightarrow_{Triv}$ is strongly normalizing.*

Proof The reduction $\rightarrow_{Triv}$ on M reduces the complexity of M. □

Corollary 2.21 *If the $\lambda_{\beta\pi}$-calculus is strongly normalizing, then the λ^{Sym}_{Prop}-calculus is also strongly normalizing.*

Proof By Corollary 2.18 and Lemmas 2.19 and 2.20. □

3 Strong normalization of the $\lambda_{\beta\pi}$-calculus

In this section, we give an arithmetical proof for the strong normalization of the $\lambda_{\beta\pi}$-calculus. In the sequel we detail the proofs for the β- and π-reductions only, all the proofs below can be extended with the cases of the $\beta^\bot$- and $\pi^\bot$-reduction rules in a straightforward way. We intend to examine how substitution behaves with respect to strong normalizability. The first milestone in this way is Lemma 3.7. Before stating the lemma, we formulate some auxiliary statements.

Definition 3.1

1. *Let $SN_{\beta\pi}$ denote the set of strongly normalizable terms of the $\lambda_{\beta\pi}$-calculus.*
2. *Let $M \in SN_{\beta\pi}$, then $\eta c(M)$ stands for the pair $\langle \eta_{\beta\pi}(M), cxty(M)\rangle$.*

Lemma 3.2 *Let us suppose $M \in SN_{\beta\pi}$, $N \in SN_{\beta\pi}$ and $(M \star N) \notin SN_{\beta\pi}$. Then there are $P \in SN_{\beta\pi}$, $Q \in SN_{\beta\pi}$ and $R \notin SN_{\beta\pi}$ such that $M \rightarrow^*_{\beta\pi} P$ and $N \rightarrow^*_{\beta\pi} Q$ and $(P \star Q) \rightarrow_{\beta\pi} R$.*

Proof By induction on $\eta c(M) + \eta c(N)$. Assume $M \in SN_{\beta\pi}$, $N \in SN_{\beta\pi}$ and $(M \star N) \notin SN_{\beta\pi}$. When $(M \star N) \rightarrow (M' \star N)$ or $(M \star N) \rightarrow (M \star N')$, then the induction hypothesis applies. Otherwise $(M \star N) \rightarrow P \notin SN_{\beta\pi}$, and we have the result. □

Definition 3.3

1. *A proper term is a term differing from a variable.*
2. *For a type A, Σ_A denotes the set of simultaneous substitutions of the form $[x_1 := N_1, \ldots, x_k := N_k]$ where N_i $(1 \leq i \leq n)$ is proper and has type A.*
3. *A simultaneous substitution $\sigma \in \Sigma_A$ is said to be in $SN_{\beta\pi}$, if, for every $x \in dom(\sigma)$, $\sigma(x) \in SN_{\beta\pi}$ holds.*

Lemma 3.4 *Let M, N be terms such that $M \rightarrow^*_{\beta\pi} N$.*

1. *If $N = \lambda x P$, then $M = \lambda x P_1$ with $P_1 \rightarrow^*_{\beta\pi} P$.*
2. *If $N = \langle P, Q\rangle$, then $M = \langle P_1, Q_1\rangle$ with $P_1 \rightarrow^*_{\beta\pi} P$ and $Q_1 \rightarrow^*_{\beta\pi} Q$.*
3. *If $N = \sigma_i(P)$, then $M = \sigma_i(P_1)$ with $P_1 \rightarrow^*_{\beta\pi} P$, for $i \in \{1, 2\}$.*

Proof Straightforward. □

We remark that in the presence of the $\to_\eta$ and $\to_\eta^\perp$ rules the above lemma would not work. For example, $\lambda x(y \star x) \to_\eta y$.

Lemma 3.5 *If $M \in SN_{\beta\pi}$ and $x \in Var$, then $(M \star x) \in SN_{\beta\pi}$ (resp. $(x \star M) \in SN_{\beta\pi}$).*

Proof Let us suppose $M \in SN_{\beta\pi}$ and $(M \star x) \notin SN_{\beta\pi}$. By Lemma 3.2, we must have $M \to^*_{\beta\pi} \lambda y M_1 \in SN_{\beta\pi}$ such that $(M \star x) \to^*_{\beta\pi} (\lambda y M_1 \star x) \to_{\beta\pi} M_1[y := x]$ and $M_1[y := x] \notin SN_{\beta\pi}$. Being a subterm of a reduct of $M \in SN_{\beta\pi}$, we also have $M_1 \in SN_{\beta\pi}$. Moreover, $M_1[y := x]$ is obtained from M_1 by α-conversion, hence $M_1[y := x] \in SN_{\beta\pi}$, a contradiction. □

Definition 3.6 *1. Let M, N be terms.*

(a) We denote by $M \leq N$ (resp. $M < N$) the fact that M is a sub-term (resp. a strict sub-term) of N.

(b) We denote by $M \prec N$ the fact that $M \leq P$ for some P such that $N \to^*_{\beta\pi} P$ and either $N \to^+_{\beta\pi} P$ or $M < P$. We denote by $\preceq$ the reflexive closure of $\prec$.

(c) Let R be a $\beta\pi$-redex. We write $M \to^R N$ if N is the term M after the reduction of R.

2. Let $\mathcal{R} = [R_1, \ldots, R_n]$ where R_i is a $\beta\pi$-redex $(1 \leq i \leq n)$. Then $\mathcal{R}$ is called zoom-in if, for every $1 \leq i < n$, $R_i \to^{R_i} R_i'$ and $R_{i+1} \leq R_i'$. Moreover, $\mathcal{R}$ is minimal, if, for each $R_i = (P_i \star Q_i)$, we have P_i, $Q_i \in SN_{\beta\pi}$ and $(P_i \star Q_i) \notin SN_{\beta\pi}$. We write $M \to^{\mathcal{R}} N$, if $M \to^{R_1} \ldots \to^{R_n} N$.

For the purpose of proving the strong normalization of the calculus, it is enough to show that the set of strongly normalizable terms is closed under substitution. To this end, we show that, if U, $S \in SN_{\beta\pi}$ and $U[x := S] \notin SN_{\beta\pi}$, then there is a term $W \leq U$ of a special form such that $W \in SN_{\beta\pi}$ and $W[x := S] \notin SN_{\beta\pi}$. Moreover, we show that the sequence of redexes leading to W is not completely general, this is a zoom-in sequence defined below. Reducing the outermost redexes of a zoom-in sequence preserve some useful properties, which is the statement of Lemma 3.9.

Lemma 3.7 *Let U, $S \in SN_{\beta\pi}$ and suppose $U[x := S] \notin SN_{\beta\pi}$. Then there are terms $P, V \preceq U$ and a zoom-in minimal $\mathcal{R}$ such that $U[x := S] \to^{\mathcal{R}} V[x := S]$, $(x \star P) \leq V$ (or $(P \star x) \leq V$), $P[x := S] \in SN_{\beta\pi}$ and $(x \star P)[x := S] \notin SN_{\beta\pi}$ (or $(P \star x)[x := P] \notin SN_{\beta\pi}$).*

Proof The proof goes by induction on $\eta c(U)$. If U is other than an application, we can apply the induction hypothesis. Assume $U = (U_1 \star U_2)$ with $U_i[x := S] \in SN_{\beta\pi}$ $(i \in \{1, 2\})$ and $U[x := S] \notin SN_{\beta\pi}$. By Lemma 3.2 and the induction hypothesis we may assume that $(U_1 \star U_2)[x := S] \to_\rho U' \notin SN_{\beta\pi}$, where $\rho \in \{\beta, \beta^\perp, \pi, \pi^\perp\}$. Let us suppose $\rho = \beta$, the other cases can be treated similarly. If $U_1 = \lambda y U_1'$, then the induction hypothesis applies to $U_1'[y := U_2]$. Otherwise $U_1 = x$, and we have obtained the result. □

Next we define an alternating substitution: we start from two sets of terms of complementer types and the substitution is defined in a way that we keep track of which newly added sets of substitutions come from which of the two sets. The reason for this is that Lemma 3.7 in itself is not enough for proving the strong normalizability of λ^{Sym}_{Prop} even if we would consider the β and $\beta^\perp$ rules alone. We have to show that, if we start from a term $(U_1 \star U_2)$, where U_1 and $U_2 \in SN_{\beta\pi}$ and we assume that $U_1[x := U_2] \notin SN_{\beta\pi}$, then there are no deep interactions between the terms which come from U_1 and from U_2. We can identify a subterm of a reduct of U_1 which is the cause for being non $SN_{\beta\pi}$, when performing a substitution with U_2.

Definition 3.8 *1. A set $\mathcal{A}$ of proper terms is called $\preceq$-closed from below if, for all terms U, U', if $U' \preceq U$, $U \in \mathcal{A}$ and U' is proper, then $U' \in \mathcal{A}$.*

2. Let $\mathcal{A}, \mathcal{B}$ be sets $\preceq$-closed from below and A a type. We define simultaneously two sets of substitutions

(a) $\Pi_A(\mathcal{B}) \subseteq \Sigma_A$ and $\Theta_{A^\perp}(\mathcal{A}) \subseteq \Sigma_{A^\perp}$ as follows.

- $\emptyset \in \Pi_A(\mathcal{B})$,
- $[y_1 := V_1\tau_1, \ldots, y_m := V_m\tau_m] \in \Pi_A(\mathcal{B})$ *if* $V_i \in \mathcal{B}$ *such that* $type(V_i) = A$ *and* $\tau_i \in \Theta_{A^\perp}(\mathcal{A})$ $(1 \leq i \leq m)$.
- $\emptyset \in \Theta_{A^\perp}(\mathcal{A})$.
- $[x_1 := U_1\rho_1, \ldots, x_m := U_m\rho_m] \in \Theta_{A^\perp}(\mathcal{A})$ *if* $U_i \in \mathcal{A}$ *such that* $type(U_i) = A^\perp$ *and* $\rho_i \in \Pi_A(\mathcal{B})$ $(1 \leq i \leq m)$.

(b) Let $\mathcal{S}_A(\mathcal{A}, \mathcal{B}) = \{U\rho \mid U \in \mathcal{A} \text{ and } \rho \in \Pi_A(\mathcal{B})\} \cup \{V\tau \mid V \in \mathcal{B} \text{ and } \tau \in \Theta_{A^\perp}(\mathcal{A})\}$. It is easy to see that, from $U \leq V$ and $V \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$, it follows $U \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$.

Lemma 3.9 *Let n be an integer, A a type of length n and $\mathcal{R} = [R_1, \ldots, R_m]$ a zoom-in minimal sequence of redexes. Assume the property H "if U, $V \in SN_{\beta\pi}$ and $cxty(type(V)) < n$, then $U[x := V] \in SN_{\beta\pi}$" holds. If $R_1 \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$ for some sets $\mathcal{A}$ and $\mathcal{B}$ $\preceq$-closed from below, then $R_m \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$.*

Proof The proof goes by induction on m. We prove the induction step from $m = 1$ to $m = 2$, the proof is the same when $m \in \mathbb{N}$ is arbitrary. We only treat the more interesting cases. Assume $R_1 \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$.

1. $R_1 = (\lambda x Q \star S) \to_\beta R_1' = Q[x := S]$ and $R_2 \leq R_1'$.

   (a) Suppose $R_1 = U\rho$ for some $U \in \mathcal{A}$ and $\rho \in \Pi_A(\mathcal{B})$. Then $U = (U_1 \star U_2)$ with $U_1\rho = \lambda x Q$ and $U_2\rho = S$, and, since $\rho \in \Sigma_A$, U_1 must be proper. Then we have $U_1 = \lambda x U_1'$ and $U_1'\rho = Q$ for some U_1'. Now, $R_1' = U_1'[x := U_2]\rho \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$, which yields $R_2 \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$.

   (b) Assume now $R_1 = V\tau$. Then $V = (V_1 \star V_2)$ with $V_1\tau = \lambda x Q$ and $V_2\tau = S$, and, since $\tau \in \Sigma_{A^\perp}$, V_2 must be proper. If V_1 is proper, then, as before, we obtain the result. Otherwise $V_1\tau = U\rho = \lambda x Q$. Since $U \in \mathcal{A}$ is proper, $U = \lambda x U_1$ and $U_1\rho = Q$ for some U_1. Then $U_1\rho_1 \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$ with $\rho_1 = \rho + [x := V_2\tau]$, since $type(V_2\tau) = type(S) = A$. This implies $R_2 \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$.

2. $R_1 = (\langle Q_1, Q_2 \rangle \star \sigma_1(S)) \to_\pi (Q_1 \star S) = R_1'$ and $R_2 \leq R_1'$.

   (a) Assume $R_1 = U\rho$ for some $U \in \mathcal{A}$ and $\rho \in \Pi_A(\mathcal{B})$. Then $U_1\rho = \langle Q_1, Q_2 \rangle$ and $U_2\rho = \sigma_1(S)$.

   - Let U_1 and U_2 be proper. Then $U_1 = \langle U_1', U_1'' \rangle$ and $U_2 = \sigma_1(U_2')$ such that $U_1'\rho = Q_1$, $U_1''\rho = Q_2$ and $U_2'\rho = S$. We have $R_1' = (U_1' \star U_2')\rho \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$, which yields the result.
   - Assume $U_2 \in Var$. Then $V\tau = \sigma(S)$, and $cxty(type(S)) < cxty(type(\sigma S)) = n$. Then assumption H and the fact that $\langle Q_1, Q_2 \rangle \in SN_{\beta\pi}$, together with Lemma 3.5, lead to $(Q_1 \star S) \in SN_{\beta\pi}$, which is not possible. Since $\rho \in \Sigma_A$, $U_1 \in Var$ is impossible.

   (b) Assume $R_1 = V\tau$ for some $V \in \mathcal{B}$ and $\tau \in \Theta_{A^\perp}(\mathcal{A})$. Then $V_1\tau = \langle Q_1, Q_2 \rangle$ and $V_2\tau = \sigma_1(S)$, where $V = (V_1 \star V_2)$.

   - Let V_1 and V_2 be proper. Then $V_1 = \langle V_1', V_1'' \rangle$ and $V_2 = \sigma_1(V_2')$ such that $V_1'\tau = Q_1$, $V_1''\tau = Q_2$ and $V_2'\tau = S$. We have $R_1' = (V_1' \star V_2')\tau \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$.

- Assume $V_1 \in Var$. Then $U\rho = \langle Q_1, Q_2 \rangle$, where $cxty(type(Q_1)) < cxty(type(\langle Q_1, Q_2 \rangle)) = n$. Then assumption H and the fact that $S \in SN_{\beta\pi}$, together with Lemma 3.5, lead to $(Q_1 \star S) \in SN_{\beta\pi}$, which is not possible. Since $\tau \in \Sigma_{A^\perp}$, the case of $V_2 \in Var$ is impossible.

□

The next lemma identifies the subterm of U being responsible for the non strong normalizability of $U[x := V]$.

Lemma 3.10 *Let n be an integer and A a type of length n. Assume the property H "if U, $V \in SN_{\beta\pi}$ and $cxty(type(V)) < n$, then $U[x := V] \in SN_{\beta\pi}$" holds.*

1. *Let U be a proper term, $\sigma \in \Sigma_A$ and $a \notin Im(\sigma)$. If $U\sigma, P \in SN_{\beta\pi}$ and $U\sigma[a := P] \notin SN_{\beta\pi}$, then there exists U' such that $(U' \star a) \preceq U$ and $\sigma' \in \Sigma_A$ such that $U'\sigma' \in SN_{\beta\pi}$ and $(U'\sigma' \star a)[a := P] \notin SN_{\beta\pi}$.*

2. *Let U be a proper term, $\sigma \in \Sigma_{A^\perp}$ and $a \notin Im(\sigma)$. If $U\sigma, P \in SN_{\beta\pi}$ and $U\sigma[a := P] \notin SN_{\beta\pi}$, then there exists U' such that $(a \star U') \preceq U$ and $\sigma' \in \Sigma_{A^\perp}$ such that $U'\sigma' \in SN_{\beta\pi}$ and $(a \star U'\sigma')[a := P] \notin SN_{\beta\pi}$.*

Proof Let us consider only case (1). We identify the reason of $U\sigma[a := P]$ being non strongly normalizable, we find a subterm $(U' \star a)$ of a reduct of U such that, for a substituted instance of $(U' \star a)$, $(U' \star a)\sigma' \in SN_{\beta\pi}$ and $(U' \star a)\sigma'[a := P] \notin SN_{\beta\pi}$. This will contradict by some minimality assumption concerning U in the next lemma. For this we define two sets of substitutions as in Definition 3.8 with the sets $\mathcal{A}$ and $\mathcal{B}$ as below. We note that Property H of the previous lemma implicitly ensures that the type of U and the type of the elements in σ can be of the same lengths.

Let

$$\mathcal{A} = \{M \mid M \preceq U \text{ and } M \text{ is proper}\},$$

$$\mathcal{B} = \{V \mid V \preceq \sigma(b) \text{ for some } b \in dom(\sigma) \text{ and } V \text{ is proper}\}.$$

Then $U\sigma \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$. By Lemma 3.7, there exists a minimal zoom-in $\mathcal{R} = [R_1, \ldots, R_n]$ and there are terms U^* and $V \preceq U\sigma$ such that $U\sigma[a := P] \to^{\mathcal{R}} V[a := P]$ and $(U^* \star a) \leq V$ and $(U^* \star a) \in SN_{\beta\pi}$ and $(U^* \star a)[a := P] \notin SN_{\beta\pi}$ or $(a \star U^*) \leq V$ and $(a \star U^*) \in SN_{\beta\pi}$ and $(a \star U^*)[a := P] \notin SN_{\beta\pi}$. Assume the former. By Lemma 3.9, $(U^* \star a) \in \mathcal{S}_A(\mathcal{A}, \mathcal{B})$. Then $(U^* \star a) = S\rho$ for some $S \in \mathcal{A}$ or $(U^* \star a) = W\tau$ for some $W \in \mathcal{B}$. Since $a \notin Im(\sigma)$, the latter is impossible. The former case, however, yields $S = (U' \star a)$ with $U'\rho = U^*$ for some $U' \in \mathcal{A}$, which proves our assertion. □

The next lemma states closure of strong normalizability under substitution.

Lemma 3.11 *If $M, N \in SN_{\beta\pi}$, then $M[x := N] \in SN_{\beta\pi}$.*

Proof We are going to prove a bit more general statement. Suppose $M, N_i \in SN_{\beta\pi}$ are proper, $type(N_i) = A$ $(1 \leq i \leq k)$. Let $\tau_i \in \Sigma_{A^\perp}$ are such that $\tau_i \in SN_{\beta\pi}$ $(1 \leq i \leq k)$ and let $\rho = [x_1 := N_1\tau_1, \ldots, x_k := N_k\tau_k]$. Then we have $M\rho \in SN_{\beta\pi}$. The proof is by induction on $(cxty(A), \eta_{\beta\pi}(M), cxty(M), \Sigma_i\, \eta_{\beta\pi}(N_i), \Sigma_i\, cxty(N_i))$ where, in $\Sigma_i\, \eta_{\beta\pi}(N_i)$ and $\Sigma_i\, cxty(N_i)$, we count each occurrence of the substituted variable. For example if $k = 1$ and x_1 has n occurrences, then $\Sigma_i\, \eta_{\beta\pi}(N_i) = n \cdot \eta_{\beta\pi}(N_1)$.

The only nontrivial case is when $M = (M_1 \star M_2)$ and $M\rho \notin SN_{\beta\pi}$. By the induction hypothesis $M_i\rho \in SN_{\beta\pi}$ $(i \in \{1, 2\})$. We select some of the typical cases.

(A) $M_1\rho \to_{\beta\pi} \lambda z M'$ and $M'[z := M_2] \notin SN_{\beta\pi}$.

1. M_1 is proper, then there is an M_3 such that $M_1 = \lambda z M_3$ and $M_3\rho \to_{\beta\pi} M'$. In this case $(M_3[z := M_2])\rho \notin SN_{\beta\pi}$ and since $\eta_{\beta\pi}(M_3[z := M_2]) < \eta_{\beta\pi}(M)$, the induction hypothesis gives the result.
2. $M_1 \in Var$. Then $M_1 = x \in dom(\rho)$, $\rho(x) = N_j\tau_j \to_{\beta\pi} \lambda z M'$ for some $(1 \leq j \leq k)$. Since N_j is proper, there is an N' such that $N_j = \lambda z N'$, $N'\tau_j \to_{\beta\pi} M'$. Then $N'\tau_j[z := M_2\rho] \notin SN_{\beta\pi}$ and $type(z) = type(N_j)^{\perp} = type(\tau_j)$, so, by the previous lemma, we have $N'' \prec N'$ and τ' such that $(N''\tau' \star M_2\rho) \notin SN_{\beta\pi}$. Now we have $(N''\tau' \star M_2\rho) = (y \star M_2\rho)[y := N''\tau']$, $type(N'') = type(\tau')^{\perp} = A$ and $\eta_{\beta\pi} cxty(N'') < \eta c(N_j)$, which contradicts the induction hypothesis.

(B) $M_1\rho \to_{\beta\pi} \langle M', M''\rangle$ and either $(M' \star M_2) \notin SN_{\beta\pi}$ or $(M'' \star M_2) \notin SN_{\beta\pi}$. Suppose the former.

1. M_1, M_2 are proper, then there are M_3, M_4 such that $M_1 = \langle M_3, M_4\rangle$ and $M_3\rho \to_{\beta\pi} M'$, or $M_4\rho \to_{\beta\pi} M''$. Assume the former. Then we have $(M_3 \star M_2)\rho \notin SN_{\beta\pi}$ and $\eta_{\beta\pi}((M_3 \star M_2)) < \eta_{\beta\pi}(M)$, a contradiction.
2. $M_1 = x \in dom(\rho)$, then $\rho(x) = N_j\tau_j \to_{\beta\pi} \langle M', M''\rangle$, N_j is proper and $N_j = \langle U, V\rangle$, $U\tau_j \to_{\beta\pi} M'$ or $V\tau_j \to_{\beta\pi} M''$. Now $(U\tau_j \star M_2) = (y \star M_2)[y := U\tau_j] \notin SN_{\beta\pi}$, but $cxty(type(U)) < cxty(type(N_j))$, a contradiction again.
3. $M_2 \in Var$. This is similar to the previous case. By the same argument as in part (A)-2.-(a) of the proof of the previous lemma, M_1 and M_2 cannot be both variables. This completes the proof of the lemma.

□

Theorem 3.12 *The $\lambda_{\beta\pi}$-calculus is strongly normalizing.*

Proof It is enough to show that, for every term, $M, N \in SN_{\beta\pi}$ implies $(M \star N) \in SN_{\beta\pi}$. Supposing $M, N \in SN_{\beta\pi}$, Lemma 3.5 gives $(M \star x) \in SN_{\beta\pi}$, which yields, by the previous lemma, $(M \star N) = (M \star x)[x := N] \in SN_{\beta\pi}$. □

4 The $\overline{\lambda}\mu\tilde{\mu}$- and the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus

In this section we introduce the $\overline{\lambda}\mu\tilde{\mu}$-calculus together with one of its extensions, the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, by which we establish a translation of the λ^{Sym}_{Prop}-calculus and thus obtain the strong normalization of the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus as a consequence.

4.1 The $\overline{\lambda}\mu\tilde{\mu}$-calculus

The $\overline{\lambda}\mu\tilde{\mu}$-calculus was introduced by Curien and Herbelin (cf. [10] and [3]). We examine here the calculus defined in [3], which is a simply typed one. The $\overline{\lambda}\mu\tilde{\mu}$-calculus was invented for representing proofs in classical Gentzen-style sequent calculus: under the Curry-Howard correspondence a version of Gentzen-style sequent calculus is obtained as a system of simple types for the $\overline{\lambda}\mu\tilde{\mu}$-calculus. Moreover, the system presents a clear duality between call-by-value and call-by-name evaluations.

Definition 4.1 *There are three kinds of terms, defined by the following grammar, and there are two kinds of variables. We assume that we use the same set of variables in the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, too. In the literature, different authors use different terminology. Here, we will call them either c-terms, or l-terms or r-terms. Similarly,*

the variables will be called either l-variables (and denoted as $x, y, ...$) or r-variables (and denoted as $a, b, ...$).

$$\begin{array}{lcl} p & ::= & \lfloor t, e \rfloor \\ t & ::= & x \quad | \quad \lambda x t \quad | \quad \mu \alpha p \\ e & ::= & \alpha \quad | \quad (t.e) \quad | \quad \tilde{\mu} x p \end{array}$$

As usual, we denote by $Fv(u)$, the set of the free variables of the term u.

Definition 4.2 *The types are built from atomic formulas (or, in other words, atomic types) with the connector $\rightarrow$. We assume that the same set of type variables $\mathcal{A}$ is used in the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, also. The typing system is a sequent calculus based on judgements of the following form.*

$$p : (\Gamma \ \rhd \ \triangle) \qquad \Gamma \ \rhd \ t : A \mid \triangle \qquad \Gamma \mid e : A \ \rhd \ \triangle$$

where Γ (resp. $\triangle$) is a set of declarations of the form $x : A$ (resp. $a : A$), x (resp. a) denoting a l-variable (resp. an r-variable) and A representing a type, such that x (resp. a) occurs at most once in an expression of Γ (resp. $\triangle$) of the form $x : A$ (resp. $a : A$). We say that Γ an l-context and $\triangle$ is an r-context, respectively. The typing rules are as follows

$$Var_1 \ \frac{}{\Gamma, x : A \ \rhd \ x : A \mid \triangle} \qquad Var_2 \ \frac{}{\Gamma \mid \alpha : A \ \rhd \ \alpha : A, \triangle}$$

$$\lambda \ \frac{\Gamma, x : A \ \rhd \ t : B \mid \triangle}{\Gamma \ \rhd \ \lambda x t : A \rightarrow B \mid \triangle} \qquad (.) \ \frac{\Gamma \ \rhd \ t : A \mid \triangle \quad \Gamma \mid e : B \ \rhd \ \triangle}{\Gamma \mid (t.e) : A \rightarrow B \ \rhd \ \triangle}$$

$$\lfloor , \rfloor \ \frac{\Gamma \ \rhd \ t : A \mid \triangle \quad \Gamma \mid e : A \ \rhd \ \triangle}{\lfloor t, e \rfloor : (\Gamma \ \rhd \ \triangle)}$$

$$\mu \ \frac{p : (\Gamma \ \rhd \ \alpha : A, \triangle)}{\Gamma \ \rhd \ \mu \alpha p : A \mid \triangle} \qquad \tilde{\mu} \ \frac{p : (\Gamma, x : A \ \rhd \ \triangle)}{\Gamma \mid \tilde{\mu} x p : A \ \rhd \ \triangle}$$

Definition 4.3 *The cut-elimination procedure (on the logical side) corresponds to the reduction rules (on the terms) given below.*

$$\begin{array}{llll} (\lambda) & \lfloor \lambda x t, (t'.e) \rfloor & \hookrightarrow_{\lambda} & \lfloor t', \tilde{\mu} x \lfloor t, e \rfloor \rfloor \\ (\mu) & \lfloor \mu \alpha p, e \rfloor & \hookrightarrow_{\mu} & p[\alpha := e] \\ (\tilde{\mu}) & \lfloor t, \tilde{\mu} x p \rfloor & \hookrightarrow_{\tilde{\mu}} & p[x := t] \\ (s_l) & \mu \alpha \lfloor t, \alpha \rfloor & \hookrightarrow_{s_l} & t \qquad \text{if } \alpha \notin Fv(t) \\ (s_r) & \tilde{\mu} x \lfloor x, e \rfloor & \hookrightarrow_{s_r} & e \qquad \text{if } x \notin Fv(e) \end{array}$$

Let us take the union of the above rules. Let $\hookrightarrow$ stand for the compatible closure of this union and, as usual, $\hookrightarrow^*$ denote the reflexive, symmetric and transitive closure of $\hookrightarrow$. The notions of reduction sequence, normal form and normalization are defined with respect to $\hookrightarrow$.

We present below some theoretical properties of the $\overline{\lambda}\mu\tilde{\mu}$-calculus (cf. [10], [3], [9], [16] and [5]).

Proposition 4.4 (Type-preservation property) *If $\Gamma \ \rhd \ t : A \mid \triangle$ (resp. $\Gamma \mid e : A \ \rhd \ \triangle$, resp. $p : (\Gamma \ \rhd \ \triangle)$) and $t \hookrightarrow^* t'$ (resp. $e \hookrightarrow^* e'$, resp. $p \hookrightarrow^* p'$), then $\Gamma \ \rhd \ t' : A \mid \triangle$ (resp. $\Gamma \mid e' : A \ \rhd \ \triangle$, resp. $p' : (\Gamma \ \rhd \ \triangle)$).*

Proposition 4.5 (Subformula property) *If Π is a derivation of $\Gamma \ \rhd \ t : A \mid \triangle$ (resp. $\Gamma \mid e : A \ \rhd \ \triangle$, resp. $p : (\Gamma \ \vdash \ \triangle)$) and t (resp. e, resp. p) is in normal form, then every type occurring in Π is a subformula of a type occurring in $\Gamma \cup \triangle$, or a subformula of A (only for t and e).*

Theorem 4.6 (Strong normalization property) *If $\Gamma \rhd t : A \mid \triangle$ (resp. $\Gamma \mid e : A \rhd \triangle$, resp. $p : (\Gamma \rhd \triangle)$), then t (resp. e, resp. p) is strongly normalizable, i.e. evrey reduction sequence starting from t (resp. e, resp. p) is finite.*

The proof of Theorem 4.6 can be found in [16], as well as in [5], where an arithmetical proof is presented.

4.2 The $\overline{\lambda}\mu\tilde{\mu}^*$-calculus

Since we work in a sequent calculus, where dealing with negation is implicitly built in the rules, the typing rules of the $\overline{\lambda}\mu\tilde{\mu}$-calculus do not handle negation. However, for a full treatment of propositional logic we found it more convenient to introduce rules concerning negation. Since c-terms, which could have been candidates for objects of type $\perp$, are distinctly separated from terms, adding new term- and type-forming operators seems to be the easiest way to define negation.

Definition 4.7 *1. The terms of the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus are defined by the following grammar.*

$$\begin{array}{rcllllll} p & ::= & \lfloor t, e \rfloor & & & & & \\ t & ::= & x & \mid & \lambda x\, t & \mid & \mu\alpha\, p & \mid & \overline{e} \\ e & ::= & \alpha & \mid & (t.e) & \mid & \tilde{\mu} x\, p & \mid & \widetilde{t} \end{array}$$

As an abuse of terminology, in the sequel when speaking about the syntactic elements of the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, we may not distinguish l-, r- and c-terms, we may speak about terms in general. We denote by $\mathfrak{T}$ the set of terms of the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus.

2. The complexity of a term of $\mathfrak{T}$ is defined as follows.

$$\begin{aligned} &cxty(x) = cxty(\alpha) = 0, \\ &cxty(\lambda x t) = cxty(\widetilde{t}) = cxty(t) + 1, \\ &cxty(\overline{e}) = cxty(e) + 1, \\ &cxty(\mu\alpha p) = cxty(\tilde{\mu} x\, p) = cxty(p) + 1, \\ &cxty(\lfloor t, e \rfloor) = cxty((t.e)) = cxty(t) + cxty(e). \end{aligned}$$

Definition 4.8 *The type inference rules are the same as in the $\overline{\lambda}\mu\tilde{\mu}$-calculus with two extra rules added for the types of the complemented terms. Moreover, we introduce an equation between types (for all types A, $(A^\perp)^\perp = A$) to ensure that our negation is involutive.*

$$\overline{\cdot}\ \frac{\Gamma \mid e : A \rhd \triangle}{\Gamma \rhd \overline{e} : A^\perp \mid \triangle} \qquad \widetilde{\cdot}\ \frac{\Gamma \rhd t : A \mid \triangle}{\Gamma \mid \widetilde{t} : A^\perp \rhd \triangle}$$

We also define the complexity of types in the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus.

$$cxty(A) = 0 \text{ for atomic types,}$$

$$cxty(A \to B) = cxty(A) + cxty(B) + 1,$$

$$cxty(A^\perp) = cxty(A).$$

That is, the complexity of a type A provides us with the number of arrows in A. The presence of negation makes it necessary for us to introduce new rules handling negation.

Definition 4.9 *Besides the reduction rules already present in $\overline{\lambda}\mu\tilde{\mu}$, we endow the calculus with some more new rules to handle the larger set of terms. In what follows cl stands for the name: complementer rule. We shall denote the $cl_{1,l}$- and $cl_{1,r}$-rules with a common notation as the cl_1-rules.*

$$\begin{array}{llll} (cl_{1,l}) & \overline{\widetilde{t}} & \hookrightarrow_{cl_{1,l}} & t \\ (cl_{1,r}) & \widetilde{\overline{e}} & \hookrightarrow_{cl_{1,r}} & e \\ (cl_2) & \lfloor \overline{e}, \widetilde{t} \rfloor & \hookrightarrow_{cl_2} & \lfloor t, e \rfloor \end{array}$$

In the sequel, we continue to apply the notation $\hookrightarrow$ and $\hookrightarrow^*$ in relation with this new calculus.

Obviously, the statements analogous to Propositions 4.4 and 4.5 are still valid.

5 Relating the λ^{Sym}_{Prop}-calculus to the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus

In [17], Rocheteau defined a translation between the $\overline{\lambda}\mu\tilde{\mu}$-calculus and the $\lambda\mu$-calculus, treating both a call-by-value and a call-by-name aspect of $\overline{\lambda}\mu\tilde{\mu}$. In this subsection, we give a translation (in both directions) between the λ^{Sym}_{Prop}-calculus and the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, which is a version of the $\overline{\lambda}\mu\tilde{\mu}$-calculus extended with negation. The translations are such that strong normalization of one calculus follows from that of the other in both directions. We omit issues of evaluation strategies, however. In the end of the section we give an exact description of the correspondence between the two translations. Preparatory to presenting the translations, let us introduce some definitions and notation below. We assume that the two calculi have the same sets of variables and atomic types. Moreover, as an abuse of notation, if $\overline{\alpha} : A^{\perp}$ stems form the r-variable $\alpha : A$ in the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, then we suppose that in the λ^{Sym}_{Prop}-calculus $\overline{\alpha}$ denotes a variable with type $A^{\perp}$.

5.1 A translation of the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus into the λ^{Sym}_{Prop}-calculus

Definition 5.1 *1. Let us consider the λ^{Sym}_{Prop}-calculus. For $i \in \{1, 2\}$, we write $\pi_i(y) = \lambda z(y \star \sigma_i(z))$. Then, we can observe that $y : A_1 \wedge A_2 \vdash \pi_i(y) : A_i$, for $i \in \{1, 2\}$.*

2. We define a translation $.^{\mathfrak{e}} : \mathfrak{T} \longrightarrow \mathcal{T}$ as follows.

$$p^{\mathfrak{e}} = (u^{\mathfrak{e}} \star v^{\mathfrak{e}}) \quad \textit{if} \quad p = \lfloor v, u \rfloor.$$

$$t^{\mathfrak{e}} = \begin{cases} x & \textit{if} \quad t = x, \\ \lambda y(\lambda x(\pi_2(y) \star u^{\mathfrak{e}}) \star \pi_1(y)) & \textit{if} \quad t = \lambda x u, \\ \lambda x(e^{\mathfrak{e}} \star t^{\mathfrak{e}}) & \textit{if} \quad t = \tilde{\mu} x \lfloor t, e \rfloor, \\ u^{\mathfrak{e}} & \textit{if} \quad t = \overline{u}. \end{cases}$$

$$e^{\mathfrak{e}} = \begin{cases} \overline{\alpha} & \textit{if} \quad e = \alpha, \\ \langle t^{\mathfrak{e}}, h^{\mathfrak{e}} \rangle & \textit{if} \quad e = t.h, \\ \lambda \overline{\alpha}(e^{\mathfrak{e}} \star t^{\mathfrak{e}}) & \textit{if} \quad e = \mu \alpha \lfloor t, e \rfloor, \\ h^{\mathfrak{e}} & \textit{if} \quad e = \widetilde{h}. \end{cases}$$

3. The translation $.^{\mathfrak{e}}$ also applies to types.

- *$A^{\mathfrak{e}} = A$, where A is an atomic type,*
- *$(A^{\perp})^{\mathfrak{e}} = (A^{\mathfrak{e}})^{\perp}$,*
- *$(A \to B)^{\mathfrak{e}} = (A^{\mathfrak{e}})^{\perp} \vee B^{\mathfrak{e}}$.*

4. *Let* Γ, $\triangle$ *be* l*- and* r*-contexts, respectively. Then* $\Gamma^{\mathfrak{e}} = \{x : A^{\mathfrak{e}} \mid x : A \in \Gamma\}$ *and similarly for* $\triangle$. *Furthermore, for any* r*-context* $\triangle$, *let* $\triangle^{\perp} = \{\overline{\alpha} : A^{\perp} \mid \alpha : A \in \triangle\}$.

Lemma 5.2 1. *If* $\Gamma \rhd t : A \mid \triangle$, *then* $\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash t^{\mathfrak{e}} : A^{\mathfrak{e}}$.

2. *If* $\Gamma \mid e : A \rhd \triangle$, *then* $\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash e^{\mathfrak{e}} : (A^{\mathfrak{e}})^{\perp}$.

3. *If* $p : (\Gamma \rhd \triangle)$, *then* $\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash p^{\mathfrak{e}} : \perp$.

Proof The above statements are proved simultaneously according to the length of the $\overline{\lambda}\mu\tilde{\mu}^*$-deduction. We remark that $.^{\mathfrak{e}}$ is defined in Definition 5.1 exactly in the way to make the assertions of the lemma true. Let us examine some of the more interesting cases.

1. Suppose
$$\frac{\Gamma, x : A \rhd u : B \mid \triangle}{\Gamma \rhd \lambda x u : A \to B \mid \triangle}.$$
Then we have, by the induction hypothesis and Notation 5.1,
$$\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp}, x : A^{\mathfrak{e}}, y : A^{\mathfrak{e}} \wedge (B^{\mathfrak{e}})^{\perp} \vdash u^{\mathfrak{e}} : B^{\mathfrak{e}},$$
$$\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp}, y : A^{\mathfrak{e}} \wedge (B^{\mathfrak{e}})^{\perp} \vdash \pi_1(y) : A^{\mathfrak{e}},$$
$$\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp}, y : A^{\mathfrak{e}} \wedge (B^{\mathfrak{e}})^{\perp} \vdash \pi_2(y) : (B^{\mathfrak{e}})^{\perp}.$$
Thus we can conclude
$$\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp}, x : A^{\mathfrak{e}}, y : A^{\mathfrak{e}} \wedge (B^{\mathfrak{e}})^{\perp} \vdash (\pi_2(y) \star u^{\mathfrak{e}}) : \perp,$$
$$\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp}, y : A^{\mathfrak{e}} \wedge (B^{\mathfrak{e}})^{\perp} \vdash \lambda x(\pi_2(y) \star u^{\mathfrak{e}}) : (A^{\mathfrak{e}})^{\perp}.$$
From which, we obtain
$$\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash \lambda y(\lambda x(\pi_2(y) \star u^{\mathfrak{e}}) \star \pi_1(y)) : (A^{\mathfrak{e}})^{\perp} \vee B^{\mathfrak{e}}.$$

2. Assume now
$$\frac{\Gamma \rhd t : A \mid \triangle \qquad \Gamma \mid e : B \rhd \triangle}{\Gamma \mid t.e : A \to B \rhd \triangle}.$$
Then we have
$$\frac{\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash t^{\mathfrak{e}} : A^{\mathfrak{e}} \qquad \Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash e^{\mathfrak{e}} : (B^{\mathfrak{e}})^{\perp}}{\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash \langle t^{\mathfrak{e}}, e^{\mathfrak{e}} \rangle : A^{\mathfrak{e}} \wedge (B^{\mathfrak{e}})^{\perp}}.$$

3. From
$$\frac{\Gamma \rhd t : A \mid \triangle \qquad \Gamma \mid e : A \rhd \triangle}{\lfloor t, e \rfloor : (\Gamma \rhd \triangle)},$$
we obtain
$$\frac{\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash t^{\mathfrak{e}} : A^{\mathfrak{e}} \qquad \Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash e^{\mathfrak{e}} : (A^{\mathfrak{e}})^{\perp}}{\Gamma^{\mathfrak{e}}, (\triangle^{\mathfrak{e}})^{\perp} \vdash (e^{\mathfrak{e}} \star t^{\mathfrak{e}}) : \perp}.$$

□

Our next aim is to prove that $\overline{\lambda}\mu\tilde{\mu}^*$ can be simulated by the λ^{Sym}_{Prop}-calculus. To this end we introduce a new notion of equality in the λ^{Sym}_{Prop}-calculus.

Definition 5.3 *We define an equivalence relation $\sim$ on $\mathcal{T}$.*

- $x \sim x$,
- *if $M \sim M'$, then $\lambda x M \sim \lambda x M'$ and $\sigma_i(M) \sim \sigma_i(M')$ for $i \in \{1, 2\}$,*
- *if $M \sim M'$ and $N \sim N'$, then $\langle M, N\rangle \sim \langle M', N'\rangle$ and $(M \star N) \sim (M' \star N')$ and $(M \star N) \sim (N' \star M')$.*

We say that M and N are equal up to symmetry provided $M \sim N$.

Lemma 5.4 *Let $M, M', N, N' \in \mathcal{T}$.*

1. *If $M \sim M'$ and $N \sim N'$, then $M[x := N] \sim M'[x := N']$.*
2. *If $M \sim M'$ and $M' \to N$, then there is N' for which $M \to N'$ and $N \sim N'$.*

Proof 1. By induction on $cxty(M)$. 2. By 1. □

Lemma 5.5 *Let $u, t, e \in \mathfrak{T}$. Then $(u[x := t])^{\mathfrak{e}} = u^{\mathfrak{e}}[x := t^{\mathfrak{e}}]$ and $(u[a := e])^{\mathfrak{e}} = u^{\mathfrak{e}}[a := e^{\mathfrak{e}}]$.*

Proof By induction on $cxty(u)$. □

Now we can formulate our assertion about the simulation of the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus by the λ^{Sym}_{Prop}-calculus.

Theorem 5.6 *Let $v, w \in \mathfrak{T}$.*

1. *If $v \hookrightarrow_r w$ and $r \in \{\beta\,,\, \mu\,,\, \tilde{\mu}\,,\, s_l\,,\, s_r\}$, then $v^{\mathfrak{e}} \to^+ w^{\mathfrak{e}}$.*
2. *If $v \hookrightarrow_r w$ and $r \in \{cl_{1,l}\,,\, cl_{1,r}\,,\, cl_2\}$, then $v^{\mathfrak{e}} \sim w^{\mathfrak{e}}$.*

Proof

1. Let us only treat the typical cases.
   (a) If $v = \lfloor \lambda x u, (t.e)\rfloor \hookrightarrow_\beta \lfloor t, \tilde{\mu}x\lfloor u, e\rfloor\rfloor = w$, then $v^{\mathfrak{e}} = (\langle t^{\mathfrak{e}}, e^{\mathfrak{e}}\rangle \star \lambda y(\lambda x(\pi_2(y) \star u^{\mathfrak{e}}) \star \pi_1(y))) \to_{\beta^\perp} (\lambda x(\pi_2(\langle t^{\mathfrak{e}}, e^{\mathfrak{e}}\rangle) \star u^{\mathfrak{e}}) \star \pi_1(\langle t^{\mathfrak{e}}, e^{\mathfrak{e}}\rangle)) \to^* (\lambda x(e^{\mathfrak{e}} \star u^{\mathfrak{e}}) \star t^{\mathfrak{e}}) = w^{\mathfrak{e}}$.
   (b) If $v = \lfloor \mu a p, e\rfloor \hookrightarrow_\mu p[a := e] = w$, then, by Lemma 5.5, $v^{\mathfrak{e}} = (e^{\mathfrak{e}} \star \lambda a p^{\mathfrak{e}}) \to_{\beta^\perp} p^{\mathfrak{e}}[a = e^{\mathfrak{e}}] = w^{\mathfrak{e}}$.
   (c) If $v = \mu a\lfloor w, a\rfloor \hookrightarrow_{s_l} w$, $a \notin w$, then $v^{\mathfrak{e}} = \lambda a(a \star w^{\mathfrak{e}}) \to_{\eta^\perp} w^{\mathfrak{e}}$.
2. (a) If $v = \overline{\overline{\tilde{u}}} \hookrightarrow_{cl_{1,l}} u = w$, then $v^{\mathfrak{e}} = (\overline{\overline{\tilde{u}}})^{\mathfrak{e}} = u^{\mathfrak{e}} = w^{\mathfrak{e}}$.
   (b) If $v = \lfloor \overline{v}, \tilde{u}\rfloor \hookrightarrow_{cl_2} \lfloor u, v\rfloor = w$, then $v^{\mathfrak{e}} = \lfloor \overline{v}, \tilde{u}\rfloor^{\mathfrak{e}} = (u^{\mathfrak{e}} \star v^{\mathfrak{e}}) \sim w^{\mathfrak{e}}$.

□

Corollary 5.7 *The $\overline{\lambda}\mu\tilde{\mu}^*$-calculus is strongly normalizable.*

Proof Let S be an infinite reduction sequence in the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus. By Theorem 5.6 and Lemma 5.4, S cannot contain an infinite number of β-, μ-, $\tilde{\mu}$-, s_l- and s_r-reductions. Thus, there would exist an infinite reduction sequence consisting entirely of $cl_{1,l}$-, $cl_{1,r}$- and cl_2-reductions, which is impossible. □

5.2 A translation of the λ^{Sym}_{Prop}-calculus into the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus

Now we are going to deal with the converse relation. That is we will present a translation of the λ^{Sym}_{Prop}-calculus into the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus which faithfully reflects the typability relations of one calculus in the other one. Then we prove that our translation is in fact a simulation of the λ^{Sym}_{Prop}-calculus in the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus.

Definition 5.8 *1. The translation $.^{\mathfrak{f}} : \mathcal{T} \longrightarrow \mathfrak{T}$ is defined as follows.*

$$M^{\mathfrak{f}} = \begin{cases} x & \text{if } M = x, \\ \lfloor Q^{\mathfrak{f}}, \widetilde{P^{\mathfrak{f}}} \rfloor & \text{if } M = (P \star Q), \\ \overline{\tilde{\mu} x N^{\mathfrak{f}}} & \text{if } M = \lambda x N, \\ \overline{(P^{\mathfrak{f}}.\widetilde{Q^{\mathfrak{f}}})} & \text{if } M = \langle P, Q \rangle, \\ \lambda x \mu \beta \lfloor N^{\mathfrak{f}}, \widetilde{x} \rfloor & \text{if } M = \sigma_1(N),\ x \notin Fv(N^{\mathfrak{f}}) \text{ and } \beta \notin Fv(\lfloor N^{\mathfrak{f}}, \widetilde{x} \rfloor), \\ \lambda x N^{\mathfrak{f}} & \text{if } M = \sigma_2(N) \text{ and } x \notin Fv(N^{\mathfrak{f}}). \end{cases}$$

2. The translation $.^{\mathfrak{f}}$ applies to the types as follows.

- $\alpha^{\mathfrak{f}} = \alpha$,
- $(\alpha^{\perp})^{\mathfrak{f}} = \alpha^{\perp}$,
- $(A \wedge B)^{\mathfrak{f}} = (A^{\mathfrak{f}} \to (B^{\mathfrak{f}})^{\perp})^{\perp}$,
- $(A \vee B)^{\mathfrak{f}} = (A^{\mathfrak{f}})^{\perp} \to B^{\mathfrak{f}}$.

We remark that $.^{\mathfrak{f}}$ maps the terms of the λ^{Sym}_{Prop}-calculus with type $\perp$ to c-terms of the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, which have no types. We also have, for all types A, $(A^{\perp})^{\mathfrak{f}} = (A^{\mathfrak{f}})^{\perp}$. Therefore the translation $.^{\mathfrak{f}}$ maps equal types to equal types.

Lemma 5.9 *1. If $\Gamma \vdash M : A$ and $A \neq \perp$, then $\Gamma^{\mathfrak{f}} \rhd M^{\mathfrak{f}} : A^{\mathfrak{f}}$.*

2. If $\Gamma \vdash M : \perp$, then $M^{\mathfrak{f}} : (\Gamma^{\mathfrak{f}} \rhd)$.

Proof The proof proceeds by a simultaneous induction on the length of the derivation in the λ^{Sym}_{Prop}-calculus. We can observe again that the notion of $.^{\mathfrak{f}}$ in Definition 5.8 is conceived in a way to make the statements of the lemma true. Let us only examine some of the typical cases of the first assertion.

1. Suppose
$$\frac{\Gamma, x : A \vdash u : \perp}{\Gamma \ \vdash \ \lambda x u : A^{\perp}}.$$
Then, applying the induction hypothesis,
$$\frac{\dfrac{u^{\mathfrak{f}} : (\Gamma^{\mathfrak{f}}, x : A^{\mathfrak{f}} \rhd)}{\Gamma^{\mathfrak{f}} \mid \tilde{\mu} x u^{\mathfrak{f}} : A^{\mathfrak{f}} \rhd}}{\Gamma^{\mathfrak{f}} \ \rhd \ \overline{\tilde{\mu} x u^{\mathfrak{f}}} : (A^{\mathfrak{f}})^{\perp}}.$$

2. If
$$\frac{\Gamma \vdash u : A}{\Gamma \vdash \sigma_1(u) : A \vee B},$$
then, we obtain
$$\frac{\dfrac{\dfrac{\dfrac{}{\Gamma^{\mathfrak{f}}, x : (A^{\mathfrak{f}})^{\perp} \rhd u^{\mathfrak{f}} : A^{\mathfrak{f}} \mid \beta : B^{\mathfrak{f}}} \quad \dfrac{\dfrac{\Gamma^{\mathfrak{f}}, x : (A^{\mathfrak{f}})^{\perp} \rhd x : (A^{\mathfrak{f}})^{\perp} \mid \beta : B^{\mathfrak{f}}}{\Gamma^{\mathfrak{f}}, x : (A^{\mathfrak{f}})^{\perp} \mid \widetilde{x} : A^{\mathfrak{f}} \rhd \beta : B^{\mathfrak{f}}}}{}}{\lfloor u^{\mathfrak{f}}, \widetilde{x} \rfloor : (\Gamma^{\mathfrak{f}}, x : (A^{\mathfrak{f}})^{\perp} \rhd \beta : B^{\mathfrak{f}})}}{\Gamma^{\mathfrak{f}}, x : (A^{\mathfrak{f}})^{\perp} \rhd \mu\beta \lfloor u^{\mathfrak{f}}, \widetilde{x} \rfloor : B^{\mathfrak{f}}}}{\Gamma^{\mathfrak{f}} \rhd \lambda x \mu \beta \lfloor u^{\mathfrak{f}}, \widetilde{x} \rfloor : (A^{\mathfrak{f}})^{\perp} \to B^{\mathfrak{f}}}.$$

3. From

$$\frac{\Gamma \vdash u : A^{\perp} \qquad \Gamma \vdash v : A}{\Gamma \vdash (u \star v) : \perp},$$

we obtain

$$\frac{\dfrac{\Gamma^{\mathfrak{f}} \rhd u^{\mathfrak{f}} : (A^{\mathfrak{f}})^{\perp}}{\Gamma^{\mathfrak{f}} \mid \widetilde{u^{\mathfrak{f}}} : A^{\mathfrak{f}} \rhd} \qquad \overline{\Gamma^{\mathfrak{f}} \rhd v^{\mathfrak{f}} : A^{\mathfrak{f}}}}{\lfloor v^{\mathfrak{f}}, \widetilde{u^{\mathfrak{f}}} \rfloor : (\Gamma^{\mathfrak{f}} \rhd \,)}.$$

□

Now we turn to the proof of the simulation of the λ^{Sym}_{Prop}-calculus in the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus.

Lemma 5.10 *Let $M, N \in \mathcal{T}$. Then $(M[x := N])^{\mathfrak{f}} = M^{\mathfrak{f}}[x := N^{\mathfrak{f}}]$.*

Proof By induction on $cxty(M)$. □

Theorem 5.11 *Let $M, N \in \mathcal{T}$. If $M \to N$, then $M^{\mathfrak{f}} \hookrightarrow^{+} N^{\mathfrak{f}}$.*

Proof Let us prove some of the more interesting cases.

1. If $M = (\lambda x P \star Q) \to_{\beta} P[x := Q] = N$, then, applying Lemma 5.10,

   $M^{\mathfrak{f}} = \lfloor Q^{\mathfrak{f}}, \widetilde{\overline{\tilde{\mu} x P^{\mathfrak{f}}}} \rfloor \hookrightarrow_{cl_1} \lfloor Q^{\mathfrak{f}}, \tilde{\mu} x P^{\mathfrak{f}} \rfloor \hookrightarrow_{\tilde{\mu}} P^{\mathfrak{f}}[x := Q^{\mathfrak{f}}] = N^{\mathfrak{f}}$.

2. If $M = (Q \star \lambda x P) \to_{\beta_{\perp}} P[x := Q] = N$, then

   $M^{\mathfrak{f}} = \lfloor \overline{\tilde{\mu} x P^{\mathfrak{f}}}, \widetilde{Q^{\mathfrak{f}}} \rfloor \hookrightarrow_{cl_2} \lfloor Q^{\mathfrak{f}}, \tilde{\mu} x P^{\mathfrak{f}} \rfloor \hookrightarrow_{\tilde{\mu}} P^{\mathfrak{f}}[x := Q^{\mathfrak{f}}] = N^{\mathfrak{f}}$.

3. If $M = (\langle P, Q \rangle \star \sigma_1(R)) \to_{\pi} (P \star R) = N$, then

   $M^{\mathfrak{f}} = \lfloor \lambda x \mu b \lfloor R^{\mathfrak{f}}, \tilde{x} \rfloor, \widetilde{\overline{(P^{\mathfrak{f}}.\widetilde{Q^{\mathfrak{f}}})}} \rfloor \hookrightarrow_{cl_1} \lfloor \lambda x \mu b \lfloor R^{\mathfrak{f}}, \tilde{x} \rfloor, (P^{\mathfrak{f}}.\widetilde{Q^{\mathfrak{f}}},) \rfloor \hookrightarrow_{\lambda}$
   $\lfloor P^{\mathfrak{f}}, \tilde{\mu} x \lfloor \mu b \lfloor R^{\mathfrak{f}}, \tilde{x} \rfloor, \widetilde{Q^{\mathfrak{f}}} \rfloor \rfloor \hookrightarrow_{\tilde{\mu}} \lfloor \mu b \lfloor R^{\mathfrak{f}}, \widetilde{P^{\mathfrak{f}}} \rfloor, \widetilde{Q^{\mathfrak{f}}} \rfloor \hookrightarrow_{\mu} \lfloor R^{\mathfrak{f}}, \widetilde{P^{\mathfrak{f}}} \rfloor = N^{\mathfrak{f}}$.

□

We could as well demonstrate that the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus is strongly normalizable by applying the method presented in Section 3 (cf. [2]). The following result states that in this case the strong normalizability of the λ^{Sym}_{Prop}-calculus would arise as a direct consequence of that of the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus.

Corollary 5.12 *If the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus is strongly normalizable, then the same is true for the λ^{Sym}_{Prop}-calculus as well.*

Proof By Theorem 5.11. □

5.3 The connection between the two translations

In this subsection we examine the connection between the two transformations. We prove that both compositions $.^{\mathfrak{e}^{\mathfrak{f}}} : \mathfrak{T} \longrightarrow \mathfrak{T}$ and $.^{\mathfrak{f}^{\mathfrak{e}}} : \mathcal{T} \longrightarrow \mathcal{T}$ are such that we can get back the original terms by performing some steps of reduction on $u^{\mathfrak{e}^{\mathfrak{f}}}$ or on $M^{\mathfrak{f}^{\mathfrak{e}}}$, respectively. That is, the following theorems are valid. The case of $.^{\mathfrak{f}^{\mathfrak{e}}}$ is the easier one.

First we describe the effect of $.^{\mathfrak{f}^{\mathfrak{e}}}$ on the typing relations.

Lemma 5.13 *If $\Gamma \vdash M : A$, then $\Gamma \vdash M^{\mathfrak{f}^{\mathfrak{e}}} : A$.*

Proof Combining Lemmas 5.9 and 5.2. □

Theorem 5.14 *Let $M \in \mathcal{T}$. Then $M^{\mathfrak{f}^{\mathfrak{e}}} \to^* M$.*

Proof By induction on $cxty(M)$. We consider only the more interesting cases.

1. If $M = (P \star Q)$, then $(P \star Q)^{\mathfrak{f}^{\mathfrak{e}}} = \lfloor Q^{\mathfrak{f}}, \widetilde{P^{\mathfrak{f}}} \rfloor^{\mathfrak{e}} = (P^{\mathfrak{f}^{\mathfrak{e}}} \star Q^{\mathfrak{f}^{\mathfrak{e}}}) \to^* (P \star Q)$.

2. If $M = \langle P, Q \rangle$, then $\langle P, Q \rangle^{\mathfrak{f}^{\mathfrak{e}}} = \overline{(P^{\mathfrak{f}}.\widetilde{Q^{\mathfrak{f}}})}^{\mathfrak{e}} = \langle P^{\mathfrak{f}^{\mathfrak{e}}}, Q^{\mathfrak{f}^{\mathfrak{e}}} \rangle \to^* \langle P, Q \rangle$.

3. If $M = \sigma_1(N)$, then
$\sigma_1(N)^{\mathfrak{f}^{\mathfrak{e}}} = \lambda x \mu \beta \lfloor N^{\mathfrak{f}}, \widetilde{x} \rfloor^{\mathfrak{e}} = \lambda y(\lambda x(\pi_2(y) \star (\mu \beta \lfloor N^{\mathfrak{f}}, \widetilde{x} \rfloor)^{\mathfrak{e}}) \star \pi_1(y)) =$
$\lambda y(\lambda x(\pi_2(y) \star \lambda \overline{\beta}(x \star N^{\mathfrak{f}^{\mathfrak{e}}})) \star \pi_1(y)) \to_{\beta^\perp} \lambda y(\lambda x(x \star N^{\mathfrak{f}^{\mathfrak{e}}}) \star \pi_1(y)) \to_{\eta^\perp}$
$\lambda y(N^{\mathfrak{f}^{\mathfrak{e}}} \star \pi_1(y)) \to_\beta \lambda y(y \star \sigma_1(N^{\mathfrak{f}^{\mathfrak{e}}})) \to_{\eta^\perp} \sigma_1(N^{\mathfrak{f}^{\mathfrak{e}}}) \to^* \sigma_1(N)$.

□

We begin to examine the composition $.^{\mathfrak{e}\mathfrak{f}} : \mathfrak{T} \to \mathfrak{T}$ for an arbitrary u. First we make the following observation.

Lemma 5.15 *1. If $\Gamma \rhd t : A \mid \triangle$, then $\Gamma, \triangle^\perp \rhd t^{\mathfrak{e}\mathfrak{f}} : A$.*

2. If $\Gamma \mid e : A \rhd \triangle$, then $\Gamma, \triangle^\perp \rhd e^{\mathfrak{e}\mathfrak{f}} : A^\perp$.

3. If $p : (\Gamma \rhd \triangle)$, then $p^{\mathfrak{e}\mathfrak{f}} : (\Gamma, \triangle^\perp \rhd)$.

Proof Combining Lemmas 5.2 and 5.9. □

Theorem 5.14 states that, if M is an λ^{Sym}_{Prop}-term, then M can be related to $M^{\mathfrak{f}^{\mathfrak{e}}}$ by the reductions in the λ^{Sym}_{Prop}-calculus. We note that we are not able to obtain u from $u^{\mathfrak{e}\mathfrak{f}}$ in such a way. We can find a term T instead such that $u^{\mathfrak{e}\mathfrak{f}} \hookrightarrow^* T(u)$. The fonction T can intuitively be considered as the description how λ^{Sym}_{Prop}-connectives can be embedded into the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus. It turns out that the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus translates the λ^{Sym}_{Prop}-terms not so smoothly as it was the case with the other direction.

Definition 5.16 *We define a function T assigning a $\overline{\lambda}\mu\tilde{\mu}^*$-term to a $\overline{\lambda}\mu\tilde{\mu}^*$-term.*

- $T(x) = x$,
- $T(\lambda x u) = \overline{\tilde{\mu} y \lfloor T(u)[x := p_1(y)], \widetilde{p_2(y)} \rfloor}$,
- $T(\mu \alpha p) = \overline{\tilde{\mu} \overline{\alpha} T(p)}$,
- $T(\overline{u}) = T(u)$,
- $T(\alpha) = \overline{\alpha}$,
- $T((u.v)) = \langle T(u), T(v) \rangle$,
- $T(\tilde{\mu} x p) = \overline{\tilde{\mu} x T(p)}$,
- $T(\widetilde{h}) = T(h)$,
- $T(\lfloor t, e \rfloor) = \lfloor T(t), \widetilde{T(e)} \rfloor$.

Theorem 5.17 *Let $u \in \mathfrak{T}$. We have $u^{\mathfrak{e}\mathfrak{f}} \hookrightarrow^* T(u)$.*

Proof By induction on $cxty(u)$. We consider only some of the cases.

1. If $u = \lambda x v$, then

$$u^{\mathfrak{e}\mathfrak{f}} = (\lambda y(\lambda x(\pi_2(y) \star u^{\mathfrak{e}}) \star \pi_1(y)))^{\mathfrak{f}} = \overline{\tilde{\mu} y \lfloor p_1(y), \widetilde{\overline{\tilde{\mu} x \lfloor v^{\mathfrak{e}\mathfrak{f}}, \widetilde{p_2(y)} \rfloor}} \rfloor} \hookrightarrow_{cl_{1,r}}$$
$$\overline{\tilde{\mu} y \lfloor p_1(y), \tilde{\mu} x \lfloor v^{\mathfrak{e}\mathfrak{f}}, \widetilde{p_2(y)} \rfloor \rfloor} \hookrightarrow_{\tilde{\mu}} \overline{\tilde{\mu} y \lfloor v^{\mathfrak{e}\mathfrak{f}}[x := p_1(y)], \widetilde{p_2(y)}[x := p_1(y)] \rfloor} \hookrightarrow^* T(u).$$

2. If $u = \tilde{\mu} x \lfloor t, v \rfloor$, then

$$u^{\mathfrak{e}\mathfrak{f}} = (\lambda x(v^{\mathfrak{e}} \star t^{\mathfrak{e}}))^{\mathfrak{f}} = \overline{\tilde{\mu} x \lfloor t^{\mathfrak{e}\mathfrak{f}}, \widetilde{v^{\mathfrak{e}\mathfrak{f}}} \rfloor} \hookrightarrow^* \overline{\tilde{\mu} x \lfloor T(t), \widetilde{T(v)} \rfloor} = \overline{\tilde{\mu} x T(\lfloor t, v \rfloor)} = T(u).$$

□

Remark 5.18 *We remark that we cannot expect $T(u)$ to be expressible with the help of $\mathfrak{T}$. Namely, we can show that, if $=_{\overline{\lambda}\mu\tilde{\mu}^*}$ denotes the reflexive, transitive closure of the compatible union of the reduction relations in the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus, then none of the assertions below are valid.*

1. *There exists a a $\overline{\lambda}\mu\tilde{\mu}^*$-term Φ such that, for every c-term c, $T(c) =_{\overline{\lambda}\mu\tilde{\mu}^*} \Phi(c)$.*
2. *There exists a a $\overline{\lambda}\mu\tilde{\mu}^*$-term Φ_1 such that, for every l-term t, $T(t) =_{\overline{\lambda}\mu\tilde{\mu}^*} \Phi_1(t)$.*
3. *There exists a a $\overline{\lambda}\mu\tilde{\mu}^*$-term Φ_2 such that, for every r-term e, $T(e) =_{\overline{\lambda}\mu\tilde{\mu}^*} \Phi_2(e)$.*

6 Conclusion

The paper is mainly devoted to an arithmetical proof of the strong normalization of the λ^{Sym}_{Prop}-calculus introduced by Berardi and Barbanera (cf. [1]). The proof is an adaptation of the work in [4]. The novelty of our paper is the application of the method of zoom-in sequences of redexes: we achieve the main theorem by identifying the minimal non-strongly normalizing redexes of an infinite reduction sequence, which we call a zoom-in sequence of redexes. The idea of zoom-in sequences was inspired by the notion of perpetual reduction strategies in [19]. Following the proof of the strong normalization of the λ^{Sym}_{Prop}-calculus, the $\overline{\lambda}\mu\tilde{\mu}$-calculus is introduced, which was defined in [3]. The same proof of strong normalization as we have presented for the λ^{Sym}_{Prop}-calculus would also work for the calculus of Curien and Herbelin (cf. [2]). However, instead of adapting the proof method for the $\overline{\lambda}\mu\tilde{\mu}$-calculus, we designed a translation of the λ^{Sym}_{Prop}-calculus in the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus and vice versa, where the $\overline{\lambda}\mu\tilde{\mu}^*$-calculus is the $\overline{\lambda}\mu\tilde{\mu}$-calculus augmented with terms explicitly expressing negation and with rules handling them. The translation allows us to assert strong normalization for the $\overline{\lambda}\mu\tilde{\mu}^*$- and, hence, for the $\overline{\lambda}\mu\tilde{\mu}$-calculus.

On the technical side, we remark that there were two main difficulties that rendered the proof a little more involved. First, we had to work with an alternating substitution defined inductively starting from two sets of terms. The reason was that we had to prove a more general statement to locate the supposedly non strongly normalizing part of a term emerging as a result of a substitution. Simple substitutions would not have been enough for our purpose. The second difficulty was that in order to establish a key property of zoom-in sequences in Lemma 3.9 we had to move forward the Hypothesis "H" from the main theorem, thus making the application of the hypothesis implicit in the sequel. We think that the elimination of both problems would considerably enhance the paper's intelligibility.

It seems promising to investigate whether the present method of verifying strong normalization can be applied to systems other than simple typed logical calculi, for example, proof nets (see [7]). Another fields of interest could be intuitionistic and classical typed systems with explicit substitutions (cf. [18]). To handle these systems, the present proof must be simplified, we have to pay attention in our proof, for example, that the substitutions are defined by two sets of terms of different types. Finally, we remark that it is a natural requirement of a proof formalizable in first order arithmetic to enable us to find an upper bound for the lengths of the reduction sequences. At its present form, our proof does not make it possible, this raises another demand for the simplification of the results.

Acknowledgements. We wish to thank René David and the anonymous referees for helpful discussions and remarks.

References

[1] F. Barbanera and S. Berardi. *A symmetric lambda calculus for classical program extraction.* Information and computation (125), pp. 103-117, 1996.

[2] P. Battyányi. *Normalization properties of symmetric logical calculi.* PhD thesis, University of Chambéry, 2007.

[3] P.-L. Curien and H. Herbelin. *The duality of computation.* ACM SIGPLAN Notices, 2000.

[4] R. David and K. Nour. *Arithmetical proofs of strong normalization results for the symmetric $\lambda\mu\mu'$-calculus.* TLCA 2005, Lecture Notes in Computer Science (3461), pp. 162-178, Springer Verlag, Berlin, 2005.

[5] R. David and K. Nour. *Arithmetical proofs of strong normalization results for symmetric lambda calculi.* Fundamenta Informaticae (77), pp. 1001-1022, 2007.

[6] T. Griffin. *A formulae-as-type notion of control*, POPL 1990, pp. 47-58, ACM Press, New York, 1990.

[7] O. Laurent. *An introduction to proof nets*, Course Notes, Ecole normale supérieure de Lyon, 2013.

[8] C. R. Murthy. *An evaluation semantics for classical proofs.* Proceedings of the sixth annual IEEE symposium, pp. 96-107, 1991.

[9] P. de Groote. *Strong normalization of classical natural deduction with disjunction.* TLCA 2001, Lecture Notes in Computer Science (2044), pp. 182-196, Springer Verlag, Berlin, 2001.

[10] H. Herbelin. *Séquents qu'on calcule.* PhD thesis, University of Paris 7, 1995.

[11] K. Nour. *La valeur d'un entier classique en $\lambda\mu$-calcul.* Archive for Mathematical Logic (36), pp. 461-471, 1997.

[12] K. Nour. *Classical combinatory logic.* Computational Logic and Application, DMTCS proc. AF, pp. 87-96, 2006.

[13] M. Parigot. *$\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction.* Lecture Notes in Computer Science (624), pp. 190-201, Springer Verlag, Berlin, 1992.

[14] M. Parigot. *Classical proofs as programs.* Lecture Notes in Computer Science (713), pp. 263-276, Springer Verlag, Berlin, 1993.

[15] M. Parigot. *Free Deduction: An Analysis of "Computations" in Classical Logic.* Lecture Notes in Computer Science (592), pp. 361-380, Springer-Verlag, Berlin, 1990.

[16] E. Polonovski. *Substitutions explicites, logique et normalisation.* PhD thesis, Paris 7, 2004.

[17] J. Rocheteau. *$\lambda\mu$-calculus and duality: call-by-value and call-by-name.* TLCA 2005, Lecture Notes in Computer Science (3461), pp. 204-218, Springer Verlag, Berlin, 2005.

[18] K. Rose. *Explicit subsitution.* Technical report, University of Aarhus, 1996.

[19] F. van Raamsdonk and P. Severi and M. H. Sørensen and H. Xi. *Perpetual Reductions in Lambda-Calculus.* Journal of Information and Computation (149), pp. 173-225, 1999.

[20] W. W. Tait. *Intensional Interpretations of Functionals of Finite Type I.* Journal of Symbolic Logic (32), pp. 198-212, 1967.