



## Strong normalization results by translation

René David, Karim Nour

► **To cite this version:**

| René David, Karim Nour. Strong normalization results by translation. 2009. hal-00385206

**HAL Id: hal-00385206**

**<https://hal.archives-ouvertes.fr/hal-00385206>**

Submitted on 18 May 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Strong normalization results by translation

René DAVID and Karim NOUR

LAMA - Equipe LIMD  
Université de Chambéry  
73376 Le Bourget du Lac  
e-mail: {david,nour}@univ-savoie.fr

## Abstract

We prove the strong normalization of full classical natural deduction (i.e. with conjunction, disjunction and permutative conversions) by using a translation into the simply typed  $\lambda\mu$ -calculus. We also extend Mendler's result on recursive equations to this system.

## 1 Introduction

It is well known that, when the underlying logic is the classical one (i.e. the absurdity rule is allowed) the connectives  $\vee$  and  $\wedge$  are redundant (they can be coded by using  $\rightarrow$  and  $\perp$ ). From a logical point of view, considering the full logic is thus somehow useless. However, from the computer science point of view, considering the full logic is interesting because, by the so-called Curry-Howard correspondence, formulas can be seen as types for functional programming languages and correct programs can be extracted from proofs. The connectives  $\wedge$  and  $\vee$  have a functional counter-part ( $\wedge$  corresponds to a product and  $\vee$  to a co-product, i.e. a *case of*) and it is thus useful to have them as primitive.

In this paper, we study the typed  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus. This calculus, introduced by de Groote in [7], is an extension of Parigot's  $\lambda\mu$ -calculus. It is the computational counterpart of classical natural deduction with  $\rightarrow$ ,  $\wedge$  and  $\vee$ . Three notions of conversions are necessary in order to have the sub-formula property : logical, classical and permutative conversions.

The proofs of the strong normalization of the cut-elimination procedure for the full classical logic are quite recent and three kinds of proofs are given in the literature.

*Proofs by CPS-translation.* In [7] de Groote also gave a proof of the strong normalization of the typed  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus using a CPS-translation into the simply typed  $\lambda$ -calculus i.e. the implicative intuitionistic logic but his proof contains an error as Matthes pointed out in [8]. Nakazawa and Tatsuta corrected de Groote's proof in [12] by using the notion of augmentations.

*Syntactical proofs.* We gave in [4] a direct and syntactical proof of strong normalization. The proof is based on a substitution lemma which stipulates that replacing in a strongly normalizable deduction an hypothesis by another strongly normalizable deduction gives a strongly normalizable deduction. The proof uses a technical lemma concerning commutative reductions. But, though the idea of the proof of this lemma (as given in [4]) works, it is not complete and (as pointed out by Matthes in a private communication) it also contains some errors.

*Semantical proofs.* K. Saber and the second author gave in [13] a semantical proof of this result by using the notion of saturated sets. This proof is a generalization of Parigot's strong normalization result of the  $\lambda\mu$ -calculus with the types

of Girard's system  $\mathcal{F}$  by using reducibility candidates. This proof uses the technical lemma of [4] concerning commutative reductions. In [9] and [17], R. Matthes and Tastuta give another semantical proofs by using a (more complex) concept of saturated sets.

This paper presents a new proof of the strong normalization of the simply typed  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus. This proof is formalizable in Peano first order arithmetic and does not need any complex lemma. It is obtained by giving a translation of this calculus into the  $\lambda\mu$ -calculus. The coding of  $\wedge$  and  $\vee$  in classical logic is the usual one but, as far as we know, the fact that this coding behaves correctly with the computation, via the Curry-Howard correspondence, has never been analyzed. This proof is much simpler than the existing ones<sup>1</sup>.

It also presents a new result. Mendler [11] has shown that strong normalization is preserved if, on types, we allow some equations satisfying natural (and necessary) conditions. Mendler's result concerned the implicative fragment of intuitionistic logic. By using the previous translation, we extend here this result to full classical logic .

The paper is organized as follows. Section 2 gives the various systems for which we prove the strong normalization. Section 6 gives the translation of the  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus into the  $\lambda\mu$ -calculus and section 7 extends Mendler's theorem to the  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus. For a first reading, sections 3, 4 and 5 may be skipped. They have been added to have complete proofs of the other results. Section 3 contains the proof, by the first author, of the the strong normalization of the simply typed  $\lambda$ -calculus. Section 4 gives a translation of the  $\lambda\mu$ -calculus into the  $\lambda$ -calculus and section 5 gives some well known properties of the  $\lambda\mu$ -calculus. Finally, the appendix gives a detailed proof of a lemma that needs a long but easy case analysis.

## 2 The systems

**Definition 2.1** *Let  $\mathcal{V}$  and  $\mathcal{W}$  be disjoint sets of variables.*

1. *The set of  $\lambda$ -terms is defined by the following grammar*

$$\mathcal{M} ::= \mathcal{V} \mid \lambda\mathcal{V}.\mathcal{M} \mid (\mathcal{M} \mathcal{M})$$

2. *The set of  $\lambda\mu$ -terms is defined by the following grammar*

$$\mathcal{M}' ::= \mathcal{V} \mid \lambda\mathcal{V}.\mathcal{M}' \mid (\mathcal{M}' \mathcal{M}') \mid \mu\mathcal{W}.\mathcal{M}' \mid (\mathcal{W} \mathcal{M}')$$

3. *The set of  $\lambda\mu^{\rightarrow\wedge\vee}$ -terms is defined by the following grammar*

$$\mathcal{M}'' ::= \mathcal{V} \mid \lambda\mathcal{V}.\mathcal{M}'' \mid (\mathcal{M}'' \mathcal{E}) \mid \langle \mathcal{M}'', \mathcal{M}'' \rangle \mid \omega_1\mathcal{M}'' \mid \omega_2\mathcal{M}'' \mid \mu\mathcal{W}.\mathcal{M}'' \mid (\mathcal{W} \mathcal{M}'')$$

$$\mathcal{E} ::= \mathcal{M}'' \mid \pi_1 \mid \pi_2 \mid [\mathcal{V}.\mathcal{M}'', \mathcal{V}.\mathcal{M}'']$$

Note that, for the  $\lambda\mu$ -calculus, we have adopted here the so-called de Groot calculus which is the extension of Parigot's calculus where the distinction between named and un-named terms is forgotten. In this calculus,  $\mu\alpha$  is not necessarily followed by  $[\beta]$ . We also write  $(\alpha M)$  instead of  $[\alpha]M$ .

---

<sup>1</sup>Recently, we have been aware of a paper by Wojdyga [18] who uses the same kind of translations but where all the atomic types are collapsed to  $\perp$ . Our translation allows us to extend trivially Mendler's result whereas the one of Wojdyga, of course, does not.

**Definition 2.2** 1. The reduction rule for the  $\lambda$ -calculus is the  $\beta$ -rule.

$$(\lambda x.M N) \triangleright_{\beta} M[x := N]$$

2. The reduction rules for the  $\lambda\mu$ -calculus are the  $\beta$ -rule and the  $\mu$ -rule

$$(\mu\alpha.M N) \triangleright_{\mu} \mu\alpha.M[(\alpha L) := (\alpha (L N))]$$

3. The reduction rules for the  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus are those of the  $\lambda\mu$ -calculus together with the following rules

$$\begin{aligned} & ((M_1, M_2) \pi_i) \triangleright M_i \\ & (\omega_i M [x_1.N_1, x_2.N_2]) \triangleright N_i[x_i := M] \\ & (M [x_1.N_1, x_2.N_2] \varepsilon) \triangleright (M [x_1.(N_1 \varepsilon), x_2.(N_2 \varepsilon)]) \\ & (\mu\alpha.M \varepsilon) \triangleright \mu\alpha.M[(\alpha N) := (\alpha (N \varepsilon))] \end{aligned}$$

**Definition 2.3** Let  $\mathcal{A}$  be a set of atomic constants.

1. The set  $\mathcal{T}$  of types is defined by the following grammar

$$\mathcal{T} ::= \mathcal{A} \cup \{\perp\} \mid \mathcal{T} \rightarrow \mathcal{T}$$

2. The set  $\mathcal{T}'$  of types is defined by the following grammar

$$\mathcal{T}' ::= \mathcal{A} \cup \{\perp\} \mid \mathcal{T}' \rightarrow \mathcal{T}' \mid \mathcal{T}' \wedge \mathcal{T}' \mid \mathcal{T}' \vee \mathcal{T}'$$

As usual  $\neg A$  is an abbreviation for  $A \rightarrow \perp$ .

**Definition 2.4** 1. A  $\lambda$ -context is a set of declarations of the form  $x : A$  where  $x \in \mathcal{V}$ ,  $A \in \mathcal{T}$  and where a variable may occur at most once.

2. A  $\lambda\mu$ -context is a set of declarations of the form  $x : A$  or  $\alpha : \neg B$  where  $x \in \mathcal{V}$ ,  $\alpha \in \mathcal{W}$ ,  $A, B \in \mathcal{T}$  and where a variable may occur at most once.

3. A  $\lambda\mu^{\rightarrow\wedge\vee}$ -context is a set of declarations of the form  $x : A$  or  $\alpha : \neg B$  where  $x \in \mathcal{V}$ ,  $\alpha \in \mathcal{W}$ ,  $A, B \in \mathcal{T}'$  and where a variable may occur at most once.

**Definition 2.5** 1. The simply typed  $\lambda$ -calculus (denoted  $\mathcal{S}$ ) is defined by the following typing rules where  $\Gamma$  is a  $\lambda$ -context,

$$\begin{aligned} & \frac{}{\Gamma, x : A \vdash x : A} \text{ax} & \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \rightarrow B} \rightarrow_i \\ & & \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash (M N) : B} \rightarrow_e \end{aligned}$$

2. The simply typed  $\lambda\mu$ -calculus (denoted  $\mathcal{S}^{\mu}$ ) is obtained by adding to the previous rules (where  $\Gamma$  now is a  $\lambda\mu$ -context) the following rules.

$$\frac{\Gamma, \alpha : \neg A \vdash M : A}{\Gamma, \alpha : \neg A \vdash (\alpha M) : \perp} \perp_i \quad \frac{\Gamma, \alpha : \neg A \vdash M : \perp}{\Gamma \vdash \mu\alpha.M : A} \perp_e$$

3. The simply typed  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus (denoted  $\mathcal{S}^{\rightarrow\wedge\vee}$ ) is defined by adding to the previous rules (where  $\Gamma$  now is a  $\lambda\mu^{\rightarrow\wedge\vee}$ -context) the following rules.

$$\begin{array}{c}
\frac{\Gamma \vdash M : A_1 \quad \Gamma \vdash N : A_2}{\Gamma \vdash \langle M, N \rangle : A_1 \wedge A_2} \wedge_i \quad \frac{\Gamma \vdash M : A_1 \wedge A_2}{\Gamma \vdash (M \pi_i) : A_i} \wedge_e \\
\\
\frac{\Gamma \vdash M : A_j}{\Gamma \vdash \omega_j M : A_1 \vee A_2} \vee_i \\
\\
\frac{\Gamma \vdash M : A_1 \vee A_2 \quad \Gamma, x_1 : A_1 \vdash N_1 : C \quad \Gamma, x_2 : A_2 \vdash N_2 : C}{\Gamma \vdash (M [x_1.N_1, x_2.N_2]) : C} \vee_e
\end{array}$$

4. If  $\approx$  is a congruence on  $\mathcal{T}$  (resp.  $\mathcal{T}'$ ), we define the systems  $\mathcal{S}_{\approx}$ , (resp.  $\mathcal{S}_{\approx}^{\mu}$ ,  $\mathcal{S}_{\approx}^{\rightarrow \wedge \vee}$ ) as the system  $\mathcal{S}$  (resp.  $\mathcal{S}^{\mu}$ ,  $\mathcal{S}^{\rightarrow \wedge \vee}$ ) where we have added the following typing rule.

$$\frac{\Gamma \vdash M : A \quad A \approx B}{\Gamma \vdash M : B} \approx$$

**Notation 2.1** • We will denote by  $\text{size}(M)$  the complexity of the term  $M$ .

- Let  $\vec{P}$  be a finite (possibly empty) sequence of terms and  $M$  be a term. We denote by  $(M \vec{P})$  the term  $(M P_1 \dots P_n)$  where  $\vec{P} = P_1, \dots, P_n$ .
- In the rest of the paper  $\triangleright$  will represent the reduction determined by all the rules of the corresponding calculus.
- If we want to consider only some of the rules we will mention them as a subscript of  $\triangleright$ . For example, in the  $\lambda\mu^{\rightarrow \wedge \vee}$ -calculus,  $M \triangleright_{\beta\mu} N$  means that  $M$  reduces to  $N$  either by the  $\beta$ -rule or by the  $\mu$ -rule.
- As usual,  $\triangleright_r^*$  (resp.  $\triangleright_r^+$ ) denotes the symmetric and transitive closure of  $\triangleright_r$  (resp. the transitive closure of  $\triangleright_r$ ). We denote  $M \triangleright_r^+ N$  iff  $M = N$  or  $M \triangleright_r N$ .
- A term  $M$  is strongly normalizable for a reduction  $\triangleright_r$  (denoted as  $M \in SN_r$ ) if there is no infinite sequence of reductions  $\triangleright_r$  starting from  $M$ . For  $M \in SN_r$ , we denote by  $\eta_r(M)$  the length of the longest reduction of  $M$ .
- If  $M \triangleright_r^* N$ , we denote by  $\text{lg}(M \triangleright_r^* N)$  the number of steps in the reduction  $M \triangleright_r^* N$ . If  $M \triangleright^* N$ , we denote by  $\text{lg}_r(M \triangleright^* N)$  the number of  $\triangleright_r$  steps of the reduction in  $M \triangleright^* N$ .

### 3 Strong normalization of $\mathcal{S}$

This section gives a simple proof (due to the first author) of the strong normalization of the simply typed  $\lambda$ -calculus.

**Lemma 3.1** Let  $M, N, \vec{O} \in \mathcal{M}$ . If  $M, N, \vec{O} \in SN_{\beta}$  and  $(M N \vec{O}) \notin SN_{\beta}$ , then  $(M_1[x := N] \vec{O}) \notin SN_{\beta}$  for some  $M_1$  such that  $M \triangleright_{\beta}^* \lambda x.M_1$ .

**Proof** Since  $M, N, \vec{O} \in SN_{\beta}$ , the infinite reduction of  $T = (M N \vec{O})$  looks like:  $T \triangleright_{\beta}^* (\lambda x.M_1 N_1 \vec{O}_1) \triangleright_{\beta} (M_1[x := N_1] \vec{O}_1) \triangleright_{\beta}^* \dots$ . The result immediately follows from the fact that  $(M_1[x := N] \vec{O}) \triangleright_{\beta}^* (M_1[x := N_1] \vec{O}_1)$ .  $\square$

**Lemma 3.2** *If  $M, N \in SN_\beta$  are typed  $\lambda$ -terms, then  $M[x := N] \in SN_\beta$ .*

**Proof** By induction on  $(type(N), \eta_\beta(M), size(M))$  where  $type(N)$  is the complexity of the type of  $N$ . The cases  $M = \lambda x.M_1$  and  $M = (y \vec{O})$  for  $y \neq x$  are trivial.

- $M = (\lambda y.P Q \vec{O})$ . By the induction hypothesis,  $P[x := N], Q[x := N]$  and  $\vec{O}[x := N]$  are in  $SN_\beta$ . By lemma 3.1 it is enough to show that  $(P[x := N][y := Q[x := N]] \vec{O}[x := N]) = M'[x := N] \in SN_\beta$  where  $M' = (P[y := Q] \vec{O})$ . But  $\eta_\beta(M') < \eta_\beta(M)$  and the result follows from the induction hypothesis.
- $M = (x P \vec{O})$ . By the induction hypothesis,  $P_1 = P[x := N]$  and  $\vec{O}_1 = \vec{O}[x := N]$  are in  $SN_\beta$ . By lemma 3.1 it is enough to show that if  $N \triangleright_\beta^* \lambda y.N_1$  then  $M_1 = (N_1[y := P_1] \vec{O}_1) \in SN_\beta$ . By the induction hypothesis (since  $type(P_1) < type(N)$ )  $N_1[y := P_1] \in SN_\beta$  and thus, by the induction hypothesis (since  $M_1 = (z \vec{O}_1) [z := N_1[y := P_1]]$  and  $type(N_1) < type(N)$ )  $M_1 \in SN_\beta$ . □

**Theorem 3.1** *The simply typed  $\lambda$ -calculus is strongly normalizing.*

**Proof** By induction on  $M$ . The cases  $M = x$  or  $M = \lambda x.P$  are trivial. If  $M = (N P) = (z P)[z := N]$  this follows from lemma 3.2 and the induction hypothesis. □

## 4 A translation of the $\lambda\mu$ -calculus into the $\lambda$ -calculus

We give here a translation of the simply typed  $\lambda\mu$ -calculus into the simply typed  $\lambda$ -calculus. This translation is a simplified version of Parigot's translation in [15]. His translation uses both a translation of types (by replacing each atomic formula  $A$  by  $\neg\neg A$ ) and a translation of terms. But it is known that, in the implicative fragment of propositional logic, it is enough to add  $\neg\neg$  in front of the rightmost variable. The translation we have chosen consists in decomposing the formulas (by using the terms  $T_A$ ) until the rightmost variable is found and then using the constants  $c_X$  of type  $\neg\neg X \rightarrow X$ . With such a translation the type does not change.

Since the translation of a term of the form  $\mu\alpha.M$  uses the type of  $\alpha$ , a formal presentation of this translation would need the use of  $\lambda$ -calculus and  $\lambda\mu$ -calculus à la Church. For simplicity of notations we have kept a presentation à la Curry, mentioning the types only when it is necessary.

We extend the system  $\mathcal{S}$  by adding, for each propositional variable  $X$ , a constant  $c_X$ . When the constants that occur in a term  $M$  are  $c_{X_1}, \dots, c_{X_n}$ , the notation  $\Gamma \vdash_{\mathcal{S}^c} M : A$  will mean  $\Gamma, c_{X_1} : \neg\neg X_1 \rightarrow X_1, \dots, c_{X_n} : \neg\neg X_n \rightarrow X_n \vdash_{\mathcal{S}} M : A$ .

**Definition 4.1** *For every  $A \in \mathcal{T}$ , we define a  $\lambda$ -term  $T_A$  as follows:*

- $T_\perp = \lambda x.(x \lambda y.y)$
- $T_X = c_X$
- $T_{A \rightarrow B} = \lambda x.\lambda y.(T_B \lambda u.(x \lambda v.(u (v y))))$

**Lemma 4.1** *For every  $A \in \mathcal{T}$ ,  $\vdash_{\mathcal{S}^c} T_A : \neg\neg A \rightarrow A$ .*

**Proof** By induction on  $A$ . □

**Definition 4.2** 1. We associate to each  $\mu$ -variable  $\alpha$  of type  $\neg A$  a  $\lambda$ -variable  $x_\alpha$  of type  $\neg A$ .

2. A typed  $\lambda\mu$ -term  $M$  is translated into an  $\lambda$ -term  $M^\diamond$  as follows:

- $\{x\}^\diamond = x$
- $\{\lambda x.M\}^\diamond = \lambda x.M^\diamond$
- $\{(M N)\}^\diamond = (M^\diamond N^\diamond)$
- $\{\mu\alpha.M\}^\diamond = (T_A \lambda x_\alpha.M^\diamond)$  if the type of  $\alpha$  is  $\neg A$
- $\{(\alpha M)\}^\diamond = (x_\alpha M^\diamond)$

**Lemma 4.2** 1.  $M^\diamond[x := N^\diamond] = \{M[x := N]\}^\diamond$ .

2.  $M^\diamond[x_\alpha := \lambda v.(x_\alpha (v N^\diamond))] \triangleright_\beta^* \{M[(\alpha L) := (\alpha (L N))]\}^\diamond$ .

**Proof** By induction on  $M$ . The first point is immediate. For the second, the only interesting case is  $M = (\alpha K)$ . Then,  $M^\diamond[x_\alpha := \lambda v.(x_\alpha (v N^\diamond))] = (\lambda v.(x_\alpha (v N^\diamond)) K^\diamond[x_\alpha := \lambda v.(x_\alpha (v N^\diamond))]) \triangleright_\beta (x_\alpha (K^\diamond[x_\alpha := \lambda v.(x_\alpha (v N^\diamond))]) N^\diamond) \triangleright_\beta^* (x_\alpha (\{K[(\alpha L) := (\alpha (L N))]\}^\diamond N^\diamond)) = \{M[(\alpha L) := (\alpha (L N))]\}^\diamond$ .  $\square$

**Lemma 4.3** Let  $M \in \mathcal{M}'$ .

1. If  $M \triangleright_\beta N$ , then  $M^\diamond \triangleright_\beta^+ N^\diamond$ .
2. If  $M \triangleright_\mu N$ , then  $M^\diamond \triangleright_\beta^+ N^\diamond$ .
3. If  $M \triangleright_{\beta\mu}^* N$ , then  $M^\diamond \triangleright_\beta^* N^\diamond$  and  $lg(M^\diamond \triangleright_\beta^* N^\diamond) \geq lg(M \triangleright_{\beta\mu}^* N)$ .

**Proof** By induction on  $M$ . (1) is immediate. (2) is as follows.

$(\mu\alpha^{-(A \rightarrow B)}.M N) \triangleright_\mu \mu\alpha^{-(A \rightarrow B)}.M[(\alpha^{-(A \rightarrow B)} L) := (\alpha^{-(A \rightarrow B)} (L N))]$  is translated by  $\{(\mu\alpha.M N)\}^\diamond = (T_{A \rightarrow B} \lambda x_\alpha.M^\diamond N^\diamond) \triangleright_\beta^+ (T_B \lambda u.M^\diamond[x_\alpha := \lambda v.(u (v N^\diamond))]) = (T_B \lambda x_\alpha.M^\diamond[x_\alpha := \lambda v.(x_\alpha (v N^\diamond))]) \triangleright_\beta^* (T_B \lambda x_\alpha.\{M[(\alpha L) := (\alpha (L N))]\}^\diamond) = \{\mu\alpha.M[(\alpha L) := (\alpha (L N))]\}^\diamond$ .

(3) follows immediately from (1) and (2).  $\square$

**Lemma 4.4** Let  $M \in \mathcal{M}'$ . If  $M^\diamond \in SN_\beta$ , then  $M \in SN_{\beta\mu}$ .

**Proof** Let  $n = \eta_\beta(M^\diamond) + 1$ . If  $M \notin SN_{\beta\mu}$ , there is  $N$  such that  $M \triangleright_{\beta\mu}^* N$  and  $lg(M \triangleright_{\beta\mu}^* N) \geq n$ . Thus, by lemma 4.3,  $M^\diamond \triangleright_\beta^* N^\diamond$  and  $lg(M^\diamond \triangleright_\beta^* N^\diamond) \geq lg(M \triangleright_{\beta\mu}^* N) \geq \eta_\beta(M^\diamond) + 1$ . This contradicts the definition of  $\eta_\beta(M^\diamond)$ .  $\square$

**Lemma 4.5** If  $\Gamma \vdash_{S^\mu} M : A$ , then  $\Gamma^\diamond \vdash_{S^e} M^\diamond : A$  where  $\Gamma^\diamond$  is obtained from  $\Gamma$  by replacing  $\alpha : \neg B$  by  $x_\alpha : \neg B$ .

**Proof** By induction on the typing  $\Gamma \vdash_{S^\mu} M : A$ . Use lemma 4.1.  $\square$

**Theorem 4.1** The simply typed  $\lambda\mu$ -calculus is strongly normalizing for  $\triangleright_{\beta\mu}$ .

**Proof** A consequence of lemmas 4.4, 4.5 and theorem 3.1.  $\square$

## 5 Some classical results on the $\lambda\mu$ -calculus

The translation given in the next section needs the addition, to the  $\lambda\mu$ -calculus, of the following reductions rules.

$$\begin{aligned} & (\beta \mu\alpha.M) \triangleright_{\rho} M[\alpha := \beta] \\ & \mu\alpha.(\alpha M) \triangleright_{\theta} M \text{ if } \alpha \notin Fv(M) \end{aligned}$$

We will need some classical results about these new rules. For the paper to remain self-contained, we also have added their proofs. The reader who already knows these results or is only interested by the results of the next section may skip this part.

### 5.1 Adding $\triangleright_{\rho\theta}$ does not change $SN$

**Theorem 5.1** *Let  $M \in \mathcal{M}'$  be such that  $M \in SN_{\beta\mu}$ . Then  $M \in SN_{\beta\mu\rho\theta}$ .*

**Proof** This follows from the fact that  $\triangleright_{\rho\theta}$  can be postponed (theorem 5.2 below) and that  $\triangleright_{\rho\theta}$  is strongly normalizing (lemma 5.1 below).  $\square$

**Lemma 5.1** *The reduction  $\triangleright_{\rho\theta}$  is strongly normalizing.*

**Proof** The reduction  $\triangleright_{\rho\theta}$  decreases the size.  $\square$

**Theorem 5.2** *Let  $M, N$  be such that  $M \triangleright_{\beta\mu\rho\theta}^* N$  and  $lg_{\beta\mu}(M \triangleright_{\beta\mu\rho\theta}^* N) \geq 1$ . Then  $M \triangleright_{\beta\mu}^+ P \triangleright_{\rho\theta}^* N$  for some  $P$ .*

This is proved in two steps. First we show that the  $\triangleright_{\theta}$ -reduction can be postponed w.r.t. to  $\triangleright_{\beta\mu\rho}$  (theorem 5.3). Then we show that the  $\triangleright_{\rho}$ -rule can be postponed w.r.t. the remaining rules (theorem 5.4).

**Definition 5.1** *Say that  $P \triangleright_{\mu_0} P'$  if  $P = (\mu\alpha M N)$ ,  $P' = \mu\alpha M[(\alpha L) := (\alpha(LN))]$  and  $\alpha$  occurs at most once in  $M$*

**Lemma 5.2** *1. Assume  $M \triangleright_{\theta} P \triangleright_{\beta\mu} N$ . Then either  $M \triangleright_{\beta\mu} Q \triangleright_{\theta}^* N$  for some  $Q$  or  $M \triangleright_{\mu_0} R \triangleright_{\beta\mu} Q \triangleright_{\theta} N$  for some  $R, Q$ .*

*2. Let  $M \triangleright_{\theta} P \triangleright_{\mu_0} N$ . Then either  $M \triangleright_{\mu_0} Q \triangleright_{\theta} N$  for some  $Q$  or  $M \triangleright_{\mu_0} R \triangleright_{\mu_0} Q \triangleright_{\theta} N$  for some  $R, Q$ .*

*3. Let  $M \triangleright_{\theta} P \triangleright_{\rho} N$ . Then  $M \triangleright_{\rho} Q \triangleright_{\theta} N$ .*

**Proof** By induction on  $M$ .  $\square$

**Lemma 5.3** *Let  $M \triangleright_{\theta}^* P \triangleright_{\mu_0} N$ . Then,  $M \triangleright_{\mu_0}^* Q \triangleright_{\theta}^* N$  for some  $Q$  such that  $lg(M \triangleright_{\theta}^* P) = lg(Q \triangleright_{\theta}^* N)$ .*

**Proof** By induction on  $lg(M \triangleright_{\theta}^* P)$ .  $\square$

**Theorem 5.3** *Let  $M \triangleright_{\theta}^* P \triangleright_{\beta\mu\rho} N$ . Then,  $M \triangleright_{\beta\mu\rho}^+ Q \triangleright_{\theta}^* N$  for some  $Q$ .*

**Proof** By induction on  $lg(M \triangleright_{\theta}^* P)$ .  $\square$

**Lemma 5.4** *1. Let  $M \triangleright_{\rho} P \triangleright_{\beta} N$ . Then  $M \triangleright_{\beta} Q \triangleright_{\rho}^* N$  for some  $Q$ .*

*2. Let  $M, M', N$  be such that  $M \triangleright_{\rho} M'$  and  $\alpha \notin Fv(N)$ . Then either  $M[(\alpha L) := (\alpha(LN))] \triangleright_{\rho} M'[(\alpha L) := (\alpha(LN))]$  or  $M[(\alpha L) := (\alpha(LN))] \triangleright_{\mu} P \triangleright_{\rho} M'[(\alpha L) := (\alpha(LN))]$  for some  $P$ .*

*3. Let  $M \triangleright_{\rho} P \triangleright_{\mu} N$ . Then  $M \triangleright_{\mu} Q \triangleright_{\rho}^* N$  for some  $Q$ .*

**Proof** By induction on  $M$ .  $\square$



**Theorem 5.4** *Let  $M \triangleright_{\rho}^* P \triangleright_{\beta\mu} N$ . Then  $M \triangleright_{\beta\mu} Q \triangleright_{\rho}^* N$  for some  $Q$ .*

**Proof** By induction on  $lg(M \triangleright_{\rho}^* P)$ . □

## 5.2 Commutation lemmas

The goal of this section is lemma 5.7 below. Its proof necessitates some preliminary lemmas.

**Lemma 5.5** *1. If  $M \triangleright_{\rho} P$  and  $M \triangleright_{\rho\theta} Q$ , then  $P = Q$  or  $P \triangleright_{\rho\theta} N$  and  $Q \triangleright_{\rho} N$  for some  $N$ .*

*2. If  $M \triangleright_{\rho} P$  and  $M \triangleright_{\beta\mu} Q$ , then  $P \triangleright_{\beta\mu} N$  and  $Q \triangleright_{\rho}^* N$  for some  $N$ .*

**Proof** By simple case analysis. □

**Lemma 5.6** *1. If  $M \triangleright_{\rho}^* P$  and  $M \triangleright_{\rho\theta^1} Q$ , then  $P \triangleright_{\rho\theta^1} N$  and  $Q \triangleright_{\rho}^* N$  for some  $N$ .*

*2. If  $M \triangleright_{\rho}^* P$  and  $M \triangleright_{\rho\theta}^* Q$ , then  $P \triangleright_{\rho\theta}^* N$  and  $Q \triangleright_{\rho}^* N$  for some  $N$ .*

*3. If  $M \triangleright_{\rho}^* P$  and  $M \triangleright_{\beta\mu} Q$ , then  $P \triangleright_{\beta\mu} N$  and  $Q \triangleright_{\rho}^* N$  for some  $N$ .*

**Proof**

1. By induction on  $\eta_{\rho}(M)$ . Use (1) of lemma 5.5.

2. By induction on  $lg(M \triangleright_{\rho\theta}^* Q)$ . Use (1).

3. By induction on  $\eta_{\rho}(M)$ . Use (2) of lemma 5.5. □

**Lemma 5.7** *If  $M \triangleright_{\rho}^* P$  and  $M \triangleright_{\beta\mu\rho\theta}^* Q$ , then  $P \triangleright_{\beta\mu\rho\theta}^* N$ ,  $Q \triangleright_{\rho}^* N$  for some  $N$  and  $lg_{\beta\mu}(P \triangleright_{\beta\mu\rho\theta}^* N) = lg_{\beta\mu}(M \triangleright_{\beta\mu\rho\theta}^* Q)$ .*

**Proof** By induction on  $lg_{\beta\mu}(M \triangleright_{\beta\mu\rho\theta}^* Q)$ . If  $M \triangleright_{\beta\mu\rho\theta}^* M_1 \triangleright_{\beta\mu} M_2 \triangleright_{\rho\theta}^* Q$ , then, by induction hypothesis,  $P \triangleright_{\beta\mu\rho\theta}^* N_1$ ,  $M_1 \triangleright_{\rho}^* N_1$  and  $lg_{\beta\mu}(P \triangleright_{\beta\mu\rho\theta}^* N_1) = lg_{\beta\mu}(M \triangleright_{\rho\theta}^* M_1)$ . By (3) of lemma 5.6,  $N_1 \triangleright_{\beta\mu} N_2$  and  $M_2 \triangleright_{\rho}^* N_2$  for some  $N_2$ . And finally, by (2) of lemma 5.6,  $N_2 \triangleright_{\rho\theta}^* N$  and  $Q \triangleright_{\rho}^* N$  for some  $N$ . Thus  $P \triangleright_{\beta\mu\rho\theta}^* N$ ,  $Q \triangleright_{\rho}^* N$  and  $lg_{\beta\mu}(P \triangleright_{\beta\mu\rho\theta}^* N) = lg_{\beta\mu}(M \triangleright_{\beta\mu\rho\theta}^* Q)$ . □

## 6 A translation of the $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus into the $\lambda\mu$ -calculus

We code  $\wedge$  and  $\vee$  by their usual equivalent (using  $\rightarrow$  and  $\perp$ ) in classical logic.

**Definition 6.1** *We define the translation  $A^{\circ} \in \mathcal{T}$  of a type  $A \in \mathcal{T}'$  by induction on  $A$  as follows.*

- $\{A\}^{\circ} = A$  for  $A \in \mathcal{A} \cup \{\perp\}$
- $\{A_1 \rightarrow A_2\}^{\circ} = A_1^{\circ} \rightarrow A_2^{\circ}$
- $\{A_1 \wedge A_2\}^{\circ} = \neg(A_1^{\circ} \rightarrow (A_2^{\circ} \rightarrow \perp))$
- $\{A_1 \vee A_2\}^{\circ} = \neg A_1^{\circ} \rightarrow (\neg A_2^{\circ} \rightarrow \perp)$

**Lemma 6.1** *For every  $A \in \mathcal{T}'$ ,  $A^{\circ}$  is classically equivalent to  $A$ .*

**Proof** By induction on  $A$ . □

**Definition 6.2** Let  $\varphi$  a special  $\mu$ -variable. A term  $M \in \mathcal{M}''$  is translated into a  $\lambda\mu$ -term  $M^\circ$  as follows:

- $\{x\}^\circ = x$
- $\{\lambda x.M\}^\circ = \lambda x.M^\circ$
- $\{(M N)\}^\circ = (M^\circ N^\circ)$
- $\{\mu\alpha.M\}^\circ = \mu\alpha.M^\circ$
- $\{(\alpha M)\}^\circ = (\alpha M^\circ)$
- $\{\langle M, N \rangle\}^\circ = \lambda x.(x M^\circ N^\circ)$
- $\{M\pi_i\}^\circ = \mu\alpha.(\varphi (M^\circ \lambda x_1.\lambda x_2.\mu\gamma.(\alpha x_i)))$  where  $\gamma$  is a fresh variable
- $\{M [x_1.N_1, x_2.N_2]\}^\circ = \mu\alpha.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\alpha N_1^\circ) \lambda x_2.\mu\gamma.(\alpha N_2^\circ)))$  where  $\gamma$  is a fresh variable
- $\{\omega_i M\}^\circ = \lambda x_1.\lambda x_2.(x_i M^\circ)$

### Remarks

- The introduction of the free variable  $\varphi$  in the definition of  $\{M [x_1.N_1, x_2.N_2]\}^\circ$  and  $\{M\pi_i\}^\circ$  is not necessary for lemma 6.3. The reason of this introduction is that, otherwise, to simulate the reductions of the  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus we would have to introduce new reductions rules for the  $\lambda\mu$ -calculus and thus to prove  $SN$  of this extension whereas, using  $\varphi$ , the simulation is done with the usual rules of the  $\lambda\mu$ -calculus.
- There is another way of coding  $\wedge$  and  $\vee$  by using intuitionistic second order logic.

$$\begin{aligned} & - \{A_1 \wedge A_2\}^\circ = \forall X((A_1^\circ \rightarrow (A_2^\circ \rightarrow X)) \rightarrow X) \\ & - \{A_1 \vee A_2\}^\circ = \forall X((A_1^\circ \rightarrow X) \rightarrow ((A_2^\circ \rightarrow X) \rightarrow X)) \end{aligned}$$

The translation of  $\{\langle M, N \rangle\}^\circ$  and  $\{\omega_i M\}^\circ$  are the same but the translation of  $\{M\pi_i\}^\circ$  will be  $(M^\circ \lambda x_1.\lambda x_2.x_i)$  and the one of  $\{M [x_1.N_1, x_2.N_2]\}^\circ$  would be  $(M^\circ \lambda x_1.N_1^\circ \lambda x_2.N_2^\circ)$ . But it is easily checked that the permutative conversions are not correctly simulated by this translation whereas, in our translation, they are.

- Finally note that, as given in definition 2.2, the reduction rules for the  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus do not include  $\triangleright_\rho$  and  $\triangleright_\theta$ . We could have added them and the given translation would have worked in a similar way. We decided not to do so (although these rules were already considered by Parigot) because they, usually, are not included neither in the  $\lambda\mu$ -calculus nor in the  $\lambda\mu^{\rightarrow\wedge\vee}$ -calculus. Moreover some of the lemma given below would need a bit more complex statement.

**Lemma 6.2** 1.  $\{M[x := N]\}^\circ = M^\circ[x := N^\circ]$ .

2.  $\{M[(\alpha N) := (\alpha (N \varepsilon))]\}^\circ = M^\circ[(\alpha N^\circ) := (\alpha \{(N \varepsilon)\}^\circ)]$ .

**Proof** By induction on  $M$ . □

**Lemma 6.3** If  $\Gamma \vdash_{\mathcal{S}^{\rightarrow\wedge\vee}} M : A$ , then  $\Gamma^\circ \vdash_{\mathcal{S}^\mu} M^\circ : A^\circ$  where  $\Gamma^\circ$  is obtained from  $\Gamma$  by replacing all the types by their translations and by declaring  $\varphi$  of type  $\neg\perp$ .

**Proof** By induction on a derivation of  $\Gamma \vdash_{\mathcal{S}^{\rightarrow\wedge\vee}} M : A$ . □

**Lemma 6.4** *Let  $M \in \mathcal{M}''$ . If  $M \triangleright N$ , then there is  $P \in \mathcal{M}'$  such that  $M^\circ \triangleright_{\beta\mu\rho\theta}^* P$ ,  $N^\circ \triangleright_{\rho}^* P$  and  $lg_{\beta\mu}(M^\circ \triangleright_{\beta\mu\rho\theta}^* P) \geq 1$ .*

**Proof** By case analysis. The details are given in the appendix, section 8.  $\square$

**Lemma 6.5** *Let  $M \in \mathcal{M}''$ . If  $M \triangleright^* N$ , then there is  $P \in \mathcal{M}'$  such that  $M^\circ \triangleright_{\beta\mu\rho\theta}^* P$ ,  $N^\circ \triangleright_{\rho}^* P$  and  $lg_{\beta\mu}(M^\circ \triangleright_{\beta\mu\rho\theta}^* P) \geq lg(M \triangleright^* N)$ .*

**Proof** By induction on  $lg(M \triangleright^* N)$ . If  $M \triangleright^* L \triangleright N$ , then, by induction hypothesis, there is  $Q \in \mathcal{M}'$  such that  $M^\circ \triangleright_{\beta\mu\rho\theta}^* Q$ ,  $L^\circ \triangleright_{\rho}^* Q$  and  $lg_{\beta\mu}(M^\circ \triangleright_{\beta\mu\rho\theta}^* Q) \geq lg(M \triangleright^* L)$ . By lemma 6.4, there is a  $R \in \mathcal{M}'$  such that  $L^\circ \triangleright_{\beta\mu\rho\theta}^* R$ ,  $N^\circ \triangleright_{\rho}^* R$  and  $lg_{\beta\mu}(L^\circ \triangleright_{\beta\mu\rho\theta}^* R) \geq 1$ . Then, by lemma 5.7, there is a  $P \in \mathcal{M}'$  such that  $Q \triangleright_{\beta\mu\rho\theta}^* P$ ,  $R \triangleright_{\rho}^* P$  and  $lg_{\beta\mu}(Q \triangleright_{\beta\mu\rho\theta}^* P) \geq lg_{\beta\mu}(L^\circ \triangleright_{\beta\mu\rho\theta}^* R) \geq 1$ . Thus  $M^\circ \triangleright_{\beta\mu\rho\theta}^* P$ ,  $N^\circ \triangleright_{\rho}^* P$  and  $lg_{\beta\mu}(M^\circ \triangleright_{\beta\mu\rho\theta}^* P) \geq lg(M \triangleright^* N)$ .  $\square$

**Lemma 6.6** *Let  $M \in \mathcal{M}''$  be such that  $M^\circ \in SN_{\beta\mu\rho\theta}$ . Then  $M \in SN$ .*

**Proof** Since  $M^\circ \in SN_{\beta\mu\rho\theta}$ , let  $n$  be the maximum of  $\triangleright_{\beta\mu}$  steps in the reductions of  $M^\circ$ . If  $M \notin SN$ , by lemma 5.1, let  $N$  be such that  $M \triangleright^* N$  and  $lg_{\beta\mu}(M \triangleright^* N) \geq n + 1$ . By lemma 6.5, there is  $P$  such that  $M^\circ \triangleright_{\beta\mu\rho\theta}^* P$  and  $lg_{\beta\mu}(M^\circ \triangleright_{\beta\mu\rho\theta}^* P) \geq lg_{\beta\mu}(M \triangleright^* N) \geq n + 1$ . Contradiction.  $\square$

**Theorem 6.1** *Every typed  $\lambda\mu^{-\wedge\vee}$ -term is strongly normalizable.*

**Proof** A consequence of theorems 4.1, 5.1 and lemmas 6.6, 6.3.  $\square$

## 7 Recursive equations on types

We study here systems where equations on types are allowed. These types are usually called recursive types. The subject reduction and the decidability of type assignment are preserved but the strong normalization may be lost. For example, with the equation  $X = X \rightarrow T$ , the term  $(\Delta \Delta)$  where  $\Delta = \lambda x.(xx)$  is typable but is not strongly normalizing. With the equation  $X = X \rightarrow X$ , every term can be typed. By making some natural assumptions on the recursive equations the strong normalization can be preserved. The simplest condition is to accept the equation  $X = F$  (where  $F$  is a type containing the variable  $X$ ) only when the variable  $X$  is positive in  $F$ . For a set  $\{X_i = F_i / i \in I\}$  of mutually recursive equations, Mendler [10] has given a very simple and natural condition that ensures the strong normalization of the system. He also showed that the given condition is necessary to have the strong normalization.

Mendler's result concerns the implicative fragment of intuitionistic logic. We extend here his result to full classical logic. We now assume  $\mathcal{A}$  contains a specified subset  $\mathcal{X} = \{X_i / i \in I\}$ .

**Definition 7.1** *Let  $X \in \mathcal{X}$ . We define the subsets  $\mathcal{P}^+(X)$  and  $\mathcal{P}^-(X)$  of  $\mathcal{T}$  (resp.  $\mathcal{T}'$ ) as follows.*

- $X \in \mathcal{P}^+(X)$
- If  $A \in (\mathcal{X} - \{X\}) \cup \mathcal{A}$ , then  $A \in \mathcal{P}^+(X) \cap \mathcal{P}^-(X)$ .
- If  $A \in \mathcal{P}^-(X)$  and  $B \in \mathcal{P}^+(X)$ , then  $A \rightarrow B \in \mathcal{P}^+(X)$  and  $B \rightarrow A \in \mathcal{P}^-(X)$ .
- If  $A, B \in \mathcal{P}^+(X)$ , then  $A \wedge B, B \vee A \in \mathcal{P}^+(X)$ .
- If  $A, B \in \mathcal{P}^-(X)$ , then  $A \wedge B, B \vee A \in \mathcal{P}^-(X)$ .

**Definition 7.2** • *Let  $\mathcal{F} = \{F_i / i \in I\}$  be a set of types in  $\mathcal{T}$  (resp. in  $\mathcal{T}'$ ). The congruence  $\approx$  generated by  $\mathcal{F}$  in  $\mathcal{T}$  (resp. in  $\mathcal{T}'$ ) is the least congruence such that  $X_i \approx F_i$  for each  $i \in I$ .*

- We say that  $\approx$  is good if, for each  $X \in \mathcal{X}$ , if  $X \approx A$ , then  $A \in \mathcal{P}^+(X)$ .

## 7.1 Strong normalization of $\mathcal{S}_{\approx}^{\mu}$

Let  $\approx$  be the congruence generated by a set  $\mathcal{F}$  of types of  $\mathcal{T}$ .

**Theorem 7.1 (Mendler)** *If  $\approx$  is good, then the system  $\mathcal{S}_{\approx}$  is strongly normalizing.*

**Proof** See [10] for the original proof and [5] for an arithmetical one.  $\square$

**Lemma 7.1** *If  $\Gamma \vdash_{\mathcal{S}_{\approx}^{\mu}} M : A$ , then  $\Gamma^{\circ} \vdash_{\mathcal{S}_{\approx}^{\mu}} M^{\circ} : A$ .*

**Proof** By induction on the typing  $\Gamma \vdash_{\mathcal{S}_{\approx}^{\mu}} M : A$ .  $\square$

**Theorem 7.2** *If  $\approx$  is good, then the system  $\mathcal{S}_{\approx}^{\mu}$  is strongly normalizing.*

**Proof** Let  $M \in \mathcal{M}'$  be a term typable in  $\mathcal{S}_{\approx}^{\mu}$ . By lemma 4.4, it is enough to show that  $M^{\circ} \in SN_{\beta}$ . This follows immediately from theorem 7.1 and lemma 7.1. Note that, in [5], we also had given a direct proof of this result.  $\square$

## 7.2 Strong normalization of $\mathcal{S}_{\approx}^{\rightarrow \wedge \vee}$

Let  $\mathcal{F} = \{F_i / i \in I\}$  be a set of types in  $\mathcal{T}'$  and let  $\mathcal{F}^{\circ} = \{F_i^{\circ} / i \in I\}$  be its translation in  $\mathcal{T}$ . Let  $\approx$  be the congruence generated by  $\mathcal{F}$  in  $\mathcal{T}'$  and let  $\approx^{\circ}$  be the congruence generated by  $\mathcal{F}^{\circ}$  in  $\mathcal{T}$ .

**Lemma 7.2** *1. If  $\approx$  is good, then so is  $\approx^{\circ}$ .*

*2. If  $A \approx B$ , then  $A^{\circ} \approx^{\circ} B^{\circ}$ .*

**Proof**

1. Just note that  $A_1^{\circ}$  and  $A_2^{\circ}$  are in positive position in  $\{A_1 \wedge A_2\}^{\circ}$  and  $\{A_1 \vee A_2\}^{\circ}$ .
2. By induction on the proof of  $A \approx B$ .  $\square$

**Lemma 7.3** *If  $\Gamma \vdash_{\mathcal{S}_{\approx}^{\rightarrow \wedge \vee}} M : A$ , then  $\Gamma^{\circ} \vdash_{\mathcal{S}_{\approx^{\circ}}^{\mu}} M^{\circ} : A^{\circ}$ .*

**Proof** By induction on a derivation of  $\Gamma \vdash_{\mathcal{S}_{\approx}^{\rightarrow \wedge \vee}} M : A$ .  $\square$

**Theorem 7.3** *If  $\approx$  is good, then the system  $\mathcal{S}_{\approx}^{\rightarrow \wedge \vee}$  is strongly normalizing.*

**Proof** Let  $M \in \mathcal{M}''$  be a term typable in  $\mathcal{S}_{\approx}^{\rightarrow \wedge \vee}$ , then, by lemma 7.3,  $M^{\circ}$  is typable in  $\mathcal{S}_{\approx^{\circ}}^{\mu}$ . Since, by lemma 7.2,  $\approx^{\circ}$  is good, then, by theorems 7.2 and 5.1,  $M^{\circ} \in SN_{\beta\mu\rho\theta}$ , thus by lemma 6.6,  $M \in SN$ .  $\square$

### Remark

Note that, in definition 7.1, it was necessary to define, for  $X$  to be positive in a conjunction and a disjunction, as being positive in both formulas since, otherwise, the previous theorem will not be true as the following examples shows. Let  $A, B$  be any types. Note that, in particular,  $X$  may occur in  $A$  and  $B$  and thus the negative occurrence of  $X$  in  $X \rightarrow B$  is enough to get a non normalizing term.

- Let  $F = A \wedge (X \rightarrow B)$  and  $\approx$  be the congruence generated by  $X \approx F$ . Let  $M = \lambda x.((x \pi_2) x)$ . Then  $y : A \vdash_{\mathcal{S}_{\approx}^{\rightarrow \wedge \vee}} (M \langle y, M \rangle) : B$  and  $(M \langle y, M \rangle) \notin SN$  since it reduces to itself.
- Let  $G = A \vee (X \rightarrow B)$  and  $\approx$  be the congruence generated by  $X \approx G$ . Let  $N = \lambda x(x [y.y, z.(z \omega_2 z)])$ . Then  $\vdash_{\mathcal{S}_{\approx}^{\rightarrow \wedge \vee}} (N \omega_2 N) : B$  and  $(N \omega_2 N) \notin SN$  since it reduces to itself.

## References

- [1] Y. Andou. *Church-Rosser property of simple reduction for full first-order classical natural deduction*. Annals of Pure and Applied logic 119, pp. 225-237, 2003.
- [2] P. Battyanyi. *Normalization properties of symmetric logical calculi*. PhD thesis. Université de Chambéry. 2007.
- [3] R. David and K. Nour. *A short proof of the strong normalization of the simply typed  $\lambda\mu$ -calculus*. Schedae Informaticae 12, pp. 27-33, 2003.
- [4] R. David and K. Nour. *A short proof of the strong normalization of classical natural deduction with disjunction*. Journal of Symbolic Logic, vol 68, num 4, pp. 1277-1288, 2003.
- [5] R. David and K. Nour. *An arithmetical proof of the strong normalization for the lambda-calculus with recursive equations on types*. TLCA 2007, LNCS 4583, pp. 84-101, 2007.
- [6] F. Joachimski and R. Matthes. *Short proofs of normalization for the simply-typed lambda-calculus, permutative conversions and Gödel's T*. Archive for Mathematical Logic, 42(1), pp. 59-87, 2003.
- [7] P. de Groote. *Strong normalization of classical natural deduction with disjunction*. TLCA 2001. LNCS 2044, pp. 182-196, 2001.
- [8] R. Matthes. *Stabilization - An Alternative to Double-Negation Translation for Classical Natural Deduction*. Logic Colloquium 2003, Lecture Notes in Logic, vol. 24, pp. 167-199, 2006.
- [9] R. Matthes. *Non-strictly positive fixed-points for classical natural deduction*. Annals of pure and Applied logic 133 (1-3), pp. 205-230, 2005.
- [10] N. P. Mendler. *Recursive Types and Type Constraints in Second-Order Lambda Calculus*. LICS, pp. 30-36, 1987.
- [11] N. P. Mendler. *Inductive Types and Type Constraints in the Second-Order Lambda Calculus*. Annals of pure and Applied logic 51 (1-2), pp. 159-172, 1991.
- [12] K. Nakazawa and M. Tatsuta. *Strong normalization of classical natural deduction with disjunctions*. Annals of Pure and Applied Logic 153 (1-3), pp. 21-37, 2008.
- [13] K. Nour and K. Saber. *A semantical proof of strong normalization theorem for full propositional classical natural deduction*. Archive for Mathematical Logic, vol 45, pp. 357-364, 2005.
- [14] M. Parigot.  *$\lambda\mu$ -calculus: An algorithm interpretation of classical natural deduction*. Lecture Notes in Artificial Intelligence, vol 624, pp. 190-201. Springer Verlag, 1992.
- [15] M. Parigot. *Proofs of strong normalization for second order classical natural deduction*. Journal of Symbolic Logic, vol 62 (4), pp. 1461-1479, 1997.
- [16] W. Py. *Confluence en  $\lambda\mu$ -calcul*. PhD thesis. Université de Chambéry. 1998.
- [17] M. Tatsuta. *Simple saturated sets for disjunction and second-order existential quantification*. TLCA 2007, LNCS 4583, pp. 366-380, 2007.
- [18] A. Wojdyga. *Short proofs of strong normalization*. <http://arxiv.org/abs/0804.2535v1>

## 8 Appendix

**Lemma 6.4** *Let  $M \in \mathcal{M}''$ . If  $M \triangleright N$ , then there is  $P \in \mathcal{M}'$  such that  $M^\circ \triangleright_{\beta\mu\rho\theta}^* P$ ,  $N^\circ \triangleright_\rho^* P$  and  $lg_{\beta\mu}(M^\circ \triangleright_{\beta\mu\rho\theta}^* P) \geq 1$ .*

**Proof** We consider only the case of redexes.

- If  $(\lambda x.M N) \triangleright M[x := N]$ , then
 
$$\{(\lambda x.M N)\}^\circ = (\lambda x.M^\circ N^\circ) \triangleright_\beta M^\circ[x := N^\circ] = \{M[x := N]\}^\circ.$$
- If  $(\langle M_1, M_2 \rangle \pi_i) \triangleright M_i$ , then
 
$$\begin{aligned} \{(\langle M_1, M_2 \rangle \pi_i)\}^\circ &= \mu\alpha.(\varphi (\lambda x.(x M_1^\circ M_2^\circ) \lambda x_1.\lambda x_2.\mu\gamma.(\alpha x_i))) \\ &\triangleright_\beta^+ \mu\alpha.(\varphi \mu\gamma.(\alpha M_i^\circ)) \triangleright_\rho \mu\alpha.(\alpha M_i^\circ) \triangleright_\theta M_i^\circ. \end{aligned}$$
- If  $(\omega_i M [x_1.N_1, x_2.N_2]) \triangleright N_i[x_i := M]$ , then
 
$$\begin{aligned} \{(\omega_i M [x_1.N_1, x_2.N_2])\}^\circ &= \\ &\mu\alpha.(\varphi (\lambda x_1.\lambda x_2.(x_i M^\circ) \lambda x_1.\mu\gamma.(\alpha N_1^\circ) \lambda x_2.\mu\gamma.(\alpha N_2^\circ))) \\ &\triangleright_\beta^+ \mu\alpha.(\varphi \mu\gamma.(\alpha N_i^\circ[x_i := M^\circ])) \triangleright_\rho \mu\alpha.(\alpha N_i^\circ[x_i := M^\circ]) \triangleright_\theta N_i^\circ[x_i := M^\circ] \\ &= \{N_i[x_i := M]\}^\circ. \end{aligned}$$
- If  $(M [x_1.N_1, x_2.N_2] N) \triangleright (M [x_1.(N_1 N), x_2.(N_2 N)])$ , then
 
$$\begin{aligned} \{(M [x_1.N_1, x_2.N_2] N)\}^\circ &= \\ &(\mu\alpha.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\alpha N_1^\circ) \lambda x_2.\mu\gamma.(\alpha N_2^\circ)))) N^\circ \\ &\triangleright_\mu \mu\alpha.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\alpha (N_1^\circ N^\circ)) \lambda x_2.\mu\gamma.(\alpha (N_2^\circ N^\circ)))) \\ &= \mu\alpha.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\alpha \{(N_1 N)\}^\circ) \lambda x_2.\mu\gamma.(\alpha \{(N_2 N)\}^\circ))) \\ &= \{(M [x_1.(N_1 N), x_2.(N_2 N)])\}^\circ. \end{aligned}$$
- If  $(M [x_1.N_1, x_2.N_2] \pi_i) \triangleright (M [x_1.(N_1 \pi_i), x_2.(N_2 \pi_i)])$ , then
 
$$\begin{aligned} \{(M [x_1.N_1, x_2.N_2] \pi_i)\}^\circ &= \\ &\mu\alpha.(\varphi (\mu\beta.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\beta N_1^\circ) \lambda x_2.\mu\gamma.(\beta N_2^\circ))) \lambda y_1.\lambda y_2.\mu\gamma.(\alpha y_i))) \\ &\triangleright_\mu \mu\alpha.(\varphi \mu\beta.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\beta (N_1^\circ \lambda y_1.\lambda y_2.\mu\gamma.(\alpha y_i)))) \\ &\lambda x_2.\mu\gamma.(\beta (N_2^\circ \lambda y_1.\lambda y_2.\mu\gamma.(\alpha y_i))))) \\ &\triangleright_\rho \mu\alpha.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\varphi (N_1^\circ \lambda y_1.\lambda y_2.\mu\gamma.(\alpha y_i))) \\ &\lambda x_2.\mu\gamma.(\varphi (N_2^\circ \lambda y_1.\lambda y_2.\mu\gamma.(\alpha y_i))))) = P. \\ &\text{and } \{(M [x_1.(N_1 \pi_i), x_2.(N_2 \pi_i)])\}^\circ = \\ &\mu\beta(\varphi (M^\circ \lambda x_1.\mu\gamma.(\beta \mu\alpha(\varphi (N_1^\circ \lambda y_1.\lambda y_2.\mu\gamma.(\alpha y_i)))) \\ &\lambda x_2.\mu\gamma.(\beta \mu\alpha(\varphi (N_2^\circ \lambda y_1.\lambda y_2.\mu\gamma.(\alpha y_i))))) \triangleright_\rho^+ P. \end{aligned}$$
- If  $(M [x_1.N_1, x_2.N_2] [y_1.L_1, y_2.L_2]) \triangleright$   
 $(M [x_1.(N_1 [y_1.L_1, y_2.L_2]), x_2.(N_2 [y_1.L_1, y_2.L_2])])$ , then
 
$$\begin{aligned} \{(M [x_1.N_1, x_2.N_2] [y_1.L_1, y_2.L_2])\}^\circ &= \\ &\mu\alpha.(\varphi (\mu\beta.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\beta N_1^\circ) \lambda x_2.\mu\gamma.(\beta N_2^\circ))) \lambda y_1.\mu\gamma.(\alpha L_1^\circ) \lambda y_2.\mu\gamma.(\alpha L_2^\circ))) \\ &\triangleright_\mu \mu\alpha.(\varphi \mu\beta.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\beta (N_1^\circ \lambda y_1.\mu\gamma.(\alpha L_1^\circ) \lambda y_2.\mu\gamma.(\alpha L_2^\circ))) \\ &\lambda x_2.\mu\gamma.(\beta (N_2^\circ \lambda y_1.\mu\gamma.(\alpha L_1^\circ) \lambda y_2.\mu\gamma.(\alpha L_2^\circ))))) \\ &\triangleright_\rho \mu\alpha.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\varphi (N_1^\circ \lambda y_1.\mu\gamma.(\alpha L_1^\circ) \lambda y_2.\mu\gamma.(\alpha L_2^\circ))) \\ &\lambda x_2.\mu\gamma.(\varphi (N_2^\circ \lambda y_1.\mu\gamma.(\alpha L_1^\circ) \lambda y_2.\mu\gamma.(\alpha L_2^\circ)))) = P. \\ &\text{and } \{(M [x_1.(N_1 [y_1.L_1, y_2.L_2]), x_2.(N_2 [y_1.L_1, y_2.L_2])])\}^\circ = \\ &\mu\beta.(\varphi (M^\circ \lambda x_1.\mu\gamma.(\beta \mu\alpha.(\varphi (N_1^\circ \lambda y_1.\mu\gamma.(\alpha L_1^\circ) \lambda y_2.\mu\gamma.(\alpha L_2^\circ))) \\ &\lambda x_2.\mu\gamma.(\beta \mu\alpha.(\varphi (N_2^\circ \lambda y_1.\mu\gamma.(\alpha L_1^\circ) \lambda y_2.\mu\gamma.(\alpha L_2^\circ))))) \triangleright_\rho^+ P. \end{aligned}$$

- If  $(\mu\alpha.M N) \triangleright \mu\alpha.M[(\alpha L) := (\alpha (L N))]$ , then
 
$$\begin{aligned} \{(\mu\alpha.M N)\}^\circ &= (\mu\alpha.M^\circ N^\circ) \triangleright_\mu \mu\alpha.M^\circ[(\alpha L^\circ) := (\alpha (L^\circ N^\circ))] \\ &= \mu\alpha.M^\circ[(\alpha L^\circ) := (\alpha \{(L N)\}^\circ)] = \{\mu\alpha.M[(\alpha L) := (\alpha (L N))]\}^\circ. \end{aligned}$$
- If  $(\mu\beta.M \pi_i) \triangleright \mu\beta.M[(\beta N) := (\beta (N \pi_i))]$ , then
 
$$\begin{aligned} \{(\mu\beta.M \pi_i)\}^\circ &= \mu\alpha.(\varphi (\mu\beta.M^\circ \lambda x_1.\lambda x_2.\mu\gamma.(\alpha x_i))) \\ &\triangleright_\mu \mu\alpha.(\varphi \mu\beta.M^\circ[(\beta N^\circ) := (\beta (N^\circ \lambda x_1.\lambda x_2.\mu\gamma.(\alpha x_i)))])) \\ &\triangleright_\rho \mu\alpha.M^\circ[(\beta N^\circ) := (\varphi (N^\circ \lambda x_1.\lambda x_2.\mu\gamma.(\alpha x_i)))] = P. \end{aligned}$$
 and  $\{\mu\beta.M[(\beta N) := (\beta (N \pi_i))]\}^\circ = \mu\beta.M^\circ[(\beta N^\circ) := (\beta \mu\alpha.(\varphi (N^\circ \lambda x_1.\lambda x_2.\mu\gamma.(\alpha x_i))))] \triangleright_\rho^* P.$
- If  $(\mu\beta.M [x_1.N_1, x_2.N_2]) \triangleright \mu\beta.M[(\beta N) := (\beta (N [x_1.N_1, x_2.N_2]))]$ , then
 
$$\begin{aligned} \{(\mu\beta.M [x_1.N_1, x_2.N_2])\}^\circ &= \mu\alpha.(\varphi (\mu\beta.M^\circ \lambda x_1.\mu\gamma.(\alpha N_1^\circ) \lambda x_2.\mu\gamma.(\alpha N_2^\circ))) \\ &\triangleright_\mu^+ \mu\alpha.(\varphi \mu\beta.M^\circ[(\beta N^\circ) := (\beta (N^\circ \lambda x_1.\mu\gamma.(\alpha N_1^\circ) \lambda x_2.\mu\gamma.(\alpha N_2^\circ)))])) \\ &\triangleright_\rho \mu\alpha.M^\circ[(\beta N^\circ) := (\varphi (N^\circ \lambda x_1.\mu\gamma.(\alpha N_1^\circ) \lambda x_2.\mu\gamma.(\alpha N_2^\circ)))] = P. \end{aligned}$$
 and  $\{\mu\beta.M[(\beta N) := (\beta (N [x_1.N_1, x_2.N_2]))]\}^\circ = \mu\beta.M^\circ[(\beta N^\circ) := (\beta \mu\alpha.(\varphi (N^\circ \lambda x_1.\mu\gamma.(\alpha N_1^\circ) \lambda x_2.\mu\gamma.(\alpha N_2^\circ)))] \triangleright_\rho^* P.$

□