Strong Normalization of Classical Natural Deduction with Disjunction

Philippe de Groote

LORIA UMR n° 7503 – INRIA
Campus Scientifique, B.P. 239
54506 Vandœuvre lès Nancy Cedex – France
e-mail: Philippe.de.Groote@loria.fr
phone: +33 3 83 59 30 32
fax: +33 3 83 27 83 19

Abstract. We introduce $\boldsymbol{\lambda\mu}^{\to\wedge\vee\perp}$, an extension of Parigot's $\lambda\mu$-calculus where disjunction is taken as a primitive. The associated reduction relation, which includes the permutative conversions related to disjunction, is Church-Rosser, strongly normalizing, and such that the normal deductions satisfy the subformula property. From a computer science point of view, $\boldsymbol{\lambda\mu}^{\to\wedge\vee\perp}$ may be seen as the core of a typed CBN functional language featuring product, coproduct, and control operators.

1 Introduction

During this last decade, several authors have investigated the relation existing between functional control operators, on the one hand, and normalization procedures for classical logic, on the other hand. This research originated in Griffin's observation [13] that Felleisen's control operator $\mathscr{C}$ [8, 9] may be typed with the classical tautology $\neg\neg\alpha \to \alpha$.

Griffin's discovery resulted in lot of work aiming at extending the Curry-Howard correspondance [14] to the case of classical logic [1, 2, 4, 6, 11, 15, 17–19, 26]. On the proof-theoretic side, the problem consists in defining a classical natural deduction system together with an appropriate proof normalization procedure such that the resulting normal deductions satisfy the subformula property. This ensures that proof normalization may be interpreted as an evaluation process.

In fact, it appears *a posteriori* that the line of research opened by Griffin may be traced back to Prawitz [23] who shows how to normalize deductions in the presence of the following classical absurdity rule:

$$\frac{\begin{matrix}[\neg\alpha]\\ \perp\end{matrix}}{\alpha}\ (\perp_c)$$

In order to obtain the subformula property, Prawitz restricts the use of Rule $\perp_c$ to the case where α is atomic. Then he shows how to transform any deduction in order to fulfill this requirement. For instance, any application of Rule $\perp_c$ whose conclusion is an implication may be transformed as follows:

$$\dfrac{\begin{matrix}[\,\neg(\alpha\to\beta)\,]\\ \vdots\;\Pi_1\\ \bot\end{matrix}}{\alpha\to\beta} \qquad\text{reduces to}\qquad \dfrac{\begin{matrix}\dfrac{\dfrac{\overset{(1)}{\neg\beta}\qquad \dfrac{\overset{(2)}{\alpha\to\beta}\quad \overset{(3)}{\alpha}}{\beta}}{\bot}}{\neg(\alpha\to\beta)}{\scriptstyle(2)}\\ \vdots\;\Pi_1\\ \dfrac{\bot}{\beta}{\scriptstyle(1)}\end{matrix}}{\alpha\to\beta}{\scriptstyle(3)} \tag{1}$$

It is worth noting that this reduction, which dates back to 1965, is strongly related to Felleisen's operator $\mathscr{C}$. Indeed it corresponds precisely to the following rewriting rule:

$$\mathscr{C}\,(\lambda k.\,M) \to \lambda n.\,\mathscr{C}\,(\lambda k.\,M[k{:=}\lambda f.\,k\,(f\,n)]),$$

which is reminiscent of one of Felleisen's rules [9].

Prawitz gives similar rules for conjunction and the universal quantifier. For disjunction, however, there is a problem. Indeed, restricting the application of Rule $\bot_c$ to the case where its conclusion is atomic would imply, in the presence of disjunction, the existence of a universal decision procedure. A similar problem arises with the existential quantifier. Consequently Prawitz does not take disjunction and existential quantification as primitives.

A way of circumventing this problem is to observe that reductions akin to (1) are only needed when the conclusion of Rule $\bot_c$ is the principal formula of an elimination rule. In the case of implication, this idea gives rise to the following reduction scheme:

$$\dfrac{\dfrac{\begin{matrix}[\,\neg(\alpha\to\beta)\,]\\ \vdots\;\Pi_1\\ \bot\end{matrix}}{\alpha\to\beta}\qquad \begin{matrix}\vdots\;\Pi_2\\ \alpha\end{matrix}}{\beta} \qquad\text{reduces to}\qquad \begin{matrix}\dfrac{\dfrac{\overset{(1)}{\neg\beta}\qquad \dfrac{\overset{(2)}{\alpha\to\beta}\quad \begin{matrix}\vdots\;\Pi_2\\ \alpha\end{matrix}}{\beta}}{\bot}}{\neg(\alpha\to\beta)}{\scriptstyle(2)}\\ \vdots\;\Pi_1\\ \dfrac{\bot}{\beta}{\scriptstyle(1)}\end{matrix} \tag{2}$$

which is used by both [30] and [26], and which amounts, modulo some additional β-contraction steps, to Parigot's μ-reduction [19]. Applying the same idea to the case of disjunction yields the following figure:

$$\dfrac{\dfrac{\begin{matrix}[\,\neg(\alpha\vee\beta)\,]\\ \vdots\;\Pi_1\\ \bot\end{matrix}}{\alpha\vee\beta}\qquad \begin{matrix}[\,\alpha\,]\\ \vdots\;\Pi_2\\ \gamma\end{matrix}\qquad \begin{matrix}[\,\beta\,]\\ \vdots\;\Pi_3\\ \gamma\end{matrix}}{\gamma} \qquad \text{reduces to} \qquad \dfrac{\begin{matrix}\dfrac{\dfrac{\overset{(1)}{\neg\gamma}\qquad \dfrac{\overset{(2)}{\alpha\vee\beta}\qquad \begin{matrix}[\,\alpha\,]\\ \vdots\;\Pi_2\\ \gamma\end{matrix}\qquad \begin{matrix}[\,\beta\,]\\ \vdots\;\Pi_3\\ \gamma\end{matrix}}{\gamma}}{\bot}}{\neg(\alpha\vee\beta)}\,{\scriptstyle (2)}\\ \vdots\;\Pi_1\\ \bot\end{matrix}}{\gamma}\,{\scriptstyle (1)} \qquad (3)$$

which corresponds precisely to one of the reduction rules proposed in [26]. Reduction (3), however, is not sufficient to solve the problems related to disjunction. Indeed, in order to obtain the subformula property, one also needs the so-called permutative conversions [31]. Consequently, normalization procedures for full classical logic [29, 30] might be rather intricate since they involve three kinds of reduction steps:

– the usual detour conversions of intuitionistic natural deduction,
– conversions related to Rule $\bot_c$,
– permutative conversions related to disjunction.

In the present paper, we revisit this problem from a type-theoretic point of view. Our main contribution is the definition of an extension of the $\lambda\mu$-calculus ($\boldsymbol{\lambda\mu}^{\to\wedge\vee\bot}$) such that:

(a) intuitionistic disjunction—i.e., coproduct—is taken as a primitive;
(b) normal deductions satisfy the subformula property;
(c) the reduction relation is defined by means of local reduction steps;[1]
(d) the reduction relation is proven to be strongly normalizing;
(e) the reduction relation is Church-Rosser;
(f) the reduction relation is defined at the untyped level;[2]
(g) the reduction relation satisfies the subject reduction property.

Properties (e), (f), and (g) are of special interest from a programming language perspective. Property (e) ensures the unicity of the normal forms, which allows the reduction relation to be considered as the core of an operational semantics. Properties (f) and (g), on the other hand, allow the typing information to be ignored at run time.

[1] By a *local* reduction step, we mean a rewriting rule that is compatible with the term formation rules, i.e., a rewriting rule $s \to t$ such that $C[s] \to C[t]$ independently of the shape of the context C.

[2] From a proof-theoretic point of view, it means that the proof normalization steps are defined on the shape of the derivations, according to the inference rules that are used and without any proviso on the formulas that are introduced or eliminated.

To the best of our knowledge, there is no *classical* extension of the simply typed λ-calculus (or equivalently, no normalisation procedure for classical natural deduction) that enjoys all of the above properties. In particular, neither (d) nor (e) are satisfied by [29]. The handling of the permutative conversions in [30] does not satisfy (c) and (e), while the handling of classical negation does not satisfy (f). Finally, in [26], the extension of λ_Δ that satifies (a) does not satisfy (b).

In a series of paper [25, 27, 28], Ritter, Pym, and Wallen introduce an extension of the $\lambda\mu$-calculus that also features disjunction as a primitive. Their system is rather different from ours because they take as primitive a classical form of disjunction that amounts to $\neg A \rightarrow B$. Nevertheless, in [25], Pym and Ritter give a brief account of another extension of the $\lambda\mu$-calculus with an intuitionistic disjunction. However, the reduction rules they give are not sufficient to guarantee that the normal proofs satisfy the subformula property.

In order to prove that our system is strongly normalizable, we use the method that we have introduced in [7]. This yields several auxiliary results that have independent interest:

– We reduce, by finitary means, the strong normalization of classical logic to the strong normalization of intuitionistic implicative logic. Consequently, when combined with an arithmetizable proof of the strong normalization of the simply typed λ-calculus, our method yields a completely arithmetizable proof of the strong normalization of classical propositional logic.
– We prove the strong normalization of Parigot's μ-reduction (together with similar reduction relations) on the *untyped terms*. This result confirms that the reduction relations related to the classical absurdity rule are structural reductions, akin to permutative conversions, which do not carry any real computational content.
– We provide $\boldsymbol{\lambda\mu}^{\rightarrow\wedge\vee\perp}$ with a *continuation passing style* semantics that may be used to construct denotational models of classical logic with disjunction as a primitive.
– Our CPS-translation, since it is defined on the untyped $\lambda\mu$-terms, may be raised to the second order as in [5]. This yields a new proof of the strong normalization of Parigot's original second-order $\lambda\mu$-calculus. In contrast to [20, 21], this proof does not require any extension of the reducibility candidate method since it consists in reducing the problem to the strong normalization of Girard's system F [12].

2 Classical propositional logic as an extension of the $\boldsymbol{\lambda\mu}$-calculus

The types of $\boldsymbol{\lambda\mu}^{\rightarrow\wedge\vee\perp}$ are the formulas of propositional logic built upon a finite set of atomic types, using the connectives $\rightarrow$, $\vee$, and $\wedge$. The set of atomic types contains the constant $\perp$ that stands for absurdity, and negation is defined in the usual way: $\neg\alpha \equiv \alpha \rightarrow \perp$.

The untyped terms of $\boldsymbol{\lambda\mu}^{\rightarrow\wedge\vee\perp}$ (that we will call $\lambda\mu$-terms, for short) are built upon two disjoint alphabets of variables $\mathscr{X}$ and $\mathscr{A}$, according to the following grammar:

$$\begin{aligned}\mathscr{T} \ ::= {} & \mathscr{X} \mid \lambda\mathscr{X}.\,\mathscr{T} \mid \mathscr{T}\,\mathscr{T} \mid \langle\mathscr{T},\mathscr{T}\rangle \mid \pi_1\,\mathscr{T} \mid \pi_2\,\mathscr{T} \mid \\ & \iota_1\,\mathscr{T} \mid \iota_2\,\mathscr{T} \mid \delta(\mathscr{T},\mathscr{X}.\,\mathscr{T},\mathscr{X}.\,\mathscr{T}) \mid \mu\mathscr{A}.\,\mathscr{T} \mid \mathscr{A}\,\mathscr{T}\end{aligned}$$

The elements of $\mathscr{X}$ are called λ-variables, and these of $\mathscr{A}$ are called μ-variables. We let letters from the end of the alphabet (x, y, z) range over $\mathscr{X}$, and letters from the beginning of the alphabet (a, b, c) range over $\mathscr{A}$. δ and μ are binding operators. Any free occurrence of x in N and any free occurrence of y in O is bound in the term $\delta(M, x.\,N, y.\,O)$. Similarly, any free occurrence of a in M is bound in $\mu a.\,M$. We consider that some implicit convention (e.g., [3, p. 26]) prevents clashes between free and bound variables. We write "=", without subscript, for the relation of α-conversion, and we let $M[x{:=}N]$ denote the usual capture-avoiding substitution.

We define an antecedent to be a set of declarations of the form $x{:}\alpha$ where x is a λ-variable, α is a type, and where all the declared variables are distinct. Similarly, we define a succedent to be a set of declarations of μ-variables obeying the same constraints. The typing system of $\boldsymbol{\lambda\mu}^{\rightarrow\wedge\vee\perp}$ is given by means of sequents of the form $\Gamma \vdash M : \alpha\,;\ \Delta$, where Γ is an antecedent, M is a $\lambda\mu$-term, α is a type, and Δ is a succedent. The typing rules are the following:

$$x : \alpha, \Gamma \vdash x : \alpha\,;\ \Delta$$

$$\frac{x : \alpha, \Gamma \vdash M : \beta\,;\ \Delta}{\Gamma \vdash \lambda x.\,M : \alpha \to \beta\,;\ \Delta}\ (\to\text{-I}) \qquad \frac{\Gamma \vdash M : \alpha \to \beta\,;\ \Delta \qquad \Gamma \vdash N : \alpha\,;\ \Delta}{\Gamma \vdash M\,N : \beta\,;\ \Delta}\ (\to\text{-E})$$

$$\frac{\Gamma \vdash M : \alpha\,;\ \Delta \qquad \Gamma \vdash N : \beta\,;\ \Delta}{\Gamma \vdash \langle M, N\rangle : \alpha \wedge \beta\,;\ \Delta}\ (\wedge\text{-I})$$

$$\frac{\Gamma \vdash M : \alpha \wedge \beta\,;\ \Delta}{\Gamma \vdash \pi_1\,M : \alpha\,;\ \Delta}\ (\wedge\text{-E1}) \qquad \frac{\Gamma \vdash M : \alpha \wedge \beta\,;\ \Delta}{\Gamma \vdash \pi_2\,M : \beta\,;\ \Delta}\ (\wedge\text{-E2})$$

$$\frac{\Gamma \vdash M : \alpha\,;\ \Delta}{\Gamma \vdash \iota_1\,M : \alpha \vee \beta\,;\ \Delta}\ (\vee\text{-I1}) \qquad \frac{\Gamma \vdash M : \beta\,;\ \Delta}{\Gamma \vdash \iota_2\,M : \alpha \vee \beta\,;\ \Delta}\ (\vee\text{-I2})$$

$$\frac{\Gamma \vdash M : \alpha \vee \beta\,;\ \Delta \qquad x : \alpha, \Gamma \vdash N : \gamma\,;\ \Delta \qquad y : \beta, \Gamma \vdash O : \gamma\,;\ \Delta}{\Gamma \vdash \delta(M, x.\,N, y.\,O) : \gamma\,;\ \Delta}\ (\vee\text{-E})$$

$$\frac{\Gamma \vdash M : \bot\,;\ \Delta, a : \alpha}{\Gamma \vdash \mu a.\,M : \alpha\,;\ \Delta}\ (\textsc{muabs}) \qquad \frac{\Gamma \vdash M : \alpha\,;\ \Delta, a : \alpha}{\Gamma \vdash a\,M : \bot\,;\ \Delta, a : \alpha}\ (\textsc{name})$$

The one-step reduction relation of $\boldsymbol{\lambda\mu}^{\to\wedge\vee\bot}$ is defined as the union of three different one-step reduction relations: the relation of detour-reduction ($\to_D$), which corresponds to the usual detour conversions of intuitionistic logic; the relation of δ-reduction ($\to_\delta$), which corresponds to the permutative conversions related to the elimination rule of disjunction; the relation of μ-reduction ($\to_\mu$), which corresponds to conversions that are proper to classical logic. Following Barendregt [3], we write "$\xrightarrow{+}_X$" and "$\twoheadrightarrow_X$" to denote, repectively, the transitive closure and the transitive, reflexive closure of a reduction relation "$\to_X$".

Definition 1. *(detour-reduction)*

(a) $(\lambda x.\, M)\, N \to_D M[x{:=}N]$
(b) $\pi_1\, \langle M, N\rangle \to_D M$
(c) $\pi_2\, \langle M, N\rangle \to_D N$
(d) $\delta(\iota_1\, M, x.\, N, y.\, O) \to_D N[x{:=}M]$
(e) $\delta(\iota_2\, M, x.\, N, y.\, O) \to_D O[y{:=}M]$ ∎

Definition 2. *(δ-reduction)*

(a) $\delta(M, x.\, N, y.\, O)\, P \to_\delta \delta(M, x.\, N\, P, y.\, O\, P)$
(b) $\pi_1\, \delta(M, x.\, N, y.\, O) \to_\delta \delta(M, x.\, \pi_1\, N, y.\, \pi_1\, O)$
(c) $\pi_2\, \delta(M, x.\, N, y.\, O) \to_\delta \delta(M, x.\, \pi_2\, N, y.\, \pi_2\, O)$
(d) $\delta(\delta(M, x.\, N, y.\, O), u.\, P, v.\, Q) \to_\delta \delta(M, x.\, \delta(N, u.\, P, v.\, Q), y.\, \delta(O, u.\, P, v.\, Q))$
∎

In order to define the different basic μ-reduction steps, we must first introduce a notion of structural substitution. Let $C[\,]$ be a context, (i.e., a $\lambda\mu$-term with a hole). The structural substitution $M[a * := C[*]]$ is inductively defined as follows:

$x^* = x$
$(\lambda x.\, M)^* = \lambda x.\, (M^*)$
$(M\, N)^* = M^*\, N^*$
$\langle M, N\rangle^* = \langle M^*, N^*\rangle$
$(\pi_i\, M)^* = \pi_i\, (M^*)$
$(\iota_i\, M)^* = \iota_i\, (M^*)$
$\delta(M, x.\, N, y.\, O)^* = \delta(M^*, x.\, N^*, y.\, O^*)$
$(\mu b.\, M)^* = \mu b.\, (M^*)$ if $a \neq b$
$(\mu a.\, M)^* = \mu a.\, M$
$(b\, M)^* = b\, (M^*)$ if $a \neq b$
$(a\, M)^* = C[M^*]$

where M^* stands for $M[a * := C[*]]$.

Definition 3. *(μ-reduction)*

(a) $(\mu a.\, M)\, N \to_\mu \mu a.\, M[a * := a\, (*\, N)]$
(b) $\pi_1\, (\mu a.\, M) \to_\mu \mu a.\, M[a * := a\, (\pi_1\, *)]$
(c) $\pi_2\, (\mu a.\, M) \to_\mu \mu a.\, M[a * := a\, (\pi_2\, *)]$
(d) $\delta(\mu a.\, M, x.\, N, y.\, O) \to_\mu \mu a.\, M[a * := a\, \delta(*, x.\, N, y.\, O)]$

where $a \in \mathrm{FV}(M)$. ■

In the above definition, we stipulate that a must occur free in M. Nevertheless, the above reductions also make sense when the μ-abstraction is vacuous. In this case, they correspond to the $\bot$-reductions of intuitionistic logic [31], which are also needed for the subformula property. These $\bot$-reductions may therefore be seen as particular cases of μ-reductions. However, for technical reasons, we prefer to keep them separate.

Definition 4. *($\bot$-reduction)*

(a) $(\mu a.\, M)\, N \to_{\bot} \mu a.\, M$
(b) $\pi_1\, (\mu a.\, M) \to_{\bot} \mu a.\, M$
(c) $\pi_2\, (\mu a.\, M) \to_{\bot} \mu a.\, M$
(d) $\delta(\mu a.\, M, x.\, N, y.\, O) \to_{\bot} \mu a.\, M$

where $a \notin \mathrm{FV}(M)$. ■

The next proposition ensures that the above relations of reduction, which are defined at the untyped level, correspond indeed to proof-theoretic conversions.

Proposition 1. (Subject reduction) *Let M, N, α, Γ, and Δ be such that $\Gamma \vdash M : \alpha\, ;\, \Delta$ and $M \to_{D\delta\mu\bot} N$. Then $\Gamma \vdash N : \alpha\, ;\, \Delta$.* □

We end this section by giving a characterization of the normal terms, from which we derive the subformula property. Consider the following grammar:

$$\begin{aligned}\mathscr{P} &::= \lambda \mathscr{X}.\, \mathscr{P} \mid \langle \mathscr{P}, \mathscr{P} \rangle \mid \iota_1\, \mathscr{P} \mid \iota_2\, \mathscr{P} \mid \delta(\mathscr{Q}, \mathscr{X}.\, \mathscr{P}, \mathscr{X}.\, \mathscr{P}) \mid \mu \mathscr{A}.\, \mathscr{P} \mid \mathscr{Q}\\ \mathscr{Q} &::= \mathscr{X} \mid \mathscr{Q}\, \mathscr{P} \mid \pi_1\, \mathscr{Q} \mid \pi_2\, \mathscr{Q} \mid \mathscr{A}\, \mathscr{P}\end{aligned}$$

we say that a $\lambda\mu$-term that conforms to $\mathscr{P}$ is P-canonical (or simply, canonical). Similarly, a $\lambda\mu$-term that conforms to $\mathscr{Q}$ is said to be Q-canonical.

Lemma 1. *Let M, α, Γ, and Δ be such that*

$$\Gamma \vdash M : \alpha\, ;\, \Delta \qquad (*)$$

and let Π be the derivation of $(*)$.

(a) *If M is Q-canonical then every type occurring in Π is either $\bot$, or a subformula of a type occurring in Γ or Δ.*
(b) *If M is P-canonical then every type occurring in Π is either $\bot$, or a subformula of a type occurring in Γ or Δ, or a subformula of α.* □

Lemma 2. *Let M, α, Γ, and Δ be such that $\Gamma \vdash M : \alpha\, ;\, \Delta$. If M is $D\delta\mu\bot$-normal then M is canonical.* □

We get the subformula property as a direct consequence of these two lemmas.

Proposition 2. *Let M, α, Γ, and Δ be such that*

$$\Gamma \vdash M : \alpha\, ;\, \Delta \qquad (*)$$

If M is $D\delta\mu\bot$-normal then every type occurring in the derivation of $(*)$ is either $\bot$, or a subformula of a type occurring in Γ or Δ, or a subformula of α. □

3 Strong normalisation of the structural reductions

In this section, we prove that the untyped $\lambda\mu$-terms are strongly normalizable with respect to the reduction relation induced by the structural reduction steps (i.e., δ, μ, and $\bot$). To this end, we provide the $\lambda\mu$-terms with a norm that strictly decreases under the relation of $\delta\mu\bot$-reduction. This norm is adapted from the norm that we introduced in [7]. Nevertheless it is more involved because we have to accommodate the structural substitutions of the μ-reductions.

Definition 5. *The norm $|\cdot|$ assigned to the $\lambda\mu$-terms is inductively defined as follows:*

(a) $|x| = 1$;
(b) $|\lambda x.\, M| = |M|$
(c) $|M\,N| = |M| + \#M \times |N|$
(d) $|\langle M, N\rangle| = |M| + |N|$
(e) $|\pi_1\, M| = |M| + \#M$
(f) $|\pi_2\, M| = |M| + \#M$
(g) $|\iota_1\, M| = |M|$
(h) $|\iota_2\, M| = |M|$
(i) $|\delta(M, x.\, N, y.\, O)| = |M| + \#M \times (|N| + |O|)$
(j) $|\mu a.\, M| = |M|$
(k) $|a\, M| = |M|$

where:

(a) $\#x = 1$;
(b) $\#\lambda x.\, M = 1$
(c) $\#M\,N = \#M$
(d) $\#\langle M, N\rangle = 1$
(e) $\#\pi_1\, M = \#M$
(f) $\#\pi_2\, M = \#M$
(g) $\#\iota_1\, M = 1$
(h) $\#\iota_2\, M = 1$
(i) $\#\delta(M, x.\, N, y.\, O) = 2 \times \#M \times (\#N + \#O)$
(j) $\#\mu a.\, M = \lfloor M\rfloor_a + 1$
(k) $\#a\, M = 1$

and where:

(a) $\lfloor x\rfloor_a = 0$;
(b) $\lfloor \lambda x.\, M\rfloor_a = \lfloor M\rfloor_a$
(c) $\lfloor M\,N\rfloor_a = \lfloor M\rfloor_a + \#M \times \lfloor N\rfloor_a$
(d) $\lfloor \langle M, N\rangle\rfloor_a = \lfloor M\rfloor_a + \lfloor N\rfloor_a$
(e) $\lfloor \pi_1\, M\rfloor_a = \lfloor M\rfloor_a$
(f) $\lfloor \pi_2\, M\rfloor_a = \lfloor M\rfloor_a$
(g) $\lfloor \iota_1\, M\rfloor_a = \lfloor M\rfloor_a$
(h) $\lfloor \iota_2\, M\rfloor_a = \lfloor M\rfloor_a$

(i) $\lfloor \delta(M, x.\, N, y.\, O) \rfloor_a = \lfloor M \rfloor_a + \#M \times (\lfloor N \rfloor_a + \lfloor O \rfloor_a)$
(j) $\lfloor \mu b.\, M \rfloor_a = \lfloor M \rfloor_a$
(k) $\lfloor a\, M \rfloor_a = \lfloor M \rfloor_a + \#M$
(l) $\lfloor b\, M \rfloor_a = \lfloor M \rfloor_a$ ■

This norm is strictly positive and compatible with the term formation rules.

Lemma 3. *Let $C[\,]$ be any context.*

(a) *If $\#M \geq \#N$ and $|M| > |N|$ then $|C[M]| > |C[N]|$;*
(b) *If $\#M \geq \#N$ and $\lfloor M \rfloor_a \geq \lfloor N \rfloor_a$ then $\#C[M] \geq \#C[N]$ and $\lfloor C[M] \rfloor_a \geq \lfloor C[N] \rfloor_a$.* □

The next proposition may be easily established by adapting the proof given in [7].

Proposition 3. *If $S \to_\delta T$ then $|S| > |T|$, for any $\lambda\mu$-terms S, T.* □

We end this section by sketching the proof that the norm of Definition 5 decreases under the relation of $\mu\bot$-reduction.

Lemma 4. *Let M, N be $\lambda\mu$-terms and let $M^* \equiv M[a\,{*}{:=}a\,({*}\,N)]$. If $a \notin \mathrm{FV}(N)$ then:*

(a) $\lfloor M \rfloor_a = \lfloor M^* \rfloor_a$;
(b) $\lfloor M \rfloor_b + \lfloor M \rfloor_a \times \lfloor N \rfloor_b = \lfloor M^* \rfloor_b$;
(c) $|M| + \lfloor M \rfloor_a \times |N| = |M^*|$. □

Lemma 5. *Let S, T be $\lambda\mu$-terms. If $S \to_{\mu\bot} T$ then:*

(a) $\#S \geq \#T$;
(b) $\lfloor S \rfloor_b \geq \lfloor T \rfloor_b$.

Proof. We establish the property as a consequence of Lemma 3 (b), by proving Inequations (a) and (b) for each basic reduction step. We give only the first case, and leave the other ones to the reader.

(a)
$$\begin{aligned} \#S &= \#(\mu a.\, M)\, N \\ &= \#\mu a.\, M \\ &= \lfloor M \rfloor_a + 1 \\ &= \lfloor M[a\,{*}{:=}a\,({*}\,N)] \rfloor_a + 1 \qquad \text{\textit{by Lemma 4 (a)}} \\ &= \#\mu a.\, M[a\,{*}{:=}a\,({*}\,N)] \\ &= \#T \end{aligned}$$

(b) $$\begin{aligned}\lfloor S \rfloor_b &= \lfloor (\mu a.\, M)\, N \rfloor_b \\ &= \lfloor \mu a.\, M \rfloor_b + \#\mu a.\, M \times \lfloor N \rfloor_b \\ &= \lfloor M \rfloor_b + (\lfloor M \rfloor_a + 1) \times \lfloor N \rfloor_b \\ &\geq \lfloor M \rfloor_b + \lfloor M \rfloor_a \times \lfloor N \rfloor_b \\ &= \lfloor M[a *{:=} a\,(* N)] \rfloor_b \qquad \textit{by Lemma 4 (b)} \\ &= \lfloor \mu a.\, M[a *{:=} a\,(* N)] \rfloor_b \\ &= \lfloor T \rfloor_b\end{aligned}$$

□

Proposition 4. *If $S \to_{\mu\perp} T$ then $|S| > |T|$, for any $\lambda\mu$-term S, T.*

Proof. We establish the property as a consequence of Lemma 3 (a) and Lemma 5, by proving it for each basic reduction step. We give only the first case, and leave the other ones to the reader.

$$\begin{aligned}|S| &= |(\mu a.\, M)\, N| \\ &= |\mu a.\, M| + \#\mu a.\, M \times |N| \\ &= |M| + (\lfloor M \rfloor_a + 1) \times |N| \\ &> |M| + \lfloor M \rfloor_a \times |N| \\ &= |M[a *{:=} a\,(* N)]| \qquad \textit{by Lemma 4 (c)} \\ &= |\mu a.\, M[a *{:=} a\,(* N)]| \\ &= |T|\end{aligned}$$

□

As a direct consequence of Propositions 3 and 4, we obtain that any $\lambda\mu$-term is strongly $\delta\mu\perp$-normalizable.

4 Postponement of the ⊥-reductions

All the right-hand sides of the rules of Definition 4 are of the same form: $\mu a.\, M$, where $a \notin \mathrm{FV}(M)$. One easily checks that there is no critical pair between this form and any left-hand side of the rules of Definitions 1, 2 and 3. Consequently, the following proposition may be established easily.

Proposition 5. *Let $R \in \{D, \delta, \mu\}$, and let L, M and N be three $\lambda\mu$-terms. If $L \to_\perp M \to_R N$ then there exists a $\lambda\mu$-term O such that $L \overset{+}{\to}_R O \twoheadrightarrow_\perp N$.* □

As a consequence of this proposition together with the strong normalization of the ⊥-reductions, we have that any infinite sequence of $D\delta\mu\perp$-reduction steps may be turned into an infinite sequence of $D\delta\mu$-reduction steps.

5 Negative translation and CPS-simulation

In this section, we adapt to $\boldsymbol{\lambda\mu}^{\to\wedge\vee\bot}$ the negative translation and the CPS-simulation given in [7].

Definition 6. *The negative translation* $\overline{\alpha}$ *of any type* α *is defined as* $\overline{\alpha} = \sim\sim\alpha^\circ$ *where* $\sim\alpha = \alpha \to o$ *for some distinguished atomic type* o *(that is not used elsewhere), and where:*

(a) $\bot^\circ = o$
(b) $a^\circ = a$
(c) $(\alpha \to \beta)^\circ = \overline{\alpha} \to \overline{\beta}$
(d) $(\alpha \wedge \beta)^\circ = \sim(\overline{\alpha} \to \sim\overline{\beta})$
(e) $(\alpha \vee \beta)^\circ = \sim\overline{\alpha} \to \sim\sim\overline{\beta}$ ■

Definition 7. *The CPS-translation* $\overline{M}$ *of any* $\lambda\mu$*-term* M *is inductively defined as follows:*

(a) $\overline{x} = \lambda k.\, x\, k$
(b) $\overline{\lambda x.\, M} = \lambda k.\, k\,(\lambda x.\, \overline{M})$
(c) $\overline{(M\, N)} = \lambda k.\, \overline{M}\,(\lambda m.\, m\, \overline{N}\, k)$
(d) $\overline{\langle M, N\rangle} = \lambda k.\, k\,(\lambda p.\, p\, \overline{M}\, \overline{N})$
(e) $\overline{\pi_1\, M} = \lambda k.\, \overline{M}\,(\lambda p.\, p\,(\lambda i.\, \lambda j.\, i\, k))$
(f) $\overline{\pi_2\, M} = \lambda k.\, \overline{M}\,(\lambda p.\, p\,(\lambda i.\, \lambda j.\, j\, k))$
(g) $\overline{\iota_1\, M} = \lambda k.\, k\,(\lambda i.\, \lambda j.\, i\, \overline{M})$
(h) $\overline{\iota_2\, M} = \lambda k.\, k\,(\lambda i.\, \lambda j.\, j\, \overline{M})$
(i) $\overline{\delta(M, x.\, N, y.\, O)} = \lambda k.\, \overline{M}\,(\lambda m.\, m\,(\lambda x.\, \overline{N}\, k)\,(\lambda y.\, \overline{O}\, k))$
(j) $\overline{\mu a.\, M} = \lambda a.\, \overline{M}\,(\lambda k.\, k)$
(k) $\overline{a\, M} = \lambda k.\, \overline{M}\, a$

where k, m, p, i *and* j *are fresh variables.* ■

The next proposition states that these two translations commute with the typing relation.

Proposition 6. *Let* M, α, Γ, *and* Δ *be such that* $\Gamma \vdash M : \alpha\, ;\, \Delta$ *Then* $\overline{M}$ *is a* λ*-term of the simply typed* λ*-calculus, typable with type* $\overline{\alpha}$ *under the set of declarations* $\overline{\Gamma}, \sim\Delta^\circ$. □

The translation of Definition 7 does not allow the detour-reduction steps to be simulated by β-reduction. This is due to the so-called administrative redexes [22]. In order to circumvent this problem, we introduce the following modified translation.

Definition 8. *The modified CPS-translation* $\overline{\overline{M}}$ *of any* $\lambda\mu$*-term* M *is defined as:*

$$\overline{\overline{M}} = \lambda k.\, (M : k)$$

where k *is a fresh variable, and where the infix operator ":" obeys the following definition:*

(a) $x : K = x\,K$
(b) $\lambda x.\, M : K = K\,(\lambda x.\, \overline{\overline{M}})$
(c) $(M\,N) : K = M : \lambda m.\, m\,\overline{\overline{N}}\,K$
(d) $\langle M, N\rangle : K = K\,(\lambda p.\, p\,\overline{\overline{M}}\,\overline{\overline{N}})$
(e) $\pi_1\,M : K = M : \lambda p.\, p\,(\lambda i.\,\lambda j.\, i\,K)$
(f) $\pi_2\,M : K = M : \lambda p.\, p\,(\lambda i.\,\lambda j.\, j\,K)$
(g) $\iota_1\,M : K = K\,(\lambda i.\,\lambda j.\, i\,\overline{\overline{M}})$
(h) $\iota_2\,M : K = K\,(\lambda i.\,\lambda j.\, j\,\overline{\overline{M}})$
(i) $\delta(M, x.\,N, y.\,O) : K = M : \lambda m.\, m\,(\lambda x.\,(N : K))\,(\lambda y.\,(O : K))$
(j) $\mu a.\, M : K = (M : \lambda k.\, k)\,[a{:=}K]$ *if* $a \in \mathrm{FV}(M)$
(k) $\mu a.\, M : K = (\lambda a.\,(M : \lambda k.\, k))\,K$ *if* $a \notin \mathrm{FV}(M)$
(l) $a\,M : K = M : a$

where m, p, i and j are fresh variables. ■

This modified CPS-translation is consistent with the translation of Definition 7 in the sense of the following lemma.

Lemma 6. *Let M be a $\lambda\mu$-term. Then:*

(a) $\overline{M} \twoheadrightarrow_\beta \overline{\overline{M}}$,
(b) $\overline{M}\,K \twoheadrightarrow_\beta M : K$, *for any λ-term K.* □

As a consequence of this lemma, we obtain the following proposition.

Proposition 7. *Let M, α, Γ, and Δ be such that $\Gamma \vdash M : \alpha\,;\ \Delta$. Then $\overline{\overline{M}}$ is a λ-term of the simply typed λ-calculus, typable with type $\overline{\alpha}$ under the set of declarations $\overline{\Gamma}, {\sim}\Delta^\circ$.* □

We now prove that the modified CPS-translation of Definiton 8 simulates the relation of detour-reduction by strict β-reduction, and the relations of δ- and μ-reduction by equality. We first state a few technical lemmas.

Lemma 7. *Let M and N be $\lambda\mu$-terms and K be a simple λ-term such that $x \notin \mathrm{FV}(K)$. Then $(M : K)[x{:=}\overline{\overline{N}}] \twoheadrightarrow_\beta (M[x{:=}N]) : K$.* □

Lemma 8. *Let M, N, and O be $\lambda\mu$-terms, and K be a simple λ-term such that $a \notin \mathrm{FV}(K)$. Then:*

(a) $(M : K)[a{:=}\lambda m.\, m\,\overline{\overline{N}}\,a] \twoheadrightarrow_\beta (M[a\,{*}{:=}\,a\,({*}\,N)]) : K$,
(b) $(M : K)[a{:=}\lambda p.\, p\,(\lambda i.\,\lambda j.\, i\,a)] \twoheadrightarrow_\beta (M[a\,{*}{:=}\,a\,(\pi_1\,{*})]) : K$,
(c) $(M : K)[a{:=}\lambda p.\, p\,(\lambda i.\,\lambda j.\, j\,a)] \twoheadrightarrow_\beta (M[a\,{*}{:=}\,a\,(\pi_2\,{*})]) : K$,
(d) $(M : K)[a{:=}\lambda m.\, m\,(\lambda x.\,(N : a))\,(\lambda y.\,(O : a))]$
$\twoheadrightarrow_\beta (M[a\,{*}{:=}\,a\,\delta({*}, x.\,N, y.\,O)]) : K.$ □

Lemma 9. *Let M and N be two $\lambda\mu$-terms and let $C[\,]$ be any context. Then, for any simple λ-term K:*

(a) *if* $M : K \xrightarrow{+}_{\beta} N : K$ *then* $C[M] : K \xrightarrow{+}_{\beta} C[N] : K$,
(b) *if* $M : K = N : K$ *then* $C[M] : K = C[N] : K$. □

The next two lemmas concern the simulation of the relations of detour- and δ-reduction. Their proofs may be found in [7]

Lemma 10. *Let S and T be two $\lambda\mu$-terms. If $S \to_D T$ then $\overline{\overline{S}} \xrightarrow{+}_{\beta} \overline{\overline{T}}$.* □

Lemma 11. *Let S and T be two $\lambda\mu$-terms. If $S \to_\delta T$ then $\overline{\overline{S}} = \overline{\overline{T}}$.* □

It remains to prove that the μ-reductions are interpreted as equality.

Lemma 12. *Let S and T be two $\lambda\mu$-terms such that $S \to_\mu T$. Then $\overline{\overline{S}} = \overline{\overline{T}}$.*

Proof. *We prove that, for any simple λ-term K,*

$$S : K = T : K \qquad (*)$$

from which the property follows.

Equation $(*)$ may be established as a consequence of Lemma 9 (b) by proving that it holds for each basic reduction step. We give only the first case, and leave the other ones to the reader.

$$\begin{aligned}
S : K &= (\mu a.\, M)\, N : K \\
&= \mu a.\, M : \lambda m.\, m\, \overline{\overline{N}}\, K \\
&= (M : \lambda k.\, k)[a{:=}\lambda m.\, m\, \overline{\overline{N}}\, K] \\
&= (M : \lambda k.\, k)[a{:=}\lambda m.\, m\, \overline{\overline{N}}\, a][a{:=}K] \\
&= (M[a * := a\,(* N)] : \lambda k.\, k)[a{:=}K] \qquad \text{by Lemma 8 (a)} \\
&= \mu a.\, M[a * := a\,(* N)] : K \\
&= T
\end{aligned}$$

□

6 Strong normalization

We are now in a position of proving the main proposition of this paper.

Proposition 8. (Strong Normalization) *Any well-typed $\lambda\mu$-term of $\boldsymbol{\lambda\mu}^{\to\wedge\vee\bot}$ is strongly normalizable with respect to the relation of $D\delta\mu\bot$-reduction.*

Proof. Suppose it is not the case. Then, by Proposition 4 and 5, there would exist an infinite sequence of D- and $\delta\mu$-reduction steps starting from a typable term (say, M) of $\boldsymbol{\lambda\mu}^{\to\wedge\vee\bot}$. If this infinite sequence contains infinitely many D-reduction steps, there must exist, by Lemmas 10, 11 and 12 an infinite sequence of β-reduction steps starting from $\overline{\overline{M}}$. But this, by Proposition 7, would contradict the strong normalization of the simply typed λ-calculus. Hence the infinite sequence may contain only a finite number of D-reduction steps. But then, it would contain an infinite sequence of consecutive $\delta\mu$-reduction steps, which is impossible by Propositions 3 and 4. □

7 Confluence of the reductions

We prove the Church-Rosser property by establishing the local confluence of the reductions.

Lemma 13. *Let M, N, O be $\lambda\mu$-terms such that $M \rightarrow_{D\delta\mu} N$ and $M \rightarrow_{D\delta\mu} O$ then there exists a $\lambda\mu$-term P such that $N \twoheadrightarrow_{D\delta\mu} P$ and $M \twoheadrightarrow_{D\delta\mu} P$.* □

Proposition 9. (Church-Rosser Property) *Let M, N, O be typable $\lambda\mu$-terms such that $M \twoheadrightarrow_{D\delta\mu} N$ and $M \twoheadrightarrow_{D\delta\mu} O$ then there exists a $\lambda\mu$-term P such that $N \twoheadrightarrow_{D\delta\mu} P$ and $M \twoheadrightarrow_{D\delta\mu} P$.*

Proof. A consequence of Proposition 8, Lemma 13, and Newman lemma. □

References

1. F. Barbanera and S. Berardi. Extracting constructive content from classical logic via control-like reductions. In M. Bezem and J.F. Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications, TLCA'93*, volume 664 of *Lecture Notes in Computer Science*, pages 45–59. Springer Verlag, 1993.
2. F. Barbanera and S. Berardi. A symmetric lambda-calculus for "classical" program extraction. In M. Hagiya and J.C. Mitchell, editors, *Proceedings of the International Symposium on Theoretical Aspects of Computer Software*, volume 789 of *Lecture Notes in Computer Science*, pages 494–515. Springer Verlag, 1994.
3. H.P. Barendregt. *The lambda calculus, its syntax and semantics.* North-Holland, revised edition, 1984.
4. R. Constable and C. Murthy. Finding computational content in classical proofs. In G. Huet and G. Plotkin, editors, *Logical Frameworks*, pages 341–362. Cambridge University Press, 1991.
5. Ph. de Groote. A CPS-translation of the $\lambda\mu$-calculus. In S. Tison, editor, *19th International Colloquium on Trees in Algebra and Programming, CAAP'94*, volume 787 of *Lecture Notes in Computer Science*, pages 85–99. Springer Verlag, 1994.
6. Ph. de Groote. A simple calculus of exception handling. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Second International Conference on Typed Lambda Calculi and Applications, TLCA'95*, volume 902 of *Lecture Notes in Computer Science*, pages 201–215. Springer Verlag, 1995.
7. Ph. de Groote. On the strong normalisation of natural deduction with permutation-conversions. In *10th International Conference on Rewriting Techniques and Applications, RTA'99*, volume 1631 of *Lecture Notes in Computer Science*, pages 45–59. Springer Verlag, 1999.
8. M. Felleisen, D.P. Friedman, E. Kohlbecker, and B. Duba. A syntactic theory of sequential control. *Theoretical Computer Science*, 52:205–237, 1987.
9. M. Felleisen and R. Hieb. The revised report on the syntactic theory of sequential control and state. *Theoretical Computer Science*, 102:235–271, 1992.
10. G. Gentzen. *Recherches sur la déduction logique (Untersuchungen über das logische schliessen).* Presses Universitaires de France, 1955. Traduction et commentaire par R. Feys et J. Ladrière.

11. J.-Y. Girard. A new constructive logic: Classical logic. *Mathematical Structures in Computer Science*, 1:255–296, 1991.
12. J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1989.
13. T. G. Griffin. A formulae-as-types notion of control. In *Conference record of the seventeenth annual ACM symposium on Principles of Programming Languages*, pages 47–58, 1990.
14. W.A. Howard. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *to H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, 1980.
15. J.-L. Krivine. Classical logic, storage operators and second order λ-calculus. *Annals of Pure and Applied Logic*, 68:53–78, 1994.
16. A. Meyer and M. Wand. Continuation semantics in typed lambda-calculi (summary). In R. Parikh, editor, *Logics of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 219–224. Springer Verlag, 1985.
17. C. R. Murthy. An evaluation semantics for classical proofs. In *Proceedings of the sixth annual IEEE symposium on logic in computer science*, pages 96–107, 1991.
18. C. R. Murthy. A computational analysis of Girard's translation and LC. In *Proceedings of the seventh annual IEEE symposium on logic in computer science*, pages 90–101, 1992.
19. M. Parigot. $\lambda\mu$-Calculus: an algorithmic interpretation of classical natural deduction. In A. Voronkov, editor, *Proceedings of the International Conference on Logic Programming and Automated Reasoning*, volume 624 of *Lecture Notes in Artificial Intelligence*, pages 190–201. Springer Verlag, 1992.
20. M. Parigot. Strong normalization for second order classical natural deduction. In *Proceedings of the eighth annual IEEE symposium on logic in computer science*, pages 39–46, 1993.
21. M. Parigot. Proofs of strong normalisation for second order classical natural deduction. *Journal of Symbolic Logic*, 62(4):1461–1479, 1997.
22. G. D. Plotkin. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1:125–159, 1975.
23. D. Prawitz. *Natural Deduction, A Proof-Theoretical Study.* Almqvist & Wiksell, Stockholm, 1965.
24. D. Prawitz. Ideas and results in proof-theory. In J.E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 237–309. North-Holland, 1971.
25. D. Pym and E. Ritter. On the semantics of classical disjunction. *Journal of Pure and Applied Algebra*, To appear.
26. N.J. Rehof and M.H. Sørensen. The λ_Δ-calculus. In M. Hagiya and J.C. Mitchell, editors, *Proceedings of the International Symposium on Theoretical Aspects of Computer Software, TACS'94*, pages 516–542. Lecture Notes in Computer Science, 789, Springer Verlag, 1994.
27. E. Ritter, D. Pym, and L. Wallen. On the intuitionistic force of classical search. *Theoretical Computer Science*, 232:299–333, 2000.
28. E. Ritter, D. Pym, and L. Wallen. Proof-terms for classical and intuitionistic resolution. *Journal of Logic and Computation*, 10(2):173–207, 2000.
29. J. Seldin. On the proof theory of the intermediate logic MH. *Journal of Symbolic Logic*, 51(3):626–647, 1986.
30. G. Stålmarck. Normalization theorems for full first-order classical natural deduction. *Journal of Symbolic Logic*, 56(1):129–149, 1991.
31. A. Troelstra and D. van Dalen. *Constructivism in Mathematics*, volume II. North-Holland, 1988.