Normalization in the $\lambda\mu\mu'$-calculus

Péter Battyányi

Joint work with Karim Nour

26th September 2025, Dubrovnik

Let A, B are arbitrary formulas, then the implicational fragment of the intuitionistic propositional calculus is defined as follows:

$$\frac{}{A \vdash A}\ \text{ax}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \to_i \qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\ \to_e$$

Derivations are written in the form of a proof tree. For example,

$$\dfrac{\dfrac{A, B \vdash A}{A \vdash B \to A} \to_i}{\vdash A \to (B \to A)} \to_i$$

Or,

$$\dfrac{\dfrac{\Gamma \vdash A \to (B \to C) \quad \Gamma \vdash A}{\Gamma \vdash B \to C} \to_e \quad \dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B} \to_e}{\Gamma \vdash C} \to_e ,$$

where $\Gamma = \{A \to (B \to C), A \to B, A\}$.

The λ-terms are defined by the following grammar:

$$\Lambda = \mathcal{V} \mid (\lambda\mathcal{V}.\Lambda) \mid (\Lambda)\Lambda,$$

where $\mathcal{V} = \{x, y, z, \ldots\}$ are the set of variables.

We adopt Krivine's notation, hence $(M)N$ stands for the application of N to M. Moreover, we use the abbreviation $(M_1)M_2 \ldots M_{n-1}M_n$ for the term $(\ldots(M_1)M_2)\ldots)M_{n-1})M_n$. Observe that $(M)NP$ and $(M)(N)P$ denote different terms.

Given an abstraction of the form $\lambda x.M$, we say that the outermost prefix λx binds the free occurrences of x in M. All the variable occurrences that are not bound are said to be free. The set of free variables of M is denoted by $fv(M)$. To avoid the capture of free variables, we accept the Barendregt convention for the term notation: we identify terms that only differ in the names of bound variables.

The types are built from a set $\mathcal{V}_{\mathbb{T}}$ of atomic types with the connective $\rightarrow$. The type formation rules are the following.

$$\mathbb{T} \quad := \quad \mathcal{V}_{\mathbb{T}} \quad | \quad \mathbb{T} \rightarrow \mathbb{T}$$

Introduction

Let Γ denote a (possibly empty) context, that is, a finite set of declarations of the form $x : A$. If Γ is a context, let $|\Gamma| = \{A \mid x : A \in \Gamma \text{ for some } x\}$. The typing rules are as follows.

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

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

The λ-term M is called typable with type A, if $\Gamma \vdash M : A$ is derivable using the rules and axioms.

Let $\Gamma = \{x : A \to (B \to C),\ y : A \to B,\ z : A\}$. Then,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash x : A \to (B \to C)} \quad \dfrac{}{\Gamma \vdash z : A}}{\Gamma \vdash (x)z : B \to C} \quad \dfrac{\dfrac{}{\Gamma \vdash y : A \to B} \quad \dfrac{}{\Gamma \vdash z : A}}{\Gamma \vdash (y)z : B}}{\Gamma \vdash ((x)z)(y)z : C}}{x : A \to (B \to C), y : A \to B \vdash \lambda z.((x)z)(y)z : A \to C}}{x : A \to (B \to C) \vdash \lambda yz.((x)z)(y)z : (A \to B) \to (A \to C)}}{\vdash \lambda xyz.((x)z)(y)z : (A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

When we omit the indications of the terms, we obtain a correct derivation of the end formula.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{|\Gamma| \vdash A \to (B \to C)} \quad \dfrac{}{|\Gamma| \vdash A}}{|\Gamma| \vdash B \to C} \quad \dfrac{\dfrac{}{|\Gamma| \vdash A \to B} \quad \dfrac{}{|\Gamma| \vdash A}}{|\Gamma| \vdash B}}{|\Gamma| \vdash C}}{A \to (B \to C), A \to B \vdash A \to C}}{A \to (B \to C) \vdash (A \to B) \to (A \to C)}}{\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

This phenomenon is referred to as the Curry–Howard isomorphism and was first observed by Curry in the 1930s and (re)discovered by Howard in 1968.
It was widely believed that the Curry–Howard isomorphism is valid only for intuitionistic logic, until Griffin and Murthy discovered in the 1990s that certain terms expressing control can be given the type $\neg\neg A \to A$.

Introduction

From that point on, several calculi realizing the Curry-Howard isomorphism for classical logic have been discovered. For example,

- Rehof and Sorensen's λ_{Δ},
- Curien and Herbelin's $\bar{\lambda}\mu\tilde{\mu}$,
- Krivine's $\lambda\mathcal{C}$,
- Berardi and Barbanera's λ^{Sym},
- Parigot's $\lambda\mu$.

In what follows, we look at Parigot's simply typed $\lambda\mu$-calculus in more detail.

Definitions and notation

- Let $\mathcal{V}_\lambda = \{x, y, z, \dots\}$ denote a set of λ-variables and $\mathcal{V}_\mu = \{\alpha, \beta, \gamma, \dots\}$ denote a set of μ-variables, respectively. The $\lambda\mu$-term formation rules are the following.

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

- In a $\lambda\mu$-term, the λ and μ operators bind the variables in their scope. We consider terms modulo the equivalence relation, which allows to rename the variables bound by a λ- or a μ-abstraction.

Definitions and notation

We are concerned with the $\lambda\mu$-calculus as modified by de Groote. Parigot's original definition was different but had certain shortcomings: although the calculus was strongly normalizing (Parigot), some closed formulas did not admit an assumptionless derivation in the calculus. More importantly, Böhm's theorem did not hold for the calculus (David and Py). These drawbacks were eliminated in de Groote's version (Saurin), at the price of sacrificing strong normalization.

Definitions and notation

The types are built from a set $\mathcal{V}_{\mathbb{T}}$ of atomic types and the constant $\bot$ with the connectives $\neg$ and $\to$. The type formation rules are the following.

$$\mathbb{T} \quad := \quad \mathcal{V}_{\mathbb{T}} \cup \{\bot\} \quad | \quad \neg\mathbb{T} \quad | \quad \mathbb{T} \to \mathbb{T}$$

In the definition below Γ denotes a (possibly empty) context, that is, a finite set of declarations of the form $x : A$ (resp. $\alpha : \neg A$) for a λ-variable x (resp. a μ-variable α) and type A such that a λ-variable x (resp. a μ-variable α) occurs at most once in an expression $x : A$ (resp. $\alpha : \neg A$) of Γ. We also write $\neg A$ for $A \to \bot$.

Definitions and notation

The typing rules are as follows.

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

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

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

Definitions and notation

Below, let $\Gamma = \{x : \neg B \to \neg A,\ y : A,\ z : B,\ \alpha : \neg B\}$ and $\Gamma' = \Gamma \setminus \{z : B\}$. Then

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash z : B}{\Gamma \vdash [\alpha]z : \bot}}{\Gamma' \vdash \lambda z.[\alpha]z : \neg B}}{\Gamma' \vdash (x)\lambda z.[\alpha]z : \neg A \qquad \Gamma' \vdash y : A}}{\Gamma' \vdash ((x)\lambda z.[\alpha]z)y : \bot}}{x : \neg B \to \neg A,\ y : A \vdash \mu\alpha.((x)\lambda z.[\alpha]z)y : B}}{x : \neg B \to \neg A \vdash \lambda y.\mu\alpha.((x)\lambda z.[\alpha]z)y : A \to B}}{\vdash \lambda xy.\mu\alpha.((x)\lambda z.[\alpha]z)y : (\neg B \to \neg A) \to (A \to B)}$$

If we omit the terms annotating the type formulas, we obtain a full fledged proof of $(\neg B \to \neg A) \to (A \to B)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{|\Gamma| \vdash B^{(1)} \qquad |\Gamma| \vdash \neg B^{(2)}}{|\Gamma| \vdash \bot}\,(1)}{|\Gamma'| \vdash \neg B} \qquad |\Gamma'| \vdash \neg B \to \neg A^{(4)}}{|\Gamma'| \vdash \neg A} \qquad |\Gamma'| \vdash A^{(3)}}{\dfrac{\dfrac{\dfrac{|\Gamma'| \vdash \bot}{\neg B \to \neg A, A \vdash B}\,(2)}{\neg B \to \neg A \vdash A \to B}\,(3)}{\vdash (\neg B \to \neg A) \to (A \to B)}\,(4)}$$

Reduction rules

We use several types of substitution in the $\lambda\mu$-calculus.

- β-substitution: $M[x := N]$ is obtained from M by exchanging the free occurrences of x in M with N. Throughout our discussion we assume that bound variables are renamed when necessary.
- μ-substitution: $M[\alpha :=_r N]$ is obtained from M when we replace all subterm occurrences $[\alpha]U$ of M by $[\alpha](U)N$ recursively.
- μ'-substitution: $M[\alpha :=_l N]$ is obtained from M when we replace all subterm occurrences $[\alpha]U$ of M by $[\alpha](N)U$ in a recursive manner.
- ρ-substitution: $M[\alpha := \beta]$ is obtained from M by replacing the free occurrences of α in M by β

The notation M_α stands for the erasure of every occurrence of $[\alpha]$ from M.

Reduction rules

With this in hand, we are able to define the reduction rules.

- β-reduction: $(\lambda x^A.M^B)N^A : B \to_\beta M^B[x := N] : B$
- μ-reduction: $(\mu\alpha^{\neg(A\to B)}.M^\perp)N^A : B \to \mu\alpha^{\neg B}.M^\perp[\alpha :=_r N] : B$
- μ'-reduction: $(N)^{A\to B}\mu\alpha^{\neg A}.M^\perp : B \to \mu\alpha^{\neg B}.M^\perp[\alpha :=_l N] : B$
- ρ-reduction: $[\alpha^{\neg A}]\mu\beta^{\neg A}.M^\perp : \perp \to_\rho M^\perp[\beta := \alpha] : \perp$
- θ-reduction: $\mu\alpha^{\neg A}.[\alpha^{\neg A}]M^A : A \to_\rho M^A : A$ provided $\alpha \notin fv(M)$
- ε-reduction: $\mu\alpha^{\neg A}.\mu\beta^{\neg\perp}.M^\perp : A \to_\varepsilon \mu\alpha^{\neg A}.M^\perp_\beta : A$.

Reduction rules

- β-reduction: As usual, β-reduction eliminates subsequent occurrences of $\rightarrow_i$ and $\rightarrow_e$.

$$\dfrac{\dfrac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \rightarrow B}\ \rightarrow_i \qquad \Gamma \vdash N : A}{\Gamma \vdash (\lambda x.M)N : B}\ \rightarrow_e \rightsquigarrow \Gamma \vdash M[x := N] : B$$

  where the derivation of $\Gamma \vdash M[x := N] : B$ is obtained from that of $\Gamma \vdash M : B$ when we exchange all hypotheses $x : A$ with the derivation $\Gamma \vdash N : A$.

Reduction rules

- μ-reduction: Given the declaration $\alpha : A \to B$, we have $M : \bot$. To produce the derivation connected with the contractum of a μ-reduction, it suffices to start with the declaration $\alpha : B$ and to replace in M each subterm of the form $[\alpha]P$ by $[\alpha](P)N$ to obtain a term of type B. At the proof level, μ-reduction corresponds to the simplification of a reasoning by contradiction, when an application of a $\bot$-elimination is replaced by $\bot$-eliminations of lower type complexity.

Reduction rules

$$\dfrac{\dfrac{\Gamma, \alpha : \neg(A \to B) \vdash M : \bot}{\Gamma \vdash \mu\alpha.M : A \to B}\ \bot_e \qquad \Gamma \vdash N : A}{\Gamma \vdash (\mu\alpha.M)N : B}\ \to_e \quad \rightsquigarrow \quad \Gamma \vdash \mu\alpha.M[\alpha :=_r N] : B$$

Locally,

$$[\alpha^{\neg(A\to B)}]U^{A\to B} \rightsquigarrow [\alpha^{\neg B}](U^{A\to B})N^{A}$$

Reduction rules

- μ'-reduction: It is the symmetric counterpart of rule μ. Given the declaration $\alpha : \neg A$, we have $\mu\alpha.M : A$ by an application of $\bot_e$. If we also have $N : A \to B$, then $(N)\mu\alpha.M : B$ and $(N)\mu\alpha.M : B \to \mu\alpha.M[\alpha :=_l N]$. To obtain the effect of the rule, it suffices to start with the declaration $\alpha : B$ and to replace in M each subterm of the form $[\alpha]P$ by $[\alpha](N)P$. Here, we observe that there is no direct relation between the two types associated with α before and after the reduction.

Reduction rules

$$\dfrac{\Gamma \vdash N : A \to B \qquad \dfrac{\Gamma, \alpha : \neg A \vdash M : \bot}{\Gamma \vdash \mu\alpha.M : A}\,\bot_e}{\Gamma \vdash (N)\mu\alpha.M : B}\,\to_e \rightsquigarrow \Gamma \vdash \mu\alpha.M[\alpha :=_l N] : B$$

Locally,

$$[\alpha^{\neg A}]U^A \rightsquigarrow [\alpha^{\neg B}](N^{A\to B})U^A$$

Reduction rules

- ρ-reduction: If we obtained a derivation of $[\beta]\mu\alpha.M : \bot$, then, necessarily, $\alpha, \beta : \neg A$ for some A. Hence, ρ-reduction can be interpreted as a simplification of proofs: it is enough to replace every occurrence of the assumption $\alpha : \neg A$ by the same type of assumption $\beta : \neg A$ throughout the derivation.
- θ-reduction: Given the declaration $\alpha : \neg A$, we have $M : A$, hence $\mu\alpha.[\alpha]M : A$. This term reduces simply to M, provided that α does not appear free in M.
- ε-reduction: Given the declaration $\alpha : \neg A$ and the relation $\mu\alpha.\mu\beta.M : A$, we necessarily have $\beta : \neg\bot$. We eliminate $\mu\beta$ as well as all occurrences of $[\beta]$ in M to obtain a term of type A. Hence, ε-rule removes a proof of $\bot$ obtained from $\bot$ by contradiction.

Reduction rules

In second order propositional λ-calculus, the Church numerals $\lambda x \lambda f.(f)^n x$ have type $\mathcal{N} = X \to (X \to X) \to X$. The following statement is true: A (closed) term t is of type $\mathcal{N}$ iff $t = \lambda x \lambda f.(f)^n x$ for some n, where $(f)^0 x = x$ and $(f)^{n+1} x = (f)(f)^n x$.

This is no longer true in the $\lambda\mu$-calculus. For example,

$$\lambda x \lambda f.\mu\alpha[\alpha](f)\mu\beta[\alpha](f)\mu\gamma[\beta](f)(f)\mu\beta[\gamma](f)\mu\gamma[\beta]x$$

is of type $\mathcal{N}$ (and has value 3).

Reduction rules

The following is true:

Theorem (B; Nour 2022)

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

In particular, to extract the value of a $\lambda\mu$-term of type $\mathcal{N}$, we have make use of all the reductions $\beta\mu\mu'\rho\theta\varepsilon$.
We will state that $\beta\mu\mu'\rho\theta\varepsilon$-reduction is weakly normalizing, thus the assumption $((M)x)f \in \mathcal{WN}_{\beta\mu\mu'\rho\theta\varepsilon}$ can in fact be omitted in the above corollary.

Example

Let $M = \lambda x.\mu\alpha.[\alpha]\lambda g.(g)\mu\beta.\mu\gamma.[\alpha]\lambda f.(f)x$. Let $x : X$, $f : X \to X$, $g : X \to X$, $\alpha : \neg((X \to X) \to X)$, $\beta : \neg X$ and $\gamma : \neg\bot$, then $\vdash M : \mathcal{N}$. Furthermore,

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

Let us consider $\beta\mu\rho\tau\varepsilon$-reduction, that is, we omit μ'. Let $\to_{\lambda\mu}$ denote the union of these reductions.

Theorem (Parigot 1992; Py 1998)

The $\lambda\mu$-calculus without the rule μ' is confluent. Namely, if M, M_1, M_2 are terms such that

$$M \to^*_{\lambda\mu} M_1 \ and \ M \to^*_{\lambda\mu} M_2,$$

then there exists N such that

$$M_1 \to^*_{\lambda\mu} N \ and \ M_2 \to^*_{\lambda\mu} N.$$

Proof-theoretical properties

Subject reduction holds.

Theorem (Parigot 1992; Py 1998)

Let M, N be such that $\Gamma \vdash M : A$ and $M \to^*_{\lambda\mu} N$. Then $\Gamma \vdash N : A$.

The calculus is strongly normalizing when we omit μ'-reduction.

Theorem (Parigot 1997; David and Nour 2003)

The $\lambda\mu$-calculus is strongly normalizing.

Proof-theoretical properties

We add the rule μ' to the reduction: we denote $\to_{\lambda\mu\mu'\rho\tau\varepsilon}$ by $\to_{\lambda\mu\mu'}$ when we emphasize that we consider the totality of the rules. Otherwise, we simply write $\to$ for the reduction involving all the rules. First of all, we may observe that we do not have confluency anymore:

$$(\mu\alpha.x)\mu\beta.y \to_{\mu} \mu\alpha.x \text{ and } (\mu\alpha.x)\mu\beta.y \to_{\mu'} \mu\beta.y.$$

Strangely enough, we can even find a term of integer type that reduces to two different numerals. Namely, let $M = \mu\alpha.[\alpha](\mu\beta.[\alpha]\underline{n})\mu\delta.[\alpha]\underline{m}$. Then $\vdash M : \mathcal{N}$, and both $M \to^* \underline{n}$ and $M \to^* \underline{m}$ holds.

Proof-theoretical properties

- Does the typed (untyped) $\mu\mu'$-reduction have the SN/WN property?
- Is $\beta\mu\mu'$-reduction (extended with some of the simplification rules) weakly/strongly normalizing?

Proof-theoretical properties

- The simply typed $\mu\tilde{\mu}$-calculus is strongly normalizing (Polonovski 2004)
- The untyped $\mu\mu'$-calculus is strongly normalizing (David; Nour 2007). A proof based on a non-decreasing norm was given in 2019 by Battyányi and Nour.

Proof-theoretical properties

What happens when we add one or more simplification rules?

Proposition (B 2007)

$\mu\mu'\rho$-reduction is not strongly normalizing.

Let $U = \mu\alpha.[\alpha][\alpha]x$, $V = \mu\beta.U$ and $M = (V)U$.

- The term M can be typed. Let $\alpha : \neg\bot$ and $\beta : \neg(\bot \to \bot)$, then we have $x : \bot \vdash U : \bot$, $x : \bot \vdash V : \bot \to \bot$ and $x : \bot \vdash M : \bot$.
- There exists an infinite reduction sequence starting from M.

$$\begin{array}{rcl} M & = & \underline{(V)\mu\alpha.[\alpha][\alpha]x} \\ & \to_{\mu'} & \mu\alpha.[\alpha](V)[\alpha]\underline{(V)x} \\ & \to_{\mu} & \mu\alpha.[\alpha](V)\underline{[\alpha]\mu\beta.U} \\ & \to_{\rho} & \mu\alpha.[\alpha](V)U = \mu\alpha.[\alpha]M \end{array}$$

We remark that, when θ-reduction is allowed, the example above returns the initial μ-term M.

Proposition (B 2007)

$\mu\mu'\varepsilon$-reduction is not weakly normalizing.

Let $K = \mu\alpha.[\alpha][\alpha]x$, $L = \mu\beta.[\beta](z)[\beta]y$ and $N = \mu\gamma.(L)K$. Assume $\alpha : \neg\bot$, $\beta : \neg(\bot \to \bot)$ and $\gamma : \neg\bot$. Then we have $x : \bot \vdash K : \bot$, $y : \bot \to \bot, z : \bot \to (\bot \to \bot) \vdash L : \bot \to \bot$ and $x : \bot, y : \bot \to \bot, z : \bot \to (\bot \to \bot) \vdash N : \bot$.

Proof-theoretical properties

The following reduction sequence starting from N will never come to a halt.

$$
\begin{array}{lll}
N & = & \mu\gamma.\underline{(\mu\beta.[\beta](z)[\beta]y)K} \\
& \to_{\mu} & \underline{\mu\gamma.\mu\beta.[\beta]((z)[\beta](y)K)K} \\
& \to_{\varepsilon} & \mu\gamma.\underline{((z)(y)K)K} \\
& \to_{\mu'} & \underline{\mu\gamma.\mu\alpha.[\alpha]((z)(y)K)[\alpha]((z)(y)K)x} \\
& \to_{\varepsilon} & \mu\gamma.((z)(y)K)((z)\underline{(y)K})x \\
& \to_{\mu'} & \mu\gamma.((z)(y)K)(\underline{(z)\mu\alpha.[\alpha](y)[\alpha](y)x})x \\
& \to_{\mu'} & \mu\gamma.((z)(y)K)\underline{(\mu\alpha.[\alpha](z)(y)[\alpha](z)(y)x)x} \\
& \to_{\mu} & \mu\gamma.\underline{((z)(y)K)\mu\alpha.[\alpha]((z)(y)[\alpha]((z)(y)x)x)x} \\
& \to_{\mu'} & \underline{\mu\gamma.\mu\alpha.[\alpha]((z)(y)K)((z)(y)[\alpha]\underline{((z)(y)K)((z)(y)x)x)x} \\
& \to_{\varepsilon} & \mu\gamma.((z)(y)K)((z)(y)((z)(y)K)((z)(y)x)x)x \\
& \ldots &
\end{array}
$$

Proof-theoretical properties

Theorem (B; Nour 2022)

The $\mu\mu'\rho\varepsilon$-rule is weakly normalizing.

The underlying algorithm is constructive and non-deterministic. There exist terms for which we can obtain normal forms that are not $\mu\mu'\rho\varepsilon$-equal.

What if we add the β-rule? Since the above counterexamples can be typed, weak normalization is the most that can be expected.

Weak normalization of $\beta\mu\mu'\rho\varepsilon$

Theorem (B; Nour 2022)

$\beta\mu\mu'\rho\varepsilon$-reduction is weakly normalizing.

The idea of the algorithm is as follows:

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

Weak normalization of $\beta\mu\mu'\rho\varepsilon$: the origin of a μ-redex

Lemma

Let

$$\begin{array}{ccc} M_1 & \to_\beta & M_2 \\ & & \mid\vee \\ & & (\mu\alpha^{\neg(A\to B)}.P_1^\perp)Q_1^\perp \end{array}$$

Then either

$$(\mu\alpha^{\neg(A\to B)}.P_2^\perp)Q_2^\perp \quad \leq \quad M_1$$

or

$$(\lambda x.^C.P_2^D)Q_2^C \quad \leq \quad M_1,$$

where $lh(C \to D) > lh(A \to B)$.

Weak normalization of $\beta\mu\mu'\rho\varepsilon$: the origin of a μ-redex

Lemma

Let

$$\begin{array}{ccc} M_1 & \to_\beta & M_2 \\ & & \mid\vee \\ & & (\lambda x^A.P_1^B)Q_1^A \end{array}$$

Then

$$(\lambda x^C.P_2^D)Q_2^C \quad \leq \quad M_1$$

with $lh(C \to D) \geq lh(A \to B)$.

Weak normalization of $\beta\mu\mu'\rho\varepsilon$: the origin of a μ-redex

Corollary

If $M \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$, $M \twoheadrightarrow_\beta M'$ and $(\mu\alpha^{\neg(A\to B)}.P_1^{\perp})Q_1^A \leq M'$, then $(\lambda x^C.P_2^D)Q_2^C \leq M$ for some terms P_2, Q_2 and some types C, D where $\mathrm{lh}(C \to D) > \mathrm{lh}(A \to B)$.

Weak normalization of $\beta\mu\mu'\rho\varepsilon$: the origin of a β-redex

Lemma

Let

$$\begin{array}{ccc} M_1 & \to_{\mu\mu'\rho\varepsilon} & M_2 \\ & & |\vee \\ & & (\lambda x^A.P_1^B)Q_1^A \end{array}$$

Then either

$$(\lambda x^A.P_2^B)Q_2^A \quad \leq \quad M_1$$

or

$$(\mu\alpha^{\neg(A\to B)}.P_2^{\perp})Q_2^{\perp} \quad \leq \quad M_1$$

such that there exists $[\alpha]\mu\beta_1 \ldots [\beta_1] \ldots \mu\beta_n \ldots [\beta_n]\lambda y^A.R^B \leq P_2^{\perp}$.

Weak normalization of $\beta\mu\mu'\rho\varepsilon$: the origin of a β-redex

Corollary

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

Weak normalization of $\beta\mu\mu'\rho\varepsilon$

Definition

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

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

Lemma

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

Weak normalization of $\beta\mu\mu'\rho\varepsilon$

Lemma

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

Proof.

By induction on $\mathrm{rank}(M)$. If $M \notin \mathcal{NF}_{\beta}$, then $M \twoheadrightarrow_{\beta} M'$ and $M' \in \mathcal{NF}_{\beta}$ for some M'. If $M' \notin \mathcal{NF}_{\mu\mu'\rho\varepsilon}$, then $M' \twoheadrightarrow_{\mu\mu'\rho\varepsilon} M''$ and $M'' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$. Finally, if $M'' \notin \mathcal{NF}_{\beta}$, by the previous lemma, $\mathrm{rank}(M) > \mathrm{rank}(M'')$ and, by IH, $M'' \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$, then $M \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$. □

Weak normalization of $\beta\mu\mu'\rho\varepsilon$

Theorem

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

Proof.

Let M be a $\lambda\mu$-term and $M \twoheadrightarrow_{\mu\mu'\rho\varepsilon} M'$ where $M' \in \mathcal{NF}_{\mu\mu'\rho\varepsilon}$. By the above lemma, $M' \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$, hence $M \in \mathcal{WN}_{\beta\mu\mu'\rho\varepsilon}$. □

The case of the θ-rule

We can extend our result to the case of θ-reduction. Recall that the θ-rule looks like as follows:

$$\mu\alpha^{\neg A}.[\alpha^{\neg A}]M^A : A \to_\rho M^A : A$$

provided $\alpha \notin fv(M)$. We have two lemmas:

Lemma

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

Lemma

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

The case of the θ-rule

As a consequence, we can state the following theorem:

Theorem

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

Bibliography


F. Barbanera and S. Berardi. *A symmetric lambda calculus for classical program extraction*, In: M. Hagiya and J.C. Mitchell (editors), Proceedings of theoretical aspects of computer software, TACS '94., Lecture Notes in Computer Science (789), pp. 495-515, Springer Verlag, 1994.


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


P. Battyányi and K. Nour. *Normalization proofs for the un-typed $\mu\mu'$-calculus*, Special Issue: LICMA'19 Lebanese International Conference on Mathematics and Applications., AIMS Mathematics, 5(4), pp. 3702-3713, 2020.

Bibliography


P. Battyányi, K. Nour. *Normalization in the simply typed $\lambda\mu\mu'\rho\theta\varepsilon$-calculus*, Mathematical Structures in Computer Science. 32(8), pp. 1066-1098, 2022. https://doi.org/10.1017/S096012952200041X


P.-L. Curien and H. Herbelin. *The duality of computation*, In: M. Odersky, P. Wadler (editors), Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming, ICFP '00, pp. 233-243, ACM Press, 2000.


R. David, R., K. Nour. *A Short Proof of the Strong Normalization of Classical Natural Deduction with Disjunction*, The Journal of Symbolic Logic, 68(4), pp. 1277-1288, 2003. http://www.jstor.org/stable/4147762

Bibliography

- R. David and K. Nour. *Arithmetical proofs of strong normalization results for symmetric lambda calculi*, Fundamenta Informaticae, 77(4), pp. 489-510, 2007.
- J.-Y. Girard, Y. Lafont and P. Taylor. *Proofs and Types*, Cambridge University Press, 1989.
- T. Griffin. *A formulae-as-type notion of control*, In: F. E. Allen (editor), Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, POPL '90, ACM Press, 1990.
- P. de Groote. *An environment machine for the $\lambda\mu$-calculus*, Mathematical Structures in Computer Science 8, pp. 637-669, 1998.

Bibliography

- J.-L. Krivine. *Lambda-calculus types and models*, Ellis Horwood, 1993.
- J.-L. Krivine. *Classical logic, storage operators and 2nd order lambda-calculus*, Annals of Pure and Applied Logic 68, pp. 53-78, 1994.
- C. R. Murthy. *An evaluation semantics for classical proofs*, In: Proceedings of the sixth annual IEEE symposium on logic in computer science, pp. 96-107, 1991.
- K. Nour. *La valeur d'un entier classique en $\lambda\mu$-calcul*, Archive for Mathematical Logic 36, pp. 461-473, 1997.

Bibliography

- M. Parigot. *$\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction*, In: A. Voronkov (editor), Logic Programming and Automated Reasoning, Lecture Notes in Computer Science (624), Springer Verlag, Berlin, pp. 190-201, 1992.
- M. Parigot. *Proofs of strong normalization for second order classical natural deduction*, Journal of Symbolic Logic (62), pp. 1461-1479, 1997.
- W. Py. *Confluence en $\lambda\mu$-calcul*, PhD thesis, University of Chambéry, 1998.
- N. J. Rehof and M. H. Sørensen. *The λ_Δ-calculus*, In: M. Hagiya, J. C. Mitchell (editors), Theoretical Aspects of Computer Software, Lecture Notes in Computer Science (789), pp. 516-542, Springer Verlag, 1994.

Bibliography


A. Saurin. *On the Relations between the Syntactic Theories of $\lambda\mu$-calculi*, In: M. Kaminski, S. Martini (editors), 17th EACSL Annual Conference on Computer Science Logic, Lecture Notes in Computer Science (5213), pp. 154-168, Springer Verlag, 2008.


M. H. Sørensen and P. Urzyczyn *Lectures on the Curry-Howard Isomorphism*, Elsevier Science, 2006.


P. Wadler. *Call-by-value is dual to call-by-name*, In: C. Runciman, O. Shivers (editors), Proceedings of the Eighth ACM SIGPLAN International Conference on Functional Programming, ICFP '03, pp. 189-201, 2003.