Categorical semantics of normalization in $\lambda\mathcal{C}$-calculus

Vladimir Komendantsky*
Boole Centre for Research in Informatics
University College Cork
v.komendantsky@bcri.ucc.ie

Abstract

We investigate normalization in call-by-name formulation of $\lambda\mathcal{C}$-calculus, a constructive analogue of classical natural deduction, by inverting the evaluation functional in a general setting of $\mathcal{P}$-category theory. We obtain a decision procedure for $\lambda\mathcal{C}$-calculus by comparing normal forms of $\lambda\mathcal{C}$-terms in this setting.

Keywords: $\lambda\mathcal{C}$-calculus, $\lambda\mu$-calculus, normalization, $\mathcal{P}$-category theory

1 Introduction

$\lambda\mathcal{C}$-Calculus is often viewed as a computational version of Gentzen's classical natural deduction system ND [3, 6], and it is also useful for studying continuations [4, 8] in functional programming languages. The first mentioned aspect is of our primary interest. In this paper we consider normalization in call-by-name version of $\lambda\mathcal{C}$. The categorical approach to normalization is based on inverting the evaluation functional and has been developed in relation to λ-calculus, e.g., in [1, 2]. Particularly, in [2] there was employed a special case of enriched categories called $\mathcal{P}$-categories, i.e. categories with partial equivalence relations on arrows. Also there, $\mathcal{P}$-ccc's were proved to model normalization in simply-typed λ-calculus. We extend this approach by considering a notion weaker than that of a ccc, namely a notion of a category of continuations. This allows us to model normalization in $\lambda\mathcal{C}$-calculus. Our construction is also applicable to normalization in $\lambda\mu$-calculus.

2 $\lambda\mathcal{C}$-calculus and $\lambda\mu$-calculus

The $\lambda\mathcal{C}$-calculus is the simply-typed λ-calculus with augmented variable binding: if t is a $\lambda\mathcal{C}$-term of type $\perp$ then $\mathcal{C}x^{\neg A}.t$ is a $\lambda\mathcal{C}$-term of type A. The operator $\mathcal{C}$ only binds variables of negated type. The sequent $\Gamma \vdash t : A$ where Γ is a set of variable-type annotations of the kind $y : B$, means that the $\lambda\mathcal{C}$-term t is a

*Research funded by a BCRI postdoctoral grant.

representation of a classical natural deduction proof of the proposition A whose undischarged hypotheses are annotated propositions taken from the set Γ.

For technical reasons, we define an algorithm that translates a sequent of $\lambda\mathcal{C}$-calculus into a sequent of $\lambda\mu$-calculus of the same type. Following Ong [6], we assume that there is a bijection between variables annotating negated hypotheses of the form $\neg A$, where $A \neq \bot$, and μ-variables, given by $\overline{(-)}$, e.g. $\overline{x^{\neg A}} = \alpha^A$ and $\overline{\overline{x^{\neg A}}} = x^{\neg A}$, with the inverse being $\underline{(-)}$. Take a $\lambda\mathcal{C}$-sequent $\Gamma \vdash t : A$. Let Θ be a subset of Γ consisting only of negated hypotheses. We define a $\lambda\mu$-term $\lceil t \rceil^\Theta$ by recursion: $\lceil x \rceil^\Theta \stackrel{\text{def}}{=} x$ if $x \notin \Theta$ and $\lambda y^A.[\alpha^A]y$ otherwise; $\lceil \lambda x^A.s \rceil^\Theta \stackrel{\text{def}}{=} \lambda x^A.\lceil s \rceil^\Theta$; $\lceil rs \rceil^\Theta \stackrel{\text{def}}{=} \lceil r \rceil^\Theta \lceil s \rceil^\Theta$; $\lceil \mathcal{C}x^{\neg A}.s \rceil^\Theta \stackrel{\text{def}}{=} \mu\alpha^A.\lceil s \rceil^{\Theta, x:\neg A}$. Applying this algorithm we obtain $\Gamma \setminus \Theta \vdash \lceil t \rceil^\Theta : A \mid \overline{\Theta}$ which is a $\lambda\mu$-sequent.

One can think of the $\lambda\mu$-calculus as a variant of ND with ability to distinguish between hypotheses which ought to be discharged by the classical absurdity rule (by annotation of μ-variables), and from those which ought to be discharged by implication introduction (by annotation of λ-variables).

The inverse translation is defined as follows. Take a $\lambda\mu$-sequent $\Gamma \vdash t : A \mid \Delta$. We define a $\lambda\mathcal{C}$-term $\lfloor t \rfloor$ by recursion: $\lfloor x^A \rfloor \stackrel{\text{def}}{=} x^A$; $\lfloor \lambda x^A.s \rfloor \stackrel{\text{def}}{=} \lambda x^A.\lfloor s \rfloor$; $\lfloor rs \rfloor \stackrel{\text{def}}{=} \lfloor r \rfloor \lfloor s \rfloor$; $\lfloor [\alpha^A]s \rfloor \stackrel{\text{def}}{=} x^{\neg A} \lfloor s \rfloor$; $\lfloor \mu\alpha^A.s \rfloor \stackrel{\text{def}}{=} \mathcal{C}x^{\neg A}.\lfloor s \rfloor$. Thus we obtained a $\lambda\mathcal{C}$-sequent $\Gamma, \underline{\Delta} \vdash \lfloor t \rfloor : A$. The translation $\lfloor - \rfloor$ forgets the difference between the two types of hypotheses, and so it can be argued that the resulting $\lambda\mathcal{C}$-terms reflect properties of ND-proofs better than the $\lambda\mu$-terms do.

Now, combining argumentation of Ong [6] and of de Groote [3] about the two translation algorithms we can prove the following theorem.

Theorem 1. (i) For any $\lambda\mathcal{C}$-derivable sequent $\Gamma \vdash t : A$ and for any subset Θ of Γ consisting of negated hypotheses, the sequent $\Gamma \setminus \Theta \vdash \lceil t \rceil^\Theta : A \mid \overline{\Theta}$ is $\lambda\mu$-derivable. (ii) For any $\lambda\mu$-derivable sequent $\Gamma \vdash t : A \mid \Delta$, the sequent $\Gamma, \underline{\Delta} \vdash \lfloor t \rfloor : A$ is $\lambda\mathcal{C}$-derivable.

In fact, $\lambda\mathcal{C}$-calculus and $\lambda\mu$-calculus are isomorphic, as it was first noted by de Groote [3]. Formally:

Theorem 2. (i) $\Gamma \vdash \lceil \lfloor t \rfloor \rceil^{\underline{\Delta}} = t : A \mid \Delta$ and (ii) $\Gamma, \Theta \vdash \lfloor \lceil t \rceil^\Theta \rfloor = t : A$.

Assume a signature $(\mathcal{B}, \mathcal{K})$ consisting of base types (excluding $\bot$) and constants respectively. We define simply-typed call-by-name $\lambda\mathcal{C}$-calculus.
Axioms and rules

$$(\text{Axiom}) \quad \Gamma \vdash x : A \quad \text{if } x : A \in \Gamma \qquad (\text{Const}) \quad \Gamma \vdash c : A \quad \text{if } c : A \in \mathcal{K}$$

$$(\Rightarrow\text{-intro}) \quad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x^A.t : A \Rightarrow B} \qquad (\Rightarrow\text{-elim}) \quad \frac{\Gamma \vdash t : A \Rightarrow B \qquad \Gamma \vdash s : A}{\Gamma \vdash ts : B}$$

$$(\neg\neg\text{-elim}) \quad \frac{\Gamma \vdash \lambda x^{\neg A}.t : \neg\neg A}{\Gamma \vdash \mathcal{C}x^{\neg A}.t : A} \quad \text{if } A \neq \bot \text{ and } x : \neg A \notin \Gamma$$

Equations in context

$$\Gamma \vdash t = t : A \qquad \frac{\Gamma \vdash s = t : A}{\Gamma \vdash t = s : A} \qquad \frac{\Gamma \vdash s = t : A \quad \Gamma \vdash t = r : A}{\Gamma \vdash s = r : A}$$

$$\frac{\Gamma \vdash s_1 = t_1 : A \Rightarrow B \quad \Gamma \vdash s_2 = t_2 : A}{\Gamma \vdash s_1 s_2 = t_1 t_2 : B} \qquad \frac{\Gamma, x : A \vdash s = t : B}{\Gamma \vdash \lambda x^A.s = \lambda x^A.t : A \Rightarrow B}$$

$$\frac{\Gamma \vdash \lambda x^{\neg A} s = \lambda x^{\neg A}.t : \neg\neg A}{\Gamma \vdash \mathcal{C}x^{\neg A}.s = \mathcal{C}x^{\neg A}.t : A}$$

$$(\beta_{\Rightarrow})\quad \Gamma\vdash(\lambda x^{A}.t)s=t(s/x):B$$
$$(\beta_{\bot})\quad \Gamma\vdash x'\mathcal{C}x^{\neg A}.t=t[x/x']:\bot$$
$$(\eta_{\Rightarrow})\quad \Gamma\vdash t=\lambda x^{A}.tx:A\Rightarrow B,\quad \text{if } x:A\notin\Gamma$$
$$(\eta_{\bot})\quad \Gamma\vdash \mathcal{C}x^{\neg A}.xt=t:A,\quad \text{if } x\notin \mathrm{FV}(t)$$
$$(\zeta_{\Rightarrow})\quad \Gamma\vdash(\mathcal{C}x^{\neg(A\Rightarrow B)}.t)s=\mathcal{C}y^{\neg B}.t[y((-)s)/x(-)]:B,\quad \text{if } y\notin \mathrm{FV}(ts)$$
$$(\zeta^{\bot}_{\Rightarrow})\quad \Gamma\vdash(\mathcal{C}x^{\neg\neg A}.t)s=t[(-)s/x(-)]:\bot$$
$$(\zeta_{\bot})\quad \Gamma\vdash xt=t\{x\}:\bot\qquad \text{if } x:\neg A\in\Gamma \text{ and } t\{x\} \text{ is defined}$$

The *renaming function* $(-)\{-\}$ is defined in the following cases:
(i) $(\mathcal{C}x^{\neg B}.t)\{y\}\stackrel{\text{def}}{=}t[y/x]$ and (ii) $(\lambda x^{B}.t)\{y\}\stackrel{\text{def}}{=}t\{y'\}[y(\lambda x^{B}.s)/y's]$ for some fresh variable y if x occurs in $t\{y'\}$ only within the scope of $y's$, otherwise $(-)\{-\}$ is undefined.

Pym and Ritter [7] gave a confluent (i.e. any two reducts of a term have a common reduct) and strongly normalizing (i.e. all reduction sequences of any given term are terminating) call-by-name rewriting semantics for the $\lambda\mu$-calculus based on the translations of the above axioms. Therefore, due to Theorems 1 and 2, we can state the properties of confluence and strong normalization for the $\lambda\mathcal{C}$-calculus which are crucial for our discussion of the normalization algorithm.

3 The idea of a normal form algorithm

The decision problem for $\lambda\mathcal{C}$-calculus can be formulated as follows: For any possibly open $\lambda\mathcal{C}$-terms t and s of type A, decide whether $\Gamma\vdash t=s:A$. With each $\lambda\mathcal{C}$-term t we associate its *abstract normal form* $\mathsf{nf}(t)$ such that the following properties hold:

$$\text{(NF1)}\ \Gamma\vdash\mathsf{nf}^{-1}(\mathsf{nf}(t))=t:A,\qquad \text{(NF2)}\ \Gamma\vdash t=s:A \text{ implies } \mathsf{nf}(t)=\mathsf{nf}(s).$$

Since the conditions (NF1) and (NF2) imply $\Gamma\vdash t=s:A$ iff $\mathsf{nf}(t)=\mathsf{nf}(s)$, comparing abstract normal forms can yield a decision procedure for $\lambda\mathcal{C}$-calculus.

A categorical model of $\lambda\mathcal{C}$-calculus is a category of continuations. According to [1], such a category $\mathbf{C}$ has a distinguished class $\mathcal{T}$ of objects of $\mathbf{C}$ called type objects and a distinguished type object R of responses, provided that $\mathcal{T}$ contains an interpretation of the base types $\mathcal{B}$. Additionally, there is a chosen cartesian product $\Gamma\cdot A$ for every object Γ and a type object A, and chosen terminal objects $[]$ and 1 in $\mathcal{T}$. Also for each type object A there is a chosen exponential $R^{A}\in\mathcal{T}$, and for any two type objects A and B a chosen cartesian product $R^{A}\times B\in\mathcal{T}$. A $\lambda\mathcal{C}$-sequent $\Gamma,\Theta\vdash t:A$ is interpreted in $\mathbf{C}$ as a map $R^{[\![\Gamma]\!]}\cdot[\![\Theta]\!]\to R^{[\![A]\!]}$. An objects of $\mathbf{C}$ is an interpretation of a continuation context Θ; a morphism from Θ to A is a $\lambda\mathcal{C}$-term t such that $\Theta\vdash t:\neg A$.

Let us denote a free $\mathcal{P}$-category of continuations on the signature $\Sigma=(\mathcal{B},\mathcal{K})$ as $\mathbf{F}_{\Sigma}$. The universal property of a free $\mathcal{P}$-category of continuations $\mathbf{F}_{\Sigma}$ is as follows: for any $\mathcal{P}$-category of continuations $\mathbf{C}$, and any interpretation of the signature Σ in $\mathbf{C}$, there is a unique up to isomorphism structure preserving $\mathcal{P}$-functor $[\![-]\!]:\mathbf{F}_{\Sigma}\to\mathbf{C}$ freely extending this interpretation. There are two straightforward $\mathcal{P}$-functors preserving the structure of $\mathcal{P}$-categories of continuations: $\mathcal{P}$-categorical Yoneda embedding $Y:\mathbf{F}_{\Sigma}\to\mathcal{P}\mathbf{Set}^{\mathbf{F}_{\Sigma}{}^{op}}$ and the free extension to the $\mathcal{P}$-functor $[\![-]\!]:\mathbf{F}_{\Sigma}\to\mathcal{P}\mathbf{Set}^{\mathbf{F}_{\Sigma}{}^{op}}$. By the universal property, there is a $\mathcal{P}$-natural isomorphism $q : [\![-]\!] \to Y$. To obtain a function nf we invert the $\mathcal{P}$-natural Yoneda isomorphism q. Given a sequent $\Gamma \vdash t : A$ we define (leaving out the square brackets in subscripts to improve readability)

$$\mathsf{nf}(t) = q_{A,\Gamma}([\![t]\!]_\Gamma(q^{-1}_{\Gamma,\Gamma}(\mathrm{id}_\Gamma))) \ .$$

Since $[\![-]\!]$ is an interpretation, we have (NF2), that is $\Gamma \vdash t = s$ implies $[\![t]\!] = [\![s]\!]$, and (NF1) is proved by a straightforward induction on t. Therefore $[\![-]\!]$ is a sound and complete interpretation. Hence we have the following theorem.

Theorem 3. (i) For each $\Gamma \vdash t : A$, $\mathsf{nf}(t)$ is an element of $\mathsf{NF}(\Gamma, A)$. (ii) Every element of $\mathsf{NF}(\Gamma, A)$ is $\mathsf{nf}(t)$ for some t.

Among the possible future directions we would wish to address elsewhere we emphasise the following: 1) an application of the $\mathcal{P}$-categorical approach to normalization to $\lambda\mu$-categories [6] or control categories [8]; 2) a study of normalization in call-by-value formulation of $\lambda\mathcal{C}$, e.g., in the setting of precartesian-closed abstract Kleisli categories of Führmann and Thielecke [4]; 3) an analysis of $\mathcal{P}$-categorical models of $\lambda\mathcal{C}$-calculus in a higher category theory setting (this may be analogous to a bicategorical analysis of E-categories given by Kinoshita [5]).

References

[1] T. Altenkirch, M. Hofmann, and T. Streicher. Categorical reconstruction of a reduction-free normalization proof. In *Proc. CTCS '95*, volume 953 of *LNCS*, pages 182–199. Springer, 1995.

[2] D. Čubrić, P. Dybjer, and P. Scott. Normalization and the Yoneda embedding. *Mathematical Structures in Computer Science*, 8(2):153–192, 1998.

[3] P. de Groote. On the relation between the $\lambda\mu$-calculus and the syntactic theory of sequential control. In F. Pfenning, editor, *Proc. 5th Intl. Conf. Logic Programming and Automated Reasoning (LPAR'94)*, volume 822 of *LNCS*, pages 31–43, 1994.

[4] C. Führmann and H. Thielecke. On the call-by-value CPS transform and its semantics. *Information and Computation*, 188(2):241–283, 2004.

[5] Y. Kinoshita. A bicategorical analysis of E-categories. *Mathematica Japonica*, 47(1):157–169, 1998.

[6] C.-H. L. Ong. A semantic view of classical proofs: Type-theoretic, categorical, and denotational characterizations. In *Proc. 11th Annual IEEE Symp. on Logic in Computer Science*, pages 230–241, 1996.

[7] D. Pym and E. Ritter. On the semantics of classical disjunction. *J. Pure and Applied Algebra*, 159:315–338, 2001.

[8] P. Selinger. Control categories and duality: on the categorical semantics of the lambda-mu calculus. *Math. Struct. Comp. Science*, 11:207–260, 2001.