Completeness of continuation models for $\lambda\mu$-calculus

Martin Hofmann and Thomas Streicher*

Synopsis

We show that a certain simple call-by-name continuation semantics of Parigot's λ_μ-calculus is complete. More precisely, for every λ_μ-theory we construct a cartesian closed category such that the ensuing continuation-style interpretation of λ_μ, which maps terms to functions sending abstract continuations to responses, is full and faithful.

Thus, any λ_μ-category in the sense of [8] is isomorphic to a continuation model [6] derived from a cartesian-closed category of continuations.

We also extend this result to a later call-by-value version of λ_μ developed by Ong and Stewart [7].

1 Introduction and Summary

Parigot's λ_μ-calculus [9] is a proof term assignment system for propositional classical logic and can at the same time be considered as a prototype for a functional programming language incorporating explicit handling of continuations. The original motivation for this calculus was to give a functional interpretation for proofs in classical AF_2—a certain system of second-order arithmetic [5].

Ong [8] has defined a categorical notion of model for this calculus for which the usual categorical completeness theorem holds. In this sense Ong's semantics can be seen as a variable-free reformulation of the syntax of λ_μ. On the other hand, there exists a class of rather concrete continuation models for λ_μ where terms are interpreted as functions[1] mapping abstract continuations to answers. We prove in this paper that every λ_μ-theory (thus every model in the sense of Ong) is induced by a particular continuation model.

A similar result for call-by-value lambda-calculus with control operators has been obtained in [4] by category-theoretic means and independently by Felleisen and Sabry using syntactic back-and-forth translations [13]. The technique we use here is inspired by the method used in [4] in the sense that the morphisms of the continuation category to be constructed arise as special terms of a λ_μ-theory. Whereas in loc. cit. these special terms are defined by their syntactic form we use an equational description involving quantification over all observations.

Unlike in the case of [4] or [13] the equational axiomatisation of λ_μ under consideration was not specially tailored towards completeness for continuation models, which were apparently not known to Parigot at the time, but rather arose from syntactic considerations. For instance, it gives rise to a confluent and strongly normalising rewrite system [9].

The fact that by our result this axiomatisation is complete for continuation models thus provides evidence that these models are a very natural semantics for "proof-relevant" classical logic.

*TH Darmstadt, FB 4, Schlossgartenstr. 7, 64289 Darmstadt, Germany, phone: +49-6151-162794, fax: +49-6151-164011, e-mail:{mh|streicher}@mathematik.th-darmstadt.de

[1] In the sense of cartesian-closed categories.

$$\text{(Axiom)} \quad \frac{}{\Gamma \mid \Delta \vdash x : A} \quad \text{if } x{:}\, A \in \Gamma$$

$$\text{(Const)} \quad \frac{}{\Gamma \mid \Delta \vdash c : A} \quad \text{if } c{:}\, A \in \mathcal{K}$$

$$(\Rightarrow\text{-intro}) \quad \frac{\Gamma, x{:}\, A \mid \Delta \vdash t : B}{\Gamma \mid \Delta \vdash \lambda x{:}\, A.t : A \Rightarrow B}$$

$$(\Rightarrow\text{-elim}) \quad \frac{\Gamma \mid \Delta \vdash t : A \Rightarrow B \qquad \Gamma \mid \Delta \vdash s : A}{\Gamma \mid \Delta \vdash ts : B}$$

$$(\bot\text{-elim}) \quad \frac{\Gamma \mid \Delta, \alpha{:}\, A \vdash t : \bot}{\Gamma \mid \Delta \vdash \mu\alpha{:}\, A.t : A}$$

$$(\bot\text{-intro}) \quad \frac{\Gamma \mid \Delta \vdash t : A}{\Gamma \mid \Delta \vdash [\alpha]t : \bot} \quad \text{if } \alpha{:}\, A \in \Delta$$

Figure 1: Typing rules of λ_μ

A consequence of our result is that λ_μ-equality without nonlogical axioms can be reduced to equality of terms of simply-typed lambda calculus with products via a certain CPS translation derived from our semantics.

Later on, Ong and Stewart [7] have formulated a call-by-value version of λ_μ. The second main result in this paper is that a certain continuation semantics is complete for the latter system. Furthermore, we demonstrate that this system is isomorphic to the calculus in [4]. Therefore, our result can be transported to the latter system and thus provides a generalisation of the result in [4] as op. cit. was concerned with the bare calculus whereas we consider arbitrary equational theories.

2 The λ_μ-calculus

The presentation of λ_μ we use follows Ong's account in [8]. It differs from Parigot's original formulation only in the aspect that we omit continuation variables of type $\bot$. See loc. cit. for a more detailed comparison.

Assume a set $\mathcal{B}$ of base types. The types of λ_μ are the simple types over $\mathcal{B} \cup \{\bot\}$, i.e. every base type is a type, $\bot$ is a type, and if A, B are types so is $A \Rightarrow B$.

There are two sorts of variables. Object variables ranged over by Roman letters $x, y, z, \ldots$ and continuation variables ranged over by Greek letters $\alpha, \beta, \ldots$. An object context is a partial type assignment to object variables of the form $x_1{:}\, A_1, \ldots, x_n{:}\, A_n$. A continuation context is a partial type assignment to continuation variables of the form $\alpha_1{:}\, A_1, \ldots, \alpha_n{:}\, A_n$ where all the A_i are different from $\bot$.

Assume a set $\mathcal{K}$ of typed constants. The typing judgements of λ_μ take the form $\Gamma \mid \Delta \vdash t : A$ where Γ is an object context, Δ is a continuation context, A is a type, and t is a term. The precise form of the terms is given implicitly together with the rules defining the typing judgement set out in Figure 1. As usual we identify terms

$$
\begin{array}{ll}
(\beta) & (\lambda x\colon A.t)s = t[s/x]. \\[1em]
(\eta) & \lambda x\colon A.tx = t, \\
& \quad \text{when } x \text{ is not free in } t. \\[1em]
(\mu\text{-}\beta) & [\alpha]\mu\gamma\colon A.t = t[\alpha/\gamma]. \\[1em]
(\mu\text{-}\eta) & \mu\alpha\colon A.[\alpha]t = t, \\
& \quad \alpha \text{ not free in } t. \\[1em]
& \mu\alpha\colon A \Rightarrow B.t = \lambda x\colon A.\mu\beta\colon B.t[x{::}\beta \;/\; \alpha], \\
(\mu\text{-}\zeta) & \quad \text{when } B \not\equiv \bot. \\
& \mu\alpha\colon A \Rightarrow \bot.t = \lambda x\colon A.t[x{::}\star \;/\; \alpha].
\end{array}
$$

Figure 2: Equality axioms for λ_μ

up to renaming of both object and continuation variables. Notice that λ *and* μ bind variables as indicated, but that continuation variable α occurs free in a term of the form $[\alpha]t$.

The typing rules are such that we have $x_1\colon A_1, \ldots, x_n\colon A_n \mid \alpha_1\colon B_1, \ldots \alpha_m\colon B_m \vdash t : A$ for some t iff $A_1 \wedge \cdots \wedge A_n \wedge \neg B_1 \wedge \cdots \wedge \neg B_m \rightarrow A$ is a tautology of classical propositional logic. Under this analogy the rule ($\bot$-elim) corresponds to proof by contradiction: in order to prove A it suffices to deduce a contradiction ($\bot$) from the assumption that A is false ($\alpha : A$). Rule ($\bot$-intro), on the other hand, is the canonical way of constructing contradictions: from a proof of A and an assumption that A is false ($\beta : A$).

We can also relate λ_μ to classical sequent calculus as follows. We have

$$x_1\colon A_1, \ldots, x_n\colon A_n \mid \alpha_1\colon B_1, \ldots \alpha_m\colon B_m \vdash t : A$$

for some t iff the sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m, A$ is derivable in Gentzen's sequent calculus LK. Under this analogy the two rules ($\bot$-intro) and ($\bot$-elim) correspond to addition and removal of $\bot$ on the right hand side of the "turnstile".

The rules of λ_μ are such that logical rules always have the last conclusion as main formula. Permuting a conclusion into this "active" position is recorded by an instance of ($\bot$-elim). Thus, the purpose of the $\bot$-rules is to display (the otherwise implicit) switching of focus, where "being in focus" means to be the main formula of the next logical rule.

We differ from Ong's presentation in that we allow for side context in rule (Axiom), i.e., the fact that variables other than x may be declared in Γ, and that α may appear free in t in rule ($\bot$-intro). Our (stronger) rules are derivable in Ong's system using the structural rules (weakening and contraction) which in turn are admissible in our system. Therefore, the same sequents are derivable in either system.

Definition 2.1 A λ_μ-theory (over a signature $(\mathcal{B}, \mathcal{K})$) is a set $\mathcal{E}$ of typed equations of the form $\Gamma \mid \Delta \vdash s = t : A$ where $\Gamma \mid \Delta \vdash s : A$ and $\Gamma \mid \Delta \vdash t : A$ such that

- $\mathcal{E}$ is a congruence stable under weakening,
- $\mathcal{E}$ contains all well-typed instances of the basic equality laws depicted in Figure 2.

$$
\begin{array}{rcll}
x[s::\beta\,/\,\alpha] & = & x & \\
(tt')[s::\beta\,/\,\alpha] & = & (t[s::\beta\,/\,\alpha])(t'[s::\beta\,/\,\alpha]) & \\
(\lambda y{:}\,C.t)[s::\beta\,/\,\alpha] & = & \lambda y{:}\,C.(t[s::\beta\,/\,\alpha]) & y \text{ not free in } s \\
(\mu\gamma{:}\,C.t)[s::\beta\,/\,\alpha] & = & \mu\gamma{:}\,C.(t[s::\beta\,/\,\alpha]) & \gamma \not\equiv \beta \text{ and not free in } s \\
([\gamma]t)[s::\beta\,/\,\alpha] & = & [\gamma](t[s::\beta\,/\,\alpha]) & \gamma \not\equiv \alpha \\
([\alpha]t)[s::\beta\,/\,\alpha] & = & [\beta]((t[s::\beta\,/\,\alpha])s) &
\end{array}
$$

Figure 3: Definition of mixed substitution

◊

The notation used in the equations deserves some explanation. The term $t[s/x]$ in rule (β) denotes the capture-free substitution of s for x in t and $t[\alpha/\gamma]$ denotes the capture-free substitution of continuation variable α for γ.

The term $t[s::\beta\,/\,\alpha]$ called *mixed substitution* of $s : A$ and $\beta : B$ for continuation variable $\alpha : A \Rightarrow B$ (where $B \not\equiv \bot$) is defined inductively by the clauses in Figure 3. The substituted term has the same type as t; the substituted variable (α) does not occur in $t[s::\beta\,/\,\alpha]$ unless α is free in s. Mixed substitution of continuation variables of type $A \Rightarrow \bot$ is defined analogously the key clause being

$$([\alpha]t)[s::\star\,/\,\alpha] = (t[s::\star\,/\,\alpha])s$$

Here $\star$ is part of the operation symbol.

The idea behind this so-called *mixed substitution* is that a continuation for a function of type $A \Rightarrow B$ can be understood as an argument s and a continuation β for the ensuing result. The substitution operation $t[s::\beta\,/\,\alpha]$ allows one to substitute such an "intended" continuation for a continuation variable. Since there are no continuations of type $\bot$, an "intended" continuation of type $A \Rightarrow \bot$ is simply an object of type A.

3 Continuation models of λ_μ

The λ_μ-calculus admits a simple and intuitive continuation semantics in an arbitrary category with enough products and exponentials, in particular in any cartesian closed category with a distinguished object R of *responses*.

Definition 3.1 [Category of continuations] A *category of continuations* is given by the following data:

1. A category $\mathbf{C}$ with a distinguished class $\mathbf{T}$ of objects of $\mathbf{C}$ called *type objects*.
2. A distinguished type object R of *responses*.
3. For every object Γ and type object A a chosen cartesian product $\Gamma \cdot A$.
4. A chosen terminal object [] (for the empty context).
5. A chosen terminal object $1 \in \mathbf{T}$ (to interpret $\bot$).
6. For every type object A a chosen exponential $R^A \in \mathbf{T}$ of R by A.

7. For any two type objects A and B a chosen cartesian product $R^A \times B \in \mathbf{T}$ of R^A and B.

$\Diamond$

Clearly, $[] \cong 1$ and $R^A \times B \cong R^A \cdot B$. The presence of these isomorphic copies of terminal objects and cartesian products is not strictly necessary, for instance, we could postulate that a product $R^A \cdot B$ must be a type object if A and B are. However, they reflect syntactic distinctions and facilitate the formulation of term models.

Particular examples of continuation categories are the category of sets and various categories of domains where a natural choice for R is the set (domain) of screen outputs or alternatively truth values (in the case of sets) and the Sierpinski space (two element poset) in the case of domains. A further important example is furnished by the term model of a simply-typed lambda calculus together with a distinguished base type R. This model is generic in the sense that if a certain equation holds in it then it must hold in any other continuation category.

Assume for the rest of this section a fixed category of continuations. Any assignment of type objects $[\![B]\!]$ to base types B extends to an assignment of type objects to all types by the following two clauses.

$$\begin{array}{rcl} [\![\bot]\!] & = & 1 \\ [\![A \Rightarrow B]\!] & = & R^{[\![A]\!]} \times [\![B]\!] \end{array}$$

The intention is that $[\![A]\!]$ is the space of abstract continuations of type A. Accordingly, we call $R^{[\![A]\!]}$ the (type) object of *denotations* of type A. This may explain the definition of $[\![A \Rightarrow B]\!]$: A continuation for a function is given by an argument (a denotation of type A) and a continuation for the result.

Let $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$ be an object context and $\Delta \equiv \alpha_1 : B_1, \ldots, \alpha_m : B_m$ be a continuation context. We use the notation $R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!]$ for the object

$$[] \cdot R^{[\![A_1]\!]} \cdot \ldots \cdot R^{[\![A_n]\!]} \cdot [\![B_1]\!] \cdot \ldots \cdot [\![B_m]\!]$$

The subsequent interpretation of λ_μ is motivated by the following two natural isomorphisms familiar from the more special case of CCC's. The reader is invited to keep those in mind when going through the semantic clauses below.

Proposition 3.2 *Let A, B be type objects and X any object of a continuation category $\mathbf{C}$. We have the following two isomorphisms natural in X.*

$$\begin{array}{rcl} \mathbf{C}(X \cdot R^A, R^B) & \cong & \mathbf{C}(X, R^{R^A \times B}) \\ \mathbf{C}(X \cdot A, R^1) & \cong & \mathbf{C}(X, R^A) \end{array}$$

Assume an assignment of denotations to the constants, i.e. a morphism $[\![c]\!] : [] \to R^{[\![A]\!]}$ if $c{:}\,A$ is in $\mathcal{K}$. To each sequent $\Gamma \,|\, \Delta \vdash t : A$ we associate an arrow

$$[\![\Gamma \,|\, \Delta \vdash t : A]\!] : R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!] \to R^{[\![A]\!]}$$

by the clauses in Figure 4 where an informal lambda calculus ("internal language") is employed to simplify the notation. Notice that we use pattern matching for abstractions over product and unit types. Furthermore, we use $\star$ to denote the unique element of $[\![\bot]\!] = 1$. For the case of untyped λ_μ such semantics has been defined in [12]. The crucial observation that $R^A \times B$ is an exponential of B by A in the category $\mathbf{C}_R$ which has the same objects as $\mathbf{C}$ and homsets given by $\mathbf{C}_R(X, Y) = \mathbf{C}(R^X, R^Y)$ has been made by several people around 1990, including [1] and [6].

Theorem 3.3 (Soundness) *The λ_μ-calculus is sound with respect to this interpretation in the sense that the set of equations $\Gamma \,|\, \Delta \vdash t_1 = t_2 : A$ where t_1, t_2 are appropriately typed terms and $[\![\Gamma \,|\, \Delta \vdash t_1 : A]\!] = [\![\Gamma \,|\, \Delta \vdash t_2 : A]\!]$ is a λ_μ-theory.*

$$
\begin{array}{lcl}
[\![\Gamma \,|\, \Delta \vdash x_i : A]\!](\vec{x} \,|\, \vec{\alpha}) & = & x_i \\
[\![\Gamma \,|\, \Delta \vdash \lambda x\colon A.t : A \Rightarrow B]\!](\vec{x} \,|\, \vec{\alpha}) & = & \underline{\lambda}\langle x, \beta\rangle\colon R^{[\![A]\!]} \times [\![B]\!].[\![\Gamma, x\colon A \,|\, \Delta \vdash t : B]\!](\vec{x}, x \,|\, \vec{\alpha})\beta \\
[\![\Gamma \,|\, \Delta \vdash ts : B]\!](\vec{x} \,|\, \vec{\alpha}) & = & \underline{\lambda}\beta\colon [\![B]\!].[\![\Gamma \,|\, \Delta \vdash t : A \Rightarrow B]\!](\vec{x} \,|\, \vec{\alpha})\langle [\![\Gamma \,|\, \Delta \vdash s : A]\!](\vec{x} \,|\, \vec{\alpha}), \beta\rangle \\
[\![\Gamma \,|\, \Delta \vdash \mu\alpha\colon A.t : A]\!](\vec{x} \,|\, \vec{\alpha}) & = & \underline{\lambda}\alpha\colon [\![A]\!].[\![\Gamma \,|\, \Delta, \alpha\colon A \vdash t : \bot]\!](\vec{x} \,|\, \vec{\alpha}, \alpha)\star \\
[\![\Gamma \,|\, \Delta \vdash [\alpha_i]t : \bot]\!](\vec{x} \,|\, \vec{\alpha}) & = & \underline{\lambda}\star\colon 1.[\![\Gamma \,|\, \Delta \vdash t : A]\!](\vec{x} \,|\, \vec{\alpha})\alpha_i
\end{array}
$$

Figure 4: Interpretation of λ_μ in a category of continuations

Proof. Induction on derivations using appropriate substitution lemmas. $\square$

4 Completeness of continuation semantics

Our aim in this section is to establish the following completeness result for the continuation semantics.

Theorem 4.1 *For every λ_μ-theory $\mathcal{E}$ over a signature $(\mathcal{B}, \mathcal{K})$ there exists a continuation category $(\mathbf{C}, R)$ and an interpretation of base types and constants with the following two properties.*

1. *$\mathcal{E}$ is the theory induced by this continuation model $(\mathbf{C}, R)$.*
2. *Let Γ be an object context, Δ be a continuation context and A be a type. Every $\mathbf{C}$-morphism $f : R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!] \to R^{[\![A]\!]}$ arises as the interpretation of some λ_μ-term $\Gamma \,|\, \Delta \vdash t : A$.*

For the rest of this section assume a fixed signature $(\mathcal{B}, \mathcal{K})$ and a theory $\mathcal{E}$. We will use the notation from Def. 2.1 to refer to the continuation category which we are going to construct from these data.

Before embarking on the actual construction let us first provide some intuition. Assume for the moment that $\mathcal{E}$ happens to be induced by a hypothetical continuation category $\mathbf{C}$. It suggests itself to recover $\mathbf{C}$ from the λ_μ-theory by using the continuation contexts as objects. Unfortunately, morphisms from $[\![\Delta]\!]$ to $[\![A]\!]$ do not arise as meanings of terms so that there is no straightforward way to recover the $\mathbf{C}$-morphisms from terms. However, the meaning of a term $\cdot \,|\, \Delta \vdash t : A \Rightarrow \bot$ is a $\mathbf{C}$-morphism from $[\![\Delta]\!]$ to $R^{R^{[\![A]\!] \cdot 1}} \cong R^{R^{[\![A]\!]}}$. Now every morphism $f : [\![\Delta]\!] \to [\![A]\!]$ induces a morphism from $[\![\Delta]\!]$ to $R^{R^{[\![A]\!]}}$ by composition with the curried evaluation map $\eta_{[\![A]\!]}(a\colon [\![A]\!]) = \underline{\lambda} p\colon R^{[\![A]\!]}.pa$. Alas, in general, it seems to be impossible to tell whether a given morphism $h : [\![\Delta]\!] \to R^{R^{[\![A]\!]}}$ factors through η. However, those h which do, satisfy a certain equation. Namely[2], let $\eta_{R^2([\![A]\!])}$ and $R^2(\eta_{[\![A]\!]})$ be the two canonical maps from $R^2([\![A]\!]) \to R^4([\![A]\!])$, i.e., (in λ-calculus notation)

$$
\begin{array}{l}
\eta_{R^2([\![A]\!])}(\varphi) = \\
\qquad \lambda\Phi\colon R^3([\![A]\!]).\Phi(\varphi)
\end{array}
$$

$$
\begin{array}{l}
R^2(\eta_{[\![A]\!]})(\varphi) = \\
\qquad \lambda\Phi\colon R^3([\![A]\!]).\varphi(\lambda x\colon [\![A]\!].\Phi(\lambda k\colon R([\![A]\!]).kx))
\end{array}
$$

[2]We follow Paul Taylor in using the notation $R^n(X)$ for an R-tower of height n, so $R^2(X)$ stands for R^{R^X}.

Then, since $\eta_{R^2([\![A]\!])} \circ \eta_{[\![A]\!]} = R^2(\eta_{[\![A]\!]}) \circ \eta_{[\![A]\!]}$ by λ-calculus, we have $\eta_{R^2([\![A]\!])} \circ h = R^2(\eta_{[\![A]\!]}) \circ h$, whenever h factors through $\eta_{[\![A]\!]}$, i.e., can be written in the form $\eta_{[\![A]\!]} \circ h'$. We can also argue element-wise and conclude that if $F : R^2([\![A]\!])$ is of the form $\eta_{[\![A]\!]}(a)$ for some $a : A$, i.e., $F = \lambda k.ka$, then

$$(\dagger) \qquad \begin{array}{c} \Phi(F) = F(\lambda x{:}\, [\![A]\!].\Phi(\eta_{[\![A]\!]}(x))) \\ \text{for all } \Phi{:}\, R^3([\![A]\!]) \end{array}$$

So we are led to decree that a morphism from Δ to A is a λ_μ-term $\cdot \mid \Delta \vdash t : A \Rightarrow \bot$ satisfying this property (which, of course, has to be translated into λ_μ-equations). The morphisms into contexts rather than merely types are then constructed as tuples of these.

Unfortunately, since $\eta_{[\![A]\!]}$ is in general not the equaliser of $\eta_{R^2([\![A]\!])}, R^2(\eta_{[\![A]\!]})$ this equational condition is necessary, but not sufficient for factoring through $\eta_{[\![A]\!]}$. Therefore, the category thus obtained need not be equivalent to a possibly already existing (or hypothetically assumed) category $\mathbf{C}$, but fortunately does the job, nevertheless.

In the sequel we will carry out the construction of this syntactic category explicitly and demonstrate that it meets the requirement of the main theorem. Surprisingly, the main effort consists of showing that the C-maps (as we will call them) compose and thus form a category at all. The reason is that composition cannot be defined as syntactic substitution.

Convention 4.2 *In order to avoid messy case analyses we shall henceforth adopt the convention that lower case Greek letters range over continuation variables as well as $\bot$. We extend μ-abstraction and μ-application by the settings $\mu\bot.M := M$ and $[\bot]M = M$. It is clear that this preserves typing and that equations involving μ and $[-]-$ generalise accordingly.*

With this convention the two parts of rule (μ-ζ) can be subsumed under the first one. Indeed, if we formally extend the definition of mixed substitution to $\beta : \bot$ then under the above convention the terms $t[s{::}\, \beta \,/\, \alpha]$ and $t[s{::}\, \star \,/\, \alpha]$ are identical.

4.1 The generic continuation category

For $\alpha : A$ let v_α stand for the term $\lambda x{:}\, A.[\alpha]x$.

Definition 4.3 Let Δ be a continuation context and A be a type. A *continuation term* (*C-term* for short) in context Δ and of type A is a λ_μ-term $\cdot \mid \Delta \vdash t : A \Rightarrow \bot$ such that for every λ_μ-term $\Gamma \mid \Delta, \Delta' \vdash o : (A \Rightarrow \bot) \Rightarrow \bot$ (subsequently called an *observer*) we have

$$\Gamma \mid \Delta, \Delta' \vdash ot = t(\mu\alpha{:}\, A.o(\mathsf{v}_\alpha)) : \bot$$

More generally, a *continuation map* (*C-map*) from Δ to $\Theta \equiv \alpha_1{:}\, A_1, \ldots, \alpha_n{:}\, A_n$ is an n-tuple $(t_1, \ldots, t_n)$ such that t_i is a C-term of type A_i in context Δ. $\Diamond$

We remark that t is a C-term of type A iff for each $\vec{\alpha}{:}\, [\![\Delta]\!]$ the element $\lambda d{:}\, R^{[\![A]\!]}.[\![t]\!](\cdot \mid \vec{\alpha})\langle d, \star\rangle$ (that is $[\![t]\!](\cdot \mid \vec{\alpha})$ transported along the isomorphism $R^{R^{[\![A]\!]} \times 1} \cong R^2([\![A]\!])$) satisfies condition ($\dagger$) above.

Notice that by Convention 4.2 a term $t : \bot \Rightarrow \bot$ is a C-term of type $\bot$ iff

$$\Gamma \mid \Delta, \Delta' \vdash ot = t(o(\lambda x{:}\, \bot.x)) : \bot$$

for every observer $\Gamma \mid \Delta, \Delta' \vdash o : (\bot \Rightarrow \bot) \Rightarrow \bot$.

The desired continuation category $\mathbf{C}$ will have the continuation contexts as objects and the C-maps as morphisms. In order to define composition of C-maps we need to develop some machinery first.

Proposition 4.4 *If $\alpha{:}\,A$ is a continuation variable (declared in Δ) then the term $\mathsf{v}_\alpha := \lambda x{:}\,A.[\alpha]x$ is a C-term of type A.*

Proof. Assume $\Gamma \mid \Delta, \Delta' \vdash o : (A \Rightarrow \bot) \Rightarrow \bot$. We must show that (in context $\Gamma \mid \Delta, \Delta'$) we have $o\mathsf{v}_\alpha = \mathsf{v}_\alpha(\mu\alpha{:}\,A.o(\mathsf{v}_\alpha))$. By (β) the right hand side equals $[\alpha](\mu\alpha{:}\,A.o(\mathsf{v}_\alpha))$ which equals the left hand side by $(\mu\text{-}\beta)$. □

Henceforth, we will use capital letters M, N, U, V to range over C-terms. The following characterises C-terms of negated type.

Lemma 4.5 *The C-terms of type $(A \Rightarrow \bot)$ are precisely those of the form $\lambda f{:}\,A \Rightarrow \bot.ft$ for arbitrary term $t : A$. In particular, if M is a C-term of type $A \Rightarrow \bot$ then*

$$M = \lambda f{:}\,A \Rightarrow \bot.f(\mu\alpha{:}\,A.M(\mathsf{v}_\alpha))$$

Proof. To prove the equation let $f{:}\,A \Rightarrow \bot$ be a fresh variable. We calculate as follows. Employing the observer $o = \lambda m.f(\mu\alpha{:}\,A.m(\mathsf{v}_\alpha))$ we can rewrite the body of the left-hand side as $M(\mu\varphi{:}\,A \Rightarrow \bot.o(\mathsf{v}_\varphi))$ which equals $M(\lambda x{:}\,A.f(\mu\alpha.[\alpha]x))$ by $(\mu\text{-}\zeta)$ and finally Mf by $(\mu\text{-}\eta)$ and (η). We conclude by abstracting from f.

In order to prove that $\lambda f.ft$ is a C-map let o be an observer for $\lambda f.ft$. Then we have

$$\begin{array}{lll} & (\lambda f.ft)(\mu\varphi{:}\,A \Rightarrow \bot.o(\mathsf{v}_\varphi)) & \\ = & (\mu\varphi.o(\mathsf{v}_\varphi))t & \text{by } (\mu\text{-}\zeta)\text{and } (\beta) \\ = & o(\lambda f.ft) & \end{array}$$

so $\lambda f.ft$ is a C-term. □

Lemma 4.6 *If $M : A \Rightarrow \bot$ is a C-term of type A and $t : \bot$ does not contain the continuation variable $\alpha : A$ then $M(\mu\alpha{:}\,A.t) = t$.*

Proof. We use the observer $o = \lambda m.t$. So $oM = t$ by (β) and $M(\mu\alpha.o(\mathsf{v}_\alpha)) = M(\mu\alpha.t)$ as required. □

Definition 4.7 Let t be a term of type $B \Rightarrow \bot$ possibly containing a free continuation variable $\alpha : A$ and s be a term of type $A \Rightarrow \bot$. The term $t[\alpha{:=}\,s]$ of type $B \Rightarrow \bot$ is defined as $\lambda x{:}\,B.s(\mu\alpha{:}\,A.tx)$. ◊

Proposition 4.8 *If M is a C-term of type B (possibly containing $\alpha{:}\,A$) and N is a C-term of type A then $M[\alpha{:=}\,N]$ is a C-term of type B.*

Proof. Let $o : (B \Rightarrow \bot) \Rightarrow \bot$ be an observer. We demonstrate the required identity $o(M[\alpha{:=}\,N]) = M[\alpha{:=}\,N](\mu\beta{:}\,B.o(\mathsf{v}_\beta))$ by showing that both sides equal $N(\mu\alpha.o(M))$.

Using the observer

$$o' = \lambda n{:}\,A \Rightarrow \bot.o(\lambda x{:}\,B.n(\mu\alpha{:}\,A.Mx))$$

for the C-term N we obtain

$$(o'N =)o(M[\alpha{:=}\,N]) = N(\mu\alpha{:}\,A.o'(\mathsf{v}_\alpha)) = N(\mu\alpha.o(M))$$

where the last equality involves $(\mu\text{-}\beta)$. On the other hand, $M[\alpha{:=}\,N](\mu\beta{:}\,B.o(\mathsf{v}_\beta))$ equals $N(\mu\alpha.o(M))$ using the observer o applied to M. □

We will now show that the operations on C-terms v_{-} and $-[-{:=}-]$ behave like variables and substitution in ordinary λ-calculus.

Lemma 4.9 *The following equations hold whenever they are well-typed and M, N, U, V are C-terms as indicated.*

1. $\mathsf{v}_\alpha[\alpha{:=}M] = M$.
2. $\mathsf{v}_\beta[\alpha{:=}N] = \mathsf{v}_\beta$, *if* $\alpha \not\equiv \beta$.
3. $M[\alpha{:=}\mathsf{v}_\alpha] = M$.
4. $M[\alpha{:=}U] = M$, *if* α *not free in* M.
5. $M[\alpha{:=}U][\beta{:=}V] = M[\beta{:=}V][\alpha{:=}U[\beta{:=}V]]$, *if* α *is not free in* V.
6. $M[\alpha{:=}U][\beta{:=}V] = M[\beta{:=}V][\alpha{:=}U]$ *if* α, β *are not free in* U, V.
7. $M[\alpha{:=}U][\beta{:=}V] = M[\alpha{:=}U[\beta{:=}V]]$, *if* β *not free in* M.

Proof. First, observe that if $\alpha{:}\,A$ is a continuation variable and M is any term of type $\perp$ then $\mathsf{v}_\alpha(\mu\alpha{:}\,A.M) = M$ by (β) and $(\mu\text{-}\beta)$.

We will omit type annotations as they depend upon the unspecified typing of the equations.

Ad 1. $\mathsf{v}_\alpha[\alpha{:=}M] = \lambda x.M(\mu\alpha.\mathsf{v}_\alpha x) = \lambda x.Mx = M$ by $(\mu\text{-}\eta)$ and (η).

Ad 3. $M[\alpha{:=}\mathsf{v}_\alpha] = \lambda x.\mathsf{v}_\alpha(\mu\alpha.Mx) = M$ by the above observation followed by $(\mu\text{-}\beta)$ and (η).

Ad 5. The right hand side expands to $\lambda x.o(V)$ where

$$o = \lambda v.v(\mu\beta.U(\mu\alpha.v(\mu\beta.Mx)))$$

Since V is a C-term this equals

$$\lambda x.V(\mu\beta.\mathsf{v}_\beta(\mu\beta.U(\mu\alpha.\mathsf{v}_\beta(\mu\beta.Mx))))$$

By the above observation this equals $\lambda x.V(\mu\beta.U(\mu\alpha.Mx))$ which is a (β)-contraction of the left hand side.

Ad 4. By definition $M[\alpha{:=}U] = \lambda x.U(\mu\alpha.Mx)$. By Lemma 4.6 applied to U and $t := Mx$ this equals $\lambda x.Mx$. The conclusion follows by (η).

The remaining parts 2, 6, 7 are immediate consequences of 5 and 4. $\square$

This allows us to carry out the usual construction [10] of a category with finite products from a substitution calculus. Let M be a C-term of type A in context $\Delta \equiv \alpha_1{:}\,A_1, \ldots, \alpha_n{:}\,A_n$. Furthermore, let $f \equiv (N_1, \ldots, N_n)$ be a C-map from Θ to Δ. We define the C-term $M \circ f$ of type A in context Θ as $M[\alpha_1{:=}N_1]\ldots[\alpha_n{:=}N_n]$.

More generally, if $g = (M_1, \ldots, M_m)$ is a C-map from Δ to Ψ, we define the composition $g \circ f$ as $(M_1 \circ f, \ldots, M_m \circ f)$. Finally, the identity C-map $\mathrm{id}_\Delta : \Delta \to \Delta$ is defined as $(\mathsf{v}_{\alpha_1}, \ldots, \mathsf{v}_{\alpha_n})$.

The proof [10] that the term model of an equational theory forms a category can now be copied word for word so as to demonstrate that the continuation contexts and ($\mathcal{E}$-equivalence classes of) C-maps form a category $\mathbf{C}$. Following common practice, we will refer to morphisms in $\mathbf{C}$ via representatives, i.e., C-maps.

The continuation contexts of length one which we will henceforth identify with types form the subset $\mathbf{T}$ of $\mathbf{C}$. It also follows from this proof that the empty context $[]$ forms a terminal object and that the extended context $\Delta, \alpha{:}\,A$ forms a cartesian product of Δ and type A if $A \neq \perp$. We define the cartesian product $\Delta \cdot \perp$ as Δ.

The projection on $\perp$ is given by $\mathsf{v}_\perp$. By abuse of notation we decree that $\Delta, \alpha{:}\, \perp$ means Δ.

The type object of responses R is defined as the type $\perp \Rightarrow \perp$. This setting is motivated by the observation that the meaning of type $\perp \Rightarrow \perp$ in an arbitrary continuation category equals $R^1 \times 1$ which is isomorphic to R.

Next, we show that C-maps of type R are precisely the terms of the form $\lambda f.M$ where f is not free in M.

Proposition 4.10 *Let A be a type,* i.e. *a type object of* $\mathbf{C}$. *The type $A \Rightarrow \perp$ is an exponential of R by A with evaluation map ev_A given by*

$$\cdot \mid \varphi{:}\, A \Rightarrow \perp, \alpha{:}\, A \vdash \lambda f{:}\, \perp \Rightarrow \perp. f([\varphi](\mathsf{v}_\alpha)) : R \Rightarrow \perp$$

Proof. First note that ev_A is a C-map by Lemma 4.5.

We have to show that the operation $uncur_A : \mathbf{C}(\Delta, R^A) \to \mathbf{C}(\Delta \cdot A, R)$ sending a C-map $\cdot \mid \Delta \vdash M : (A \Rightarrow \perp) \Rightarrow \perp$ to

$$ev_A[\varphi{:=}\, M] = \lambda f{:}\, \perp \Rightarrow \perp. M(\lambda x{:}\, A. f([\alpha]x)) \quad \text{by } (\mu\text{-}\zeta)$$

is a bijection.

The candidate for the inverse to *uncur* sends a C-map $\cdot \mid \Delta, \alpha{:}\, A \vdash N : (\perp \Rightarrow \perp) \Rightarrow \perp$, i.e. a $\mathbf{C}$-morphism from $\Delta \cdot A$ to R to the C-map $cur_{\alpha:A}(N)$ given by

$$\cdot \mid \Delta \vdash \lambda f{:}\, A \Rightarrow \perp. f(\mu\alpha{:}\, A. N(\mathsf{v}_\perp)) : (A \Rightarrow \perp) \Rightarrow \perp$$

Notice that $cur_{\alpha:A}(t)$ binds α in t. Assume a C-map $\cdot \mid \Delta, \alpha{:}\, A \vdash N : (\perp \Rightarrow \perp) \Rightarrow \perp$. The expression $uncur(cur(N))$ expands to

$$\lambda f{:}\, \perp \Rightarrow \perp. f(N(\lambda x.x))$$

This equals N by Lemma 4.5.

For the other direction assume a C-map $\cdot \mid \Delta \vdash M : (A \Rightarrow \perp) \Rightarrow \perp$. The required equation $cur(uncur(M)) = M$ is after (β)-contraction an instance of Lemma 4.5

□

Proposition 4.11 *Let A, B be types. The type $A \Rightarrow B$ is a cartesian product of $R^A (= A \Rightarrow \perp)$ and B with projections $\pi \in \mathbf{C}(A \Rightarrow B, R^A)$ and $\pi' \in \mathbf{C}(A \Rightarrow B, B)$ given by*

$$\cdot \mid \varphi{:}\, A \Rightarrow B \vdash \lambda f{:}\, A \Rightarrow \perp. f(\mu\alpha{:}\, A. [\varphi](\lambda x{:}\, A. \mu\beta{:}\, B. [\alpha]x)) : (A \Rightarrow \perp) \Rightarrow \perp$$

and $\cdot \mid \varphi{:}\, A \Rightarrow B \vdash \lambda b{:}\, B. [\varphi](\lambda x{:}\, A. b) : B \Rightarrow \perp$.

Proof. The first projection is a C-map by Lemma 4.5. To see that the second projection is a C-map let o be an observer. We calculate as follows.

$$\begin{array}{lll} & o(\pi') & \\ = & o(\lambda b{:}\, B. [\varphi](\lambda x{:}\, A. b)) & \\ = & [\varphi]\mu\psi{:}\, A \Rightarrow B. o(\lambda b. [\psi](\lambda x. b)) & \text{by } (\mu\text{-}\beta)\text{-expansion} \\ = & [\varphi]\lambda x{:}\, A. \mu\beta{:}\, B. o(\lambda b. [\beta]b) & \text{by } (\mu\text{-}\zeta) \text{and } (\beta) \\ = & \pi'(\mu\beta. o(\mathsf{v}_\beta)) & \end{array}$$

Let $\cdot \mid \Delta \vdash M : R^A \Rightarrow \perp$ and $\cdot \mid \Delta \vdash N : B \Rightarrow \perp$ be C-terms of type R^A and B, resp. In view of Lemma 4.5 we can write M as $\lambda f{:}\, R^A. ft$ for some term $\cdot \mid \Delta \vdash t : A$. We define the C-term $\langle M, N\rangle$ of type $A \Rightarrow B$ in context Δ as $\lambda f\colon A \Rightarrow B.N(ft)$. To see that this is a C-term assume an observer $o : ((A \Rightarrow B) \Rightarrow \bot) \Rightarrow \bot$. Now $o(\langle M, N\rangle)$ equals $u := N(\mu\beta\colon B.o(\lambda f\colon A \Rightarrow B.[\beta](ft)))$ using the fact that N is a C-term and its surrounding context as observer. To show that $\langle M, N\rangle(\mu\varphi\colon A \Rightarrow B.o(\mathsf{v}_\varphi))$ equals u one uses (μ-ζ) on φ and (β)-steps.

Now we have $\pi \circ \langle M, N\rangle =_{\mathrm{def}}= \pi[\varphi{:=}\langle M, N\rangle] = M$ by (β) and (μ-ζ) and Lemma 4.6. Similarly, we have $\langle M, N\rangle[\varphi{:=}\pi'] = N$ using (μ-η) and (η).

For the uniqueness of pairing assume $\cdot \mid \Delta \vdash P : (A \Rightarrow B) \Rightarrow \bot$ be a C-term of type $R^A \times B$. We must show that $\langle \pi[\varphi{:=}P], \pi'[\varphi{:=}P]\rangle f = Pf$ where $f : A \Rightarrow B$ is a fresh variable. Writing the left hand side as $o(P)$ we can rewrite it to $P(\mu\varphi\colon A \Rightarrow B.\langle \pi[\varphi{:=}\mathsf{v}_\varphi], \pi'[\varphi{:=}\mathsf{v}_\varphi]\langle f)$. In view of Lemma 4.9 (1) the desired equation follows if we can demonstrate that $\mu\varphi.\langle \pi, \pi'\rangle f = f$. But this follows easily using (μ-ζ) on φ, (μ-η), and (η). □

4.1.1 Interpretation of types and contexts

We have thus shown that **C** with the described settings furnishes a continuation category. Interpreting base types by themselves we obtain immediately

Proposition 4.12 *For any λ_μ-type X the interpretation $[\![X]\!]$ in* **C** *equals X.*

Proof. Immediate from $[\![A \Rightarrow B]\!] = R^{[\![A]\!]} \times [\![B]\!] = [\![A]\!] \Rightarrow [\![B]\!]$ and $[\![\bot]\!] = \bot$. □

We will henceforth often make implicit use of this proposition by omitting semantic brackets around types.

Accordingly, the semantics of a continuation context Δ can be chosen as Δ itself (the choice only affects the names of continuation variables). Next, we examine the interpretation of combined contexts $\Gamma \mid \Delta$ where $\Gamma \equiv x_1\colon A_1, \ldots, x_n\colon A_n$ is an object context. Its interpretation $R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!]$ is the following continuation context

$$\varphi_1\colon A_1 \Rightarrow \bot, \ldots, \varphi_n\colon A_n \Rightarrow \bot, \Delta$$

where the φ_i are freshly chosen continuation variables. We introduce the notation φ_x for the continuation variable in $R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!]$ corresponding to object variable x in Γ, in other words we have $\varphi_{x_i} = \varphi_i$. If $\vec{x}$ is a sequence of object variables we write $\varphi_{\vec{x}}$ for the corresponding sequence of continuation variables.

4.1.2 Interpretation of terms

If $c\colon A$ is a constant then $[\![c]\!] := \lambda f\colon A \Rightarrow \bot.fc$ is a C-term of type $R^{[\![A]\!]}$ yielding an interpretation for the constants. We thus obtain an interpretation of our λ_μ-calculus in the continuation category **C** which associates with every λ_μ-term $\Gamma \mid \Delta \vdash t : A$ a C-term $[\![t]\!]$ of type $R^{[\![A]\!]} = A \Rightarrow \bot$ in context $R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!] = \varphi_1\colon A_1 \Rightarrow \bot, \ldots, \varphi_n\colon A_n \Rightarrow \bot, \Delta$.

Our aim is to exhibit a direct relationship between $[\![t]\!]$ and t. To that end we first introduce the following notation. If $\varphi\colon A \Rightarrow \bot$ is a continuation variable then $\overline{\varphi} := \mu\alpha\colon A.[\varphi]\mathsf{v}_\alpha$ is a λ_μ-term of type A. If t is an λ_μ-term of type B containing the object variable $x\colon A$ then we can form the λ_μ-term $t[\overline{\varphi}/x]$ also of type B which does not contain x anymore but φ instead. More generally, we write $t[\overline{\vec{\varphi}}/\vec{x}]$ for $t[\overline{\varphi_1}/x_1] \ldots [\overline{\varphi_n}/x_n]$. Thus, in particular, if $\Gamma \mid \Delta \vdash t : B$ is a λ_μ-term and $\vec{x}$ is the sequence of variables in Γ then we have

$$R^{[\![\Gamma]\!]} \cdot \Delta \vdash t[\overline{\varphi_{\vec{x}}} \,/\, \vec{x}] : B$$

Lemma 4.13 *Let t be a λ_μ-term of type $\bot$ containing object variable $x\colon A$ and let $\varphi\colon A \Rightarrow \bot$ be a fresh continuation variable. Then $\mu\varphi\colon A \Rightarrow \bot.t[\overline{\varphi}/x] = \lambda x\colon A.t$.*

Proof. We calculate as follows:

$$
\begin{array}{lll}
 & \mu\varphi.t[\overline{\varphi}/x] & \\
= & \lambda x\colon A.t[\overline{\varphi}/x][x{::}\star \,/\, \varphi] & (\mu\text{-}\zeta) \\
= & \lambda x.t[\overline{\varphi}[x{::}\star \,/\, \varphi] \,/\, x] & \text{Def. of substitution} \\
= & \lambda x.t[x/x] & (\mu\text{-}\eta) \\
= & \lambda x.t &
\end{array}
$$

□

We are now ready to state the desired relationship

Theorem 4.14 *Whenever $\Gamma \,|\, \Delta \vdash t : A$ then $[\![\Gamma \,|\, \Delta \vdash t : A]\!]$ is equal, w.r.t. $\mathcal{E}$, to the following term*

$$R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!] \vdash \lambda f\colon A \Rightarrow \bot.f(t[\overline{\varphi_{\vec{x}}} \,/\, \vec{x}]) : R^A \Rightarrow \bot$$

where $t[\overline{\varphi_{\vec{x}}} \,/\, \vec{x}]$ denotes the simultaneous substitution of $\overline{\varphi_x}$ for x in t for every object variable x in Γ.

Proof. For the proof it is appropriate to make explicit the meanings in $\mathbf{C}$ of the informal metalanguage used in the definition of the semantics of λ_μ in an arbitrary continuation category.

In addition to the already defined combinators for abstraction and pairing (*cur* and $\langle -,-\rangle$) we need a combinator for application (defined from *uncur*) and projection (defined by composition from the projection morphisms π, π'). The definition of these combinators on pseudoterms is as follows.

$$
\begin{array}{l}
app(M,N) = ev[\varphi{:=}\, M][\alpha{:=}\, N] = \lambda f.N(\mu\alpha.M(\lambda x.f([\alpha]x))) \\
cur_{\alpha:A}(M) = \lambda f\colon A \Rightarrow \bot.f(\mu\alpha\colon A.M(\mathsf{v}_\bot)) \\
\langle \lambda f\colon R^A.ft, N\rangle = \lambda f\colon A \Rightarrow B.N(ft) \\
M.1 = \pi[\varphi{:=}\, M] = \lambda f\colon R^A.M(\lambda x\colon A.\mu\beta\colon B.fx) \\
M.2 = \lambda b\colon B.M(\lambda x\colon A.b) \\
\star = \mathsf{v}_\bot
\end{array}
$$

Now we will prove the theorem by induction on the structure of t. In the course of the calculations we omit contexts and those continuation variables which are not explicitly introduced.

Case $t = c$. Immediate from the definition.

Case $t = x\colon A$. We have

$$
\begin{array}{lll}
 & [\![x]\!] & \\
= & \mathsf{v}_{\varphi_x} & \\
= & \lambda f\colon A \Rightarrow \bot.[\varphi]f & \\
= & \lambda f.f(\mu\alpha\colon A.[\varphi]\mathsf{v}_\alpha) & \text{by Lemma 4.5} \\
= & \lambda f.f(\overline{\varphi_x}) & \\
= & \lambda f.f(x[\overline{\varphi_x} \,/\, x]) &
\end{array}
$$

Case $t = \lambda x{:}\, A.s$**.** We write $\tilde{s}$ for $s[\overline{\varphi_{\vec{x}}} \,/\, \vec{x}]$ where the $\vec{x}$ are the free object variables of t. Thus the induction hypothesis reads $[\![s]\!] = \lambda f{:}\, B \Rightarrow \bot.f(\tilde{s}[\overline{\varphi_x}/x])$. Now we calculate as follows.

$$
\begin{array}{rll}
& [\![\lambda x{:}\, A.s]\!] & \\
= & cur_{\psi: R^A \times B}(app([\![s]\!][\varphi_x := \mathsf{v}_\psi.1]\,,\, \mathsf{v}_\psi.2)) & \text{by definition of } [\![\lambda x.s]\!] \\
= & \lambda f{:}\,(A \Rightarrow B) \Rightarrow \bot.f(\mu\psi{:}\, A \Rightarrow B.app([\![s]\!][\varphi_x := \mathsf{v}_\psi.1]\,,\, \mathsf{v}_\psi.2)\mathsf{v}_\bot) & \text{expansion of } cur \\
= & \lambda f.f(\mu\psi.\mathsf{v}_\psi.2(\mu\beta{:}\, B.[\![s]\!][\varphi_x := \mathsf{v}_\psi.1]\mathsf{v}_\beta)) & \text{expansion of } app \\
= & \lambda f.f(\mu\psi.\mathsf{v}_\psi.2(\mu\beta.\mathsf{v}_\psi.1(\mu\varphi_x.[\![s]\!]\mathsf{v}_\beta))) & \text{expansion of } [\![s]\!][\varphi_x := \mathsf{v}_\psi.1] \\
= & \lambda f.f(\mu\psi.\mathsf{v}_\psi.2(\mu\beta.\mathsf{v}_\psi.1(\mu\varphi_x.[\beta]\tilde{s}[\overline{\varphi_x}/x]))) & \text{induction hypothesis} \\
= & \lambda f.f(\mu\psi.\mathsf{v}_\psi.2(\mu\beta.\mathsf{v}_\psi.1(\lambda x.[\beta]\tilde{s}))) & \text{by Lemma 4.13} \\
= & \lambda f.f(\mu\psi.\mathsf{v}_\psi.2(\mu\beta.[\psi]\lambda x.\mu\beta'.[\beta]\tilde{s})) & \text{expansion of } \mathsf{v}_\psi.1 \\
= & \lambda f.f(\mu\psi.[\psi](\lambda x{:}\, A.\mu\beta.[\psi]\lambda x.\mu\beta'.[\beta]\tilde{s})) & \text{expansion of } \mathsf{v}_\psi.2 \\
= & \lambda f.f(\lambda x{:}\, A.\mu\beta{:}\, B.[\beta]\tilde{s}) & (\mu\text{-}\zeta) \text{ on } \psi,\ (\mu\text{-}\beta) \\
= & \lambda f.f(\lambda x.\tilde{s}) & (\mu\text{-}\eta) \\
= & \lambda f.f(\lambda x.s[\overline{\varphi_{\vec{x}}}/\vec{x}]) & \\
= & \lambda f.f(t[\overline{\varphi_{\vec{x}}}/\vec{x}]) &
\end{array}
$$

Case $t = s_1 s_2$**.** Again, we write $\tilde{s}_i$ for $s_i[\overline{\varphi_{\vec{x}}}/\vec{x}]$.

$$
\begin{array}{rll}
& [\![s_1 s_2]\!] & \\
= & cur_{\beta: B}(app([\![s_1]\!], \langle [\![s_2]\!], \mathsf{v}_\beta\rangle)) & \\
= & \lambda f.f(\mu\beta{:}\, B.\langle [\![s_2]\!], \mathsf{v}_\beta\rangle(\mu\psi{:}\, A \Rightarrow B.[\![s_1]\!]\mathsf{v}_\psi)) & \\
= & \lambda f.f(\mu\beta.\langle \lambda f.f\tilde{s}_2, \mathsf{v}_\beta\rangle(\mu\psi.[\![s_1]\!]\mathsf{v}_\psi)) & \text{induction hypothesis} \\
= & \lambda f.f(\mu\beta.\mathsf{v}_\beta((\mu\psi.[\![s_1]\!]\mathsf{v}_\psi)\tilde{s}_2)) & \text{expansion of } \langle -, - \rangle \\
= & \lambda f.f((\mu\psi.[\![s_1]\!]\mathsf{v}_\psi)\tilde{s}_2) & (\mu\text{-}\eta) \\
= & \lambda f.f((\mu\psi.[\psi]\tilde{s}_1)\tilde{s}_2) & \\
= & \lambda f.f(\tilde{s}_1\tilde{s}_2) &
\end{array}
$$

Case $t = \mu\alpha.s$**.** We write $\tilde{s}$ for $s[\overline{\varphi_{\vec{x}}}/\vec{x}]$.

$$
\begin{array}{rl}
& [\![\mu\alpha{:}\, A.s]\!] \\
= & cur_{\alpha: A}(app([\![s]\!], \star)) \\
= & \lambda f.f(\mu\alpha.app([\![s]\!], \star)\mathsf{v}_\bot) \\
= & \lambda f.f(\mu\alpha.\star([\![s]\!](\mathsf{v}_\bot))) \\
= & \lambda f.f(\mu\alpha.\tilde{s})
\end{array}
$$

Case $t = [\alpha]s$**.** We write $\tilde{s}$ for $s[\overline{\varphi_{\vec{x}}}/\vec{x}]$.

$$
\begin{array}{rl}
& [\![[\alpha]s]\!] \\
= & cur_\bot(app([\![s]\!], \mathsf{v}_\alpha)) \\
= & \lambda f.f(\mathsf{v}_\alpha(\mu\alpha.[\![s]\!]\mathsf{v}_\alpha)) \\
= & \lambda f.f([\![s]\!]\mathsf{v}_\alpha) \\
= & \lambda f.f([\alpha]\tilde{s})
\end{array}
$$

$\square$

This syntactic characterisation of the interpretation of λ_μ in $\mathbf{C}$ now enables us to quickly conclude the main result.

Proof of Theorem 4.1. For the first part assume that $\Gamma \mid \Delta \vdash t_i : A$ for $i = 1,2$ are two terms of λ_μ with equal semantics in the generic continuation category $\mathbf{C}$ constructed from $\mathcal{E}$. By Theorem 4.14 this implies that we have

$$\lambda f{:}\, A \Rightarrow \bot . f t_1[\overline{\varphi_{\vec{x}}}/\vec{x}] = \lambda f{:}\, A \Rightarrow \bot . f t_2[\overline{\varphi_{\vec{x}}}/\vec{x}]$$

Introducing a fresh variable $f{:}\, A \Rightarrow \bot$ we get

$$f t_1[\overline{\varphi_{\vec{x}}}/\vec{x}] = f t_2[\overline{\varphi_{\vec{x}}}/\vec{x}]$$

Using Lemma 4.13 iteratively on all the continuation variables in $\varphi_{\vec{x}}$ we obtain

$$\mu\varphi_n \ldots (\mu\varphi_2.(\mu\varphi_1.f t_i[\overline{\varphi_{\vec{x}}}/\vec{x}])x_1)x_2) \ldots x_n = f t_i$$

for $i = 1,2$. Therefore, by congruence, we obtain $f t_1 = f t_2$. Now, if $\alpha{:}\, A$ is a fresh continuation variable we can replace f by v_α (formally by λ-abstracting f and using (β)). This yields $[\alpha]t_1 = [\alpha]t_2$ from which $t_1 = t_2$ follows by $(\mu$-$\eta)$.

For the second part assume that F is a $\mathbf{C}$-morphism from $R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!]$ to $R^{[\![A]\!]}$. By definition of $\mathbf{C}$ this means that F is a C-map of type $A \Rightarrow \bot$ in context $R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!]$. In view of Lemma 4.5 this means that F takes the form

$$R^{[\![\Gamma]\!]} \cdot [\![\Delta]\!] \vdash \lambda f{:}\, A \Rightarrow \bot . f t : (A \Rightarrow \bot) \Rightarrow \bot$$

for some λ_μ-term t of type A. From t we construct another term $\tilde{t}$ of type A in context $\Gamma \mid \Delta$ as

$$\tilde{t} := t[x_1 {::} \star \,/\, \varphi_1][x_2 {::} \star \,/\, \varphi_2] \ldots [x_n {::} \star \,/\, \varphi_n]$$

We claim that $F = [\![\Gamma \mid \Delta \vdash \tilde{t}]\!]$. By Theorem 4.14 this is equivalent to demonstrating that

$$\lambda f. f t = \lambda f. f \tilde{t}[\overline{\varphi_{\vec{x}}}/\vec{x}]$$

For this it is sufficient to show that

$$t = \tilde{t}[\overline{\varphi_{\vec{x}}}/\vec{x}]$$

which is immediate by n-fold application of the following sublemma.

Sublemma If $s : C$ is a λ_μ-term containing continuation variable $\varphi{:}\, B \Rightarrow \bot$ and $x{:}\, B$ is a fresh object variable then $s[x {::} \star \,/\, \varphi][\overline{\varphi}/x] = s$.

Proof of sublemma. If $C \equiv \bot$ then $s[x {::} \star \,/\, \varphi][\overline{\varphi}/x]$ equals $[\varphi](\lambda x{:}\, B.s[x {::} \star \,/\, \varphi])$ by applying v_φ to an instance of Lemma 4.13. By $(\mu$-$\zeta)$ this equals $[\varphi]\mu\varphi.s$, thus s by $(\mu$-$\beta)$. If $C \not\equiv \bot$ then we expand s as $\mu\gamma{:}\, C.[\gamma]s$ and apply the previous case to $[\gamma]s$. □

This completes the proof of the main result.

4.2 Representation of Ong's categorical models

In [8] is defined a categorical semantics of λ_μ called λ_μ-categories. By the categorical completeness result of loc. cit. these models are in 1-1-correspondence with λ_μ-theories. Thus, our results can be stated in category-theoretic terms as follows. The soundness theorem (Thm. 3.3) says that every category of continuations can be organised into a λ_μ-category with the same objects and whose homsets $\mathbb{E}_\Delta((A_1, \ldots, A_n), B)$ are given by $\mathbf{C}(\Delta \cdot R^{A_1} \cdot \ldots \cdot R^{A_n}, R^B)$. The completeness theorem (Thm. 4.1), on the other hand, expresses that every λ_μ-category is (up to isomorphism) of this form. Namely, by Ong's categorical completeness result every λ_μ-category is isomorphic to the term model of its theory which in turn (by our main result) is isomorphic to the λ_μ- category induced by the associated syntactic category of continuations.

Thus, every model of λ_μ, i.e. any λ_μ-category is isomorphic to a continuation model. We have preferred to stick to the more traditional syntactic presentation of our results as this simplifies the calculations. It might, however, be instructive to explicitly compute the continuation model which induces the game-theoretic λ_μ-category described in loc. cit.

4.3 Completeness of λ_μ for CPS-translation

Of particular interest is the "free" λ_μ-calculus over some signature $(\mathcal{B}, \mathcal{K})$ without non-logical axioms. It can be interpreted in the (cartesian closed category associated with) simply-typed lambda calculus with products and terminal object over base types $\mathcal{B} \cup \{R\}$ and constants of appropriate type. In this particular case the interpretation gives rise to the following CPS translation of λ_μ.

$$
\begin{aligned}
&[\![x]\!] = x \\
&[\![c]\!] = c \\
&[\![\lambda x.t]\!] = \lambda p.((\lambda x.[\![t]\!])\, p.1)p.2 \\
&[\![ts]\!] = \lambda\beta.[\![t]\!]\langle[\![s]\!], \beta\rangle \\
&[\![\mu\alpha.t]\!] = \lambda\alpha.[\![t]\!]\star \\
&[\![[\alpha]t]\!] = \lambda x\colon 1.[\![t]\!]\alpha
\end{aligned}
$$

These clauses are derived by instantiating the defining clauses in Section 3 by the term model of simply typed lambda calculus where we keep using the same name for a syntactic and semantic variables. This particular semantics is already complete for free λ_μ.

Proposition 4.15 *We have* $\Gamma \mid \Delta \vdash t_1 = t_2 : A$ *if and only if* $[\![t_1]\!] = [\![t_2]\!]$.

Proof. If $[\![t_1]\!] = [\![t_2]\!]$ then t_1 and t_2 have equal interpretation in any continuation category because simply-typed lambda calculus is initial for those. Thus $\Gamma \mid \Delta \vdash t_1 = t_2 : A$ by Thm. 4.1. The other direction is immediate from soundness (Thm. 3.3). □

This result means that the free λ_μ-theory can be decided by way of the above CPS translation.

5 Call-by-value λ_μ

In [7] has been defined a call-by-value version of λ_μ which is validated by the usual continuation semantics for call-by-value lambda calculus as described e.g. in [4].

Our aim in this section is to establish a completeness result analogous to Thm 4.1 above for this system. As it happens the terms and typing rules of this call-by-value variant are exactly the same as for the call-by-name version. In order to state the equational axioms of the call-by-value version we need some syntactic machinery beforehand. The variables and λ-abstractions are called *values* and are ranged over by letters $u, v, \ldots$.

The *evaluation contexts* are inductively defined by

$$E ::= [] \mid vE \mid Et \mid [\alpha]E$$

$$(\beta_V) \qquad (\lambda x\colon A.t)v = t[v/x].$$

$$(\eta_V) \qquad \lambda x\colon A.vx = v,$$
when x is not free in v.

$$(\mu\text{-}\beta) \qquad [\alpha]\mu\gamma\colon A.t = t[\alpha/\gamma].$$

$$(\mu\text{-}\eta) \qquad \mu\alpha\colon A.[\alpha]t = t,$$
α not free in t.

$$(\mu\text{-}\zeta_V) \quad \begin{cases} E[\mu\alpha\colon A.t] = \mu\beta\colon B.t[[\beta]E \,/\, \alpha], \text{ if } B \neq \bot \\ E[\mu\alpha\colon A.t] = t[E \,/\, \alpha] \end{cases}$$

$$(\mu_\bot) \qquad v\,t = \mu\alpha\colon A.t \qquad \text{where } \alpha \neq \bot$$
$$(\mu'_\bot) \qquad v\,t = t \qquad \text{if } v : \bot \Rightarrow \bot \text{ and } t : \bot$$

$$(E) \qquad (\lambda x\colon A.E[x])t = E[t]$$

Figure 5: Equality axioms for λ_μ

We write $E[t]$ for the substitution of t for [] in evaluation context E. Note that since evaluation contexts do not involve binders no free variables in t can ever be captured in $E[t]$ so no renaming of variables in t is necessary.

Besides usual (capture-free) substitution $t[s/x]$ of a term s for a variable x in a term t we also have a substitution of evaluation contexts for continuation variables $t[E/\alpha]$ where the key clauses are

$$([\alpha]t)[E/\alpha] = E[t[E/\alpha]]$$
$$([\beta]t)[E/\alpha] = [\beta](t[E/\alpha]), \text{ if } \alpha \neq \beta$$

This is homomorphically extended to the other term formers. When substituting into a μ-abstraction capture of free continuation variables in E must be avoided by appropriate renaming.

This substitution is type correct only if the type of α and the type of the "hole" [] in E agree and, moreover, E itself is of type $\bot$.

Notice that in [7] this substitution is available only for evaluation contexts of the form $[\beta]E$ and written $t[\beta, E/\alpha]$ in this case.

Definition 5.1 A cbv-λ_μ-theory (over a signature $(\mathcal{B}, \mathcal{K})$) is a set $\mathcal{E}$ of typed equations of the form $\Gamma \,|\, \Delta \vdash s = t : A$ where $\Gamma \,|\, \Delta \vdash s : A$ and $\Gamma \,|\, \Delta \vdash t : A$ such that

- $\mathcal{E}$ is a congruence stable under weakening,
- $\mathcal{E}$ contains all well-typed instances of the basic equality laws depicted in Figure 5.

$\Diamond$

Apart from minor simplifications in notation the main difference to Ong-Stewart is that our evaluation contexts are closed under labelling (i.e. $[\alpha]E$ is an evaluation context if E is). The effect of this extension is concentrated in the following special case of axiom (E):

$$(\lambda x\colon A.[\alpha]x)t = [\alpha]t$$

Would we add this equation explicitly we could stick to the Ong-Stewart formulation. The more liberal notion of evaluation context is adopted because it is validated by the standard continuation semantics given below.

As before we will henceforth adopt Convention 4.2. Again, this allows us to formally subsume both cases of (μ-ζ_V) under the first one. Similarly, Axiom ($\mu'_\perp$) is subsumed under ($\mu_\perp$).

5.1 Continuation models for the call-by-value case

Definition 5.2 [Category of values] A *category of values* is given by the following data:

1. A category $\mathcal{V}$ with a distinguished class $\mathbf{T}$ of objects of $\mathcal{V}$ called *type objects*.
2. A distinguished type object R of *responses*.
3. For every object Γ and type object A a chosen cartesian product $\Gamma \cdot A$.
4. A chosen terminal object $[]$ (for the empty context).
5. A chosen initial object 0 (to interpret $\perp$).
6. For every type object A a chosen exponential $R^A \in \mathbf{T}$ of R by A.
7. For any two type objects A and B a chosen exponential $(R^{R^B})^A \in \mathbf{T}$ of R^{R^B} by A.

Particular examples are the category of sets (with $0 = \emptyset$) and the category of cpos with or without bottom ("predomains") with Scott-continuous maps. In this case the choice of an actual domain, i.e., with bottom element, for R guarantees the availability of a least fixpoint operator in the ensuing continuation model.

Assume a fixed signature $(\mathcal{B}, \mathcal{K})$ and a category of values $\mathcal{V}$. Any assignment of type objects $[\![B]\!]$ to base types B is extended to compound types as follows.

$$\begin{array}{rcl} [\![\perp]\!] & = & 0 \\ [\![A \Rightarrow B]\!] & = & (R^2([\![B]\!]))^{[\![A]\!]} \end{array}$$

Here the intuition is that $[\![A]\!]$ is the space of abstract values of type A. An arbitrary term of type A will not be interpreted as an element of $[\![A]\!]$ but rather as an element of $R^2([\![A]\!])$. Since $[\![\perp]\!] = 0$ is an initial object we have $R^2([\![\perp]\!]) \cong R$. Therefore, denotations of terms of type $\perp$ correspond to elements of R, i.e., responses.

Let $\Gamma \equiv x_1{:}\,A_1, \ldots, x_n{:}\,A_n$ be an object context and $\Delta \equiv \alpha_1{:}\,B_1, \ldots, \alpha_m{:}\,B_m$ be a continuation context. We use the notation $[\![\Gamma]\!] \cdot R^{[\![\Delta]\!]}$ for the object

$$[] \cdot [\![A_1]\!] \cdot \ldots \cdot [\![A_n]\!] \cdot R^{[\![B_1]\!]} \cdot \ldots \cdot R^{[\![B_m]\!]}$$

Assume an assignment of denotations to the constants, i.e. a morphism $[\![c]\!] : [] \to R^2([\![A]\!])$ if $c{:}\,A$ is in $\mathcal{K}$. To each sequent $\Gamma \,|\, \Delta \vdash t : A$ we associate a morphism

$$[\![\Gamma \,|\, \Delta \vdash t : A]\!] : [\![\Gamma]\!] \cdot R^{[\![\Delta]\!]} \to R^2([\![A]\!])$$

by the clauses in Figure 6. Again, we use an informal lambda calculus to denote constructions in $\mathcal{V}$. We write $\eta_X : X \to R^2(X)$ for the $\mathcal{V}$-morphism defined by $\eta_X(x{:}\,X) = \underline{\lambda} k{:}\,R^X.kx$. We use $?_X : 0 \to X$ for the unique morphism from the initial object 0 to X. $\Diamond$

$$
\begin{array}{lcl}
[\![\Gamma \mid \Delta \vdash x_i : A]\!](\vec{x} \mid \vec{\alpha}) & = & \eta_{[\![A]\!]}(x_i) \\
[\![\Gamma \mid \Delta \vdash \lambda x{:}\, A.t : A \Rightarrow B]\!](\vec{x} \mid \vec{\alpha}) & = & \eta_{[\![A \Rightarrow B]\!]}(\underline{\lambda} v{:}\, [\![A]\!].[\![\Gamma, x{:}\, A \mid \Delta \vdash t : B]\!](\vec{x}, x \mid \vec{\alpha})) \\
[\![\Gamma \mid \Delta \vdash ts : B]\!](\vec{x} \mid \vec{\alpha}) & = & \begin{array}{l} \underline{\lambda} k{:}\, R^{[\![B]\!]}.[\![\Gamma \mid \Delta \vdash t : A \Rightarrow B]\!](\vec{x} \mid \vec{\alpha}) \\ \quad (\underline{\lambda} f{:}\, [\![A \Rightarrow B]\!].[\![\Gamma \mid \Delta \vdash s : A]\!](\vec{x} \mid \vec{\alpha})(\underline{\lambda} v{:}\, [\![A]\!].fvk)) \end{array} \\
[\![\Gamma \mid \Delta \vdash \mu\alpha{:}\, A.t : A]\!](\vec{x} \mid \vec{\alpha}) & = & \underline{\lambda} k{:}\, R^{[\![A]\!]}.[\![\Gamma \mid \Delta, \alpha{:}\, A \vdash t : \bot]\!](\vec{x} \mid \vec{\alpha}, k)(\underline{\lambda} x{:}\, 0.?_R(x)) \\
[\![\Gamma \mid \Delta \vdash [\alpha_i]t : \bot]\!](\vec{x} \mid \vec{\alpha}) & = & \underline{\lambda} k{:}\, R^0.[\![\Gamma \mid \Delta \vdash t : A]\!](\vec{x} \mid \vec{\alpha})\alpha_i
\end{array}
$$

Figure 6: Interpretation of λ_μ in a category of values

We remark that the interpretation of the lambda calculus fragment coincides with the usual cbv continuation semantics of or equivalently the cps translation of lambda calculus given, e.g., in[11].

The following soundness theorem is proved by induction on typing derivations where we maintain the additional invariant that the interpretation of a value $\Gamma \mid \Delta \vdash v : A$ factors through $\eta_{[\![A]\!]}$. As a sublemma we use that

$$[\![\Gamma \mid \Delta \vdash E[t] : B]\!](\vec{x} \mid \alpha) = \underline{\lambda} k{:}\, R^{[\![B]\!]}.[\![\Gamma \mid \Delta \vdash t]\!](\vec{x} \mid \vec{\alpha})(\underline{\lambda} v{:}\, [\![A]\!].[\![\Gamma, x{:}\, A \mid \Delta \vdash E[x]]\!](\vec{x}, v \mid \vec{\alpha})k$$

which is easily established by induction on the structure of E.

Theorem 5.3 (Soundness) *The set of equations $\Gamma \mid \Delta \vdash t_1 = t_2 : A$ where t_1, t_2 are appropriately typed terms and $[\![\Gamma \mid \Delta \vdash t_1 : A]\!] = [\![\Gamma \mid \Delta \vdash t_2 : A]\!]$ is a cbv-λ_μ-theory.*

5.2 Completeness of continuation semantics

Again, our goal is to establish the following result.

Theorem 5.4 *For every cbv-λ_μ-theory $\mathcal{E}$ over a signature $(\mathcal{B}, \mathcal{K})$ there exists a category of values $(\mathcal{V}, R)$ and an interpretation of base types and constants with the following two properties.*

1. *$\mathcal{E}$ is the theory induced by this interpretation.*
2. *Let Γ be an object context, Δ be a continuation context and A be a type. Every $\mathcal{V}$-morphism $f : [\![\Gamma]\!] \cdot R^{[\![\Delta]\!]} \to R^2([\![A]\!])$ arises as the interpretation of some λ_μ-term $\Gamma \mid \Delta \vdash t : A$.*

Assume a fixed signature $(\mathcal{B}, \mathcal{K})$ and a theory $\mathcal{E}$. We are going to describe a particular category of values $\mathcal{V}$ from these data.

Definition 5.5 Let Γ be an object context and A be a type. A *semantic value term* (*V-term* for short) of type A in context Γ is a term $\Gamma \mid \cdot \vdash t : A$ such that $\Gamma, \Gamma' \mid \Delta \vdash (\lambda x{:}\, A.s)t = s[t/x] : C$ for every term $\Gamma, \Gamma' \mid \Delta \vdash \lambda x{:}\, A.s : A \to C$.

More generally, a *semantic value map* (*V-map* for short) from Γ to $\Theta \equiv x_1{:}\, A_1, \ldots, x_n{:}\, A_n$ is an n-tuple $(t_1, \ldots, t_n)$ such that t_i is a V-term of type A_i in context Γ. $\Diamond$

By (β_V) every syntactic value (term ranged over by $u, v, \ldots$) is a V-term. Depending on the theory there may be more. We will use capital letters $U, V, \cdot$ to range over semantic values.

For a V-term we refer to the defining property $(\lambda x.t)V = t[V/x]$ by (β_V) as well.

Theorem 5.6 *The object contexts with $\mathcal{E}$-equivalence classes of V-maps as morphisms and componentwise substitution form a category with cartesian products given by iuxtaposition.*

Proof. We only have to show that substitution is well-defined on equivalence classes. This, however, is immediate from (β_V): If $\Gamma, x{:}\, A \mid \cdot V_1 = V_2 : C$ and $\Gamma \mid \cdot \vdash U : A$ are V-maps as indicated then (in $\Gamma \mid \cdot$) we have

$$V_1[U/x] = (\lambda x{:}\, A.V_1)U = (\lambda x{:}\, A.V_2)U = V_2[U/x]$$

The rest literally follows the standard proof [10] that contexts and substitutions form a category with finite products. $\square$

Although the morphisms in $\mathcal{V}$ formally are equivalence classes we will mostly refer to them via representatives without explicitly saying so as this simplifies the exposition.

Lemma 5.7 *If $\Gamma \mid \cdot \vdash U : \bot$ then $\Gamma, \Gamma' \mid \Delta \vdash s = t : C$ for all terms s, t.*

Proof. We have $t = (\lambda x{:}\, \bot.t)U = \mu\gamma{:}\, C.U$ by (β_V) and $(\mu_\bot)$. $\square$

Corollary 5.8 *The context $0 \equiv x{:}\, \bot$ is an initial object in $\mathcal{V}$.*

Proof. The morphism $?_{x_1:A_1,\ldots,x_n:A_n}$ is the tuple consisting of $\mu\alpha_i{:}\, A_i.x$. The required instance of (β_V) as well as the uniqueness are special cases of the previous lemma. $\square$

The type object $\mathbf{T}$ are the contexts of length one. So 0 is a type object as required. We will henceforth notationally identify type objects and types thus writing e.g. $\bot$ rather than $x{:}\, \bot$ for 0.

Theorem 5.9 *The category $\mathcal{V}$ together with $R = (\bot \Rightarrow \bot) \Rightarrow \bot$ is a category of values. The exponential R^A is given by $A \Rightarrow \bot$; the exponential $R^2(B)^A$ is given by $A \Rightarrow B$.*

Proof. We only give the raw data establishing the required structure; the verifications consist of lengthy but essentially straightforward equational reasoning. For a very similar system they are explicitly carried out in [4].

Let $U : A \Rightarrow \bot$ and $V : A$ be V-terms (in some implicit ambient context Γ). We define the application $app(U, V) : R$ as $\lambda k{:}\, \bot \Rightarrow \bot.k(U\ V)$. Conversely, if $x{:}\, A \vdash V : R$ then we define $cur_{x:A}(V) : A \Rightarrow \bot$ as $\lambda x{:}\, A.V(\lambda y{:}\, \bot.y)$.

Now let $U : A \Rightarrow B$ and $V : A$ be V-terms. We define the application $app(U, V) : (B \Rightarrow \bot) \Rightarrow \bot$ as $\lambda k{:}\, B \Rightarrow \bot.k(U\ V)$. If $x{:}\, A \vdash U : (B \Rightarrow \bot) \Rightarrow \bot$ then we define $cur_{x:A}(V) : A \Rightarrow B$ as $\lambda x{:}\, A.\mu\beta{:}\, B.V(\mathsf{v}_\beta)$ where as before $\mathsf{v}_\beta = \lambda x{:}\, B.[\beta]x$. $\square$

In view of Convention 4.2 application and abstraction can be given by literally the same formulas which justifies the use of the same operator names for both.

Lemma 5.10 *Let $U : C \Rightarrow D$ and $V : C$ be V-terms possibly containing a free variable $x : A$. Then*

$$cur_{x:A}(app(U, V) = \lambda x{:}\, A.UV$$

Proof.

$$
\begin{array}{lll}
 & cur_{x:A}(app(U,V) & \\
= & \lambda x{:}\,A.(\mu\delta{:}\,D.app(U,V)\mathsf{v}_\delta) & \\
= & \lambda x{:}\,A.(\mu\delta{:}\,D.(\lambda k{:}\,D \Rightarrow \bot.k(UV))\mathsf{v}_\delta) & \\
= & \lambda x{:}\,A.\mu\delta{:}\,D.(\lambda y{:}\,D.[\delta]y)(UV) & \text{by } (\beta_V) \\
= & \lambda x{:}\,A.\mu\delta{:}\,D.[\delta](UV) & \text{by (E)} \\
= & \lambda x{:}\,A.UV & \text{by } (\mu\text{-}\eta)
\end{array}
$$

□

Now we consider the interpretation of λ_μ in $\mathcal{V}$ induced by $[\![B]\!] = B$ for base types and $[\![c]\!] = \lambda k{:}\,A \Rightarrow \bot.kc$ when $c : A$ in $\mathcal{K}$. If $\Gamma \equiv x_1{:}\,A_1, \ldots, x_n{:}\,A_n$ and $\Delta \equiv \beta_1{:}\,B_1, \cdots, \beta_m{:}\,B_m$ then we write $[\![\Gamma]\!] \cdot R^{[\![\Delta]\!]}$ as $x_1{:}\,A_1, \ldots, x_n{:}\,A_n, f_{\beta_1}{:}\,B_1 \Rightarrow \bot, \ldots, f_{\beta_m}{:}\,B_m \mid \cdot$.

Theorem 5.11 *For $\Gamma \mid \Delta \vdash t : A$ we have*

$$[\![\Gamma]\!] \cdot R^{[\![\Delta]\!]} \vdash [\![\Gamma \mid \Delta \vdash t : A]\!] = \lambda k{:}\,A \Rightarrow \bot : (A \Rightarrow \bot) \Rightarrow \bot.kt^*$$

where t^* is $t[f_{\beta_1}[]\ /\ \beta_1] \ldots [f_{\beta_n}[]\ /\ \beta_m]$.

Proof. By induction on the structure of t. We write $\Gamma \equiv x_1{:}\,A_1, \ldots, x_n{:}\,A_n$ and $\Delta \equiv \beta_1{:}\,B_1, \ldots, \beta_m{:}\,B_m$.

The cases where t is a variable or a constant are immediate from the definition as $x^* = x$ and $c^* = c$.

Case $t = \lambda x{:}\,C.s$ and $A = C \Rightarrow D$.

$$
\begin{array}{lll}
 & [\![\Gamma \mid \Delta \vdash \lambda x{:}\,C.s]\!] & \\
= & cur_{k:[\![C \Rightarrow D]\!] \Rightarrow \bot}(app(k, cur_{x:C}([\![\Gamma, x{:}\,C \mid \Delta \vdash s : D]\!]))) & \\
= & \lambda k.k(\lambda x{:}\,C.\mu\delta{:}\,D.[\![\Gamma \mid \Delta \vdash s : D]\!]\mathsf{v}_\delta) & \text{by Lemma 5.10} \\
= & \lambda k.k(\lambda x{:}\,C.\mu\delta{:}\,D.(\lambda q{:}\,D \Rightarrow \bot.q\ t^*)\mathsf{v}_\delta & \text{by IH} \\
= & \lambda k.k(\lambda x{:}\,C.\mu\delta{:}\,D.\mathsf{v}_\delta t^*) & \text{by } (\beta_V) \\
= & \lambda k.k(\lambda x{:}\,C.\mu\delta{:}\,D.[\delta]t^*) & \text{by (E)} \\
= & \lambda k.k(\lambda x{:}\,C.t^*) & \text{by } (\mu\text{-}\eta) \\
= & \lambda k.k(\lambda x{:}\,C.t)^* & \text{by def. of } -^*
\end{array}
$$

Case $t = t_1\ t_2$. We assume that $t_1 : C \Rightarrow A$ and $t_2 : C$. Now we calculate as follows.

$$
\begin{array}{lll}
 & [\![\Gamma \mid \Delta \vdash t_1\ t_2 : A]\!] & \\
= & \quad cur_{k:A \Rightarrow \bot}(app([\![\Gamma \mid \Delta \vdash t_1]\!], & \\
 & \qquad cur_{f:C \Rightarrow A}(app([\![\Gamma \mid \Delta \vdash t_2 : C]\!], cur_{v:C}(app(app(f,v),k)))))) & \\
= & \lambda k.([\![\Gamma \mid \Delta \vdash t_1 : C \Rightarrow A]\!](\lambda f.([\![\Gamma \mid \Delta \vdash t_2 : A]\!](\lambda v.k(fv))))) & \text{by Lemma 5.10} \\
= & \lambda k.(\lambda q.qt_1^*)(\lambda f.(\lambda r.rt_2^*)(\lambda v.k(fv))) & \text{by IH} \\
= & \lambda k.(\lambda q.qt_1^*)(\lambda f.(k(ft_2^*))) & \text{by } (\beta_V)\text{,(E)} \\
= & \lambda k.k(t_1^* t_2^*) & \text{by } (\beta_V)\text{,(E)} \\
= & \lambda k.k(t_1 t_2)^* &
\end{array}
$$

Case $t = \mu\alpha\colon A.t_1$**.** Let us write $\tilde{t}_1$ for $t_1[f_{\beta_1}[]\,/\,\beta_1]\ldots[f_{\beta_m}[]\,/\,\beta_m]$. We have $t_1^* = \tilde{t}_1[f_\alpha[]\,/\,\alpha]$ and furthermore $(\mu\alpha\colon A.t_1)^* = \mu\alpha\colon A.\tilde{t}_1$. Now we calculate as follows.

$$
\begin{array}{lll}
 & [\![\Gamma \mid \Delta \vdash \mu\alpha\colon A.t_1]\!] & \\
= & cur_{f_\alpha : A \Rightarrow \perp}(app([\![\Gamma \mid \Delta, \alpha\colon A \vdash t_1 : \perp]\!], cur_{x:\perp}(\mu\rho\colon R.x))) & \\
= & \lambda f_\alpha.[\![\Gamma \mid \Delta, \alpha\colon A \vdash t_1 : \perp]\!](\lambda x\colon \perp.x) & \text{Lemma 5.10+Cor. 5.8} \\
= & \lambda f_\alpha.t_1^* & \text{by IH, } (\beta_V)\text{,(E)} \\
= & \lambda f_\alpha.\tilde{t}_1[f_\alpha[]\,/\,\alpha] & \\
= & \lambda f_\alpha.f_\alpha(\mu\alpha\colon A.\tilde{t}_1) & (\mu\text{-}\zeta) \\
= & \lambda f_\alpha.f_\alpha(\mu\alpha\colon A.t_1)^* &
\end{array}
$$

Case $t = [\beta_i]t_1$**.**

$$
\begin{array}{lll}
 & [\![\Gamma \mid \Delta \vdash [\beta_i]t_1]\!] & \\
= & cur_{k:\perp \Rightarrow \perp}(app([\![\Gamma \mid \Delta \vdash t_1 : B_i]\!], f_{\beta_i})) & \\
= & \lambda k\colon \perp \Rightarrow \perp.[\![\Gamma \mid \Delta \vdash t_1]\!] f_{\beta_i} & \\
= & \lambda k\colon \perp \Rightarrow \perp.f_{\beta_i} t_1^* & IH, (\beta_V) \\
= & \lambda k\colon \perp \Rightarrow \perp.([\beta_i]t_1)^* & \text{Def. of } -^* \\
= & \lambda k\colon \perp \Rightarrow \perp.kt^* & (\mu_\perp)
\end{array}
$$

□

Proof of Theorem 5.4

Ad 1. Let $t : C$ be a term possibly containing a continuation variable $\alpha : A$. By $(\mu\text{-}\zeta_V)$ we have

$$t[f_\alpha[]\,/\,\alpha] = \mu\gamma\colon C.f_\alpha(\mu\alpha.[\gamma]t)$$

where γ is a fresh continuation variable. It follows that if $t_1 = t_2 \in \mathcal{E}$ then $t_1^* = t_2^* \in \mathcal{E}$ and hence $[\![t_1]\!] = [\![t_2]\!]$ by Thm. 5.11. Conversely, suppose $\Gamma \mid \Delta \vdash t_1, t_2 : A$. Then for $i = 1, 2$ $t_i^* = \mu\alpha.[\alpha]t_i^* = \mu\alpha.(\lambda k.kt_i^*)\mathsf{v}_\alpha = \mu\alpha.[\![t_i]\!]\mathsf{v}_\alpha$ by Thm. 5.11. So, if $[\![t_1]\!] = [\![t_2]\!]$ then $t_1^* = t_2^*$. Substituting v_{β_i} for f_{β_i} yields $t_1 = t_2$ using (β_V) and (E).

Ad 2. Let $[\![\Gamma]\!] \cdot R^{[\![\Delta]\!]} \vdash V : (A \Rightarrow \perp) \Rightarrow \perp$ be a V-term. We define a term t with $[\![\Gamma \mid \Delta \vdash t : A]\!] = V$ by $t \equiv \mu\alpha.(V[\mathsf{v}_{\beta_1}/f_{\beta_1}]\ldots[\mathsf{v}_{\beta_1}/f_{\beta_1}])\mathsf{v}_\alpha$. Now $t^* = \mu\alpha.V\mathsf{v}_\alpha$ by (E). Now

$$
\begin{array}{lll}
 & [\![t]\!] & \\
= & \lambda k.kt^* & \text{by Thm. 5.11} \\
= & \lambda k.k(\mu\alpha.V\mathsf{v}_\alpha) & \\
= & \lambda k.V(\mathsf{v}_\alpha[k[]/\alpha]) & (\mu\text{-}\zeta_V) \\
= & \lambda k.Vk & \\
= & V &
\end{array}
$$

□

It can be shown that our axiomatisation of call-by-value λ_μ without non-logical axioms is complete for the usual continuation-passing style translation of call-by-value λ_μ-calculus into simply typed lambda calculus. We omit the details as they closely follow the development in Section 4.3.

Types: $A ::= \mathcal{B} \mid \perp \mid A_1 \Rightarrow A_2$

Contexts: $\Gamma \equiv x_1 \colon A_1, \ldots, x_n \colon A_n$

Terms: $t ::= \mathcal{K} \mid \mathcal{C}_A \mid x \mid \lambda x \colon A.t \mid t_1\ t_2$

Values: $v ::= x \mid \lambda x \colon A.t$

Evaluation contexts: $E ::= [] \mid vE \mid Et$

Typing rules:

$$\frac{}{x_1 \colon A_1, \ldots x_n \colon A_n \vdash x_i : A_i} \qquad \frac{c \colon A \in \mathcal{K}}{\Gamma \vdash c : A} \qquad \frac{}{\Gamma \vdash \mathcal{C}_A : ((A \Rightarrow \perp) \Rightarrow \perp) \Rightarrow A}$$

$$\frac{\Gamma \vdash t_1 : A \Rightarrow B \quad \Gamma \vdash t_2 : A}{\Gamma \vdash t_1\ t_2 : B} \qquad \frac{\Gamma, x \colon A \vdash t : B}{\Gamma \vdash \lambda x \colon A.t : A \Rightarrow B}$$

Equations:

(β_V) $\quad (\lambda x \colon A.v)s = t[v/x]$.

(η_V) $\quad \lambda x \colon A.vx = v$,
when x is not free in v.

$(\mathcal{C}\text{-}\beta)$ $\quad \mathcal{C}_A(\lambda k \colon A \Rightarrow \perp.kt) = t$

$(\mathcal{C}\text{-}\zeta)$ $\quad E[\mathcal{C}_A\ t] = \mathcal{C}_B(\lambda k \colon B \Rightarrow \perp.t(\lambda x \colon A.kE[x]))$

$(\mathcal{C}_\perp)$ $\quad \mathcal{C}_\perp\ t = t\ (\lambda x \colon \perp.x)$

(E) $\quad (\lambda x \colon A.E[x])t = E[t]$

Figure 7: Syntax and equations for $\lambda_\mathcal{C}$

5.3 Equivalence of call-by-value λ_μ and call-by-value $\lambda_\mathcal{C}$

In a series of papers (see e.g., [3, 13]) Felleisen and his co-workers have studied extensions of the untyped lambda calculus by a control operator $\mathcal{C}$ which allows one to access the current continuation of a term. A typed version of this system called $\lambda_\mathcal{C}$ has been introduced in [4]. The main result of that paper was that a certain equational aiomatisation of $\lambda_\mathcal{C}$ is complete for the usual continuation semantics albeit over arbitrary cartesian closed categories of values.

We will now describe a slightly simplified version of $\lambda_\mathcal{C}$ and give back-and-forth translations to call-by-value λ_μ.

The types of $\lambda_\mathcal{C}$ are the same as the λ_μ-types. Accordingly, the notion of signature is not changed either. Assume a signature $(\mathcal{B}, \mathcal{K})$. The terms of $\lambda_\mathcal{C}$ are the terms of simply-typed lambda calculus over this signature extended by a family of constants

$$\mathcal{C}_A : ((A \Rightarrow \perp) \Rightarrow \perp) \Rightarrow A$$

for each type A. Figure 7 contains a formal definition.

Definition 5.12 [$\lambda_\mathcal{C}$-theory] A $\lambda_\mathcal{C}$-theory is a set of typed equations $\Gamma \vdash t_1 = t_2 : A$

closed under weakening and congruence rules containing all well-typed instances of the basic equations in Fig. 7. $\Diamond$

Definition 5.13 If Δ is a continuation context of λ_μ then let Δ^* stand for the context which contains a binding $f_\alpha : A \Rightarrow \bot$ for every binding $\alpha : A$ in Δ.

To each λ_μ-term $\Gamma \,|\, \Delta \vdash t : A$ we associate a $\lambda_{\mathcal{C}}$-term t^* of type A in context Γ, Δ^* by the following inductive definition.

$$
\begin{aligned}
&x^* = x\\
&c^* = c\\
&(\lambda x\colon A.t)^* = \lambda x\colon A.t^*\\
&(t_1\ t_2)^* = t_1^*\ t_2^*\\
&(\mu\alpha\colon A.t)^* = \mathcal{C}_A(\lambda f_\alpha\colon A \Rightarrow \bot.t^*)\\
&([\alpha]t)^* = f_\alpha\ t^*
\end{aligned}
$$

In order to formulate an inverse translation it is convenient to assume two kinds of variables in $\lambda_{\mathcal{C}}$; ordinary ones which become object variables under the translation and special ones of the form f_α which become continuation variable α under the translation. In order that this is possible we require that an f_α-variable always has negated type, i.e. one of the form $A \Rightarrow \bot$. This refinement is merely for convenience. It does not affect the equational theory and the typing rules which do not distinguish between the two kinds of variables. It would be possible to avoid the distinction between the two kinds of variables by formulating the translation relative to a list of variables which are to be translated into continuation variables. This, however, would clutter the subsequent proofs.

If Γ is a $\lambda_{\mathcal{C}}$ context let Γ_{ob} be the λ_μ-object context consisting of all bindings $x\colon A$ in Γ where x is an ordinary variable; let Γ_{cont} be the continuation context consisting of all bindings $\alpha : A$ where $f_\alpha : A \Rightarrow \bot$ is in Γ and f_α is a "special" variable. Note that Γ is a permutation of $\Gamma_{\mathrm{ob}}, \Gamma^*_{\mathrm{cont}}$ and also $(\Gamma \,|\, \Delta)^*_{\mathrm{ob}} = \Gamma$ and $(\Gamma \,|\, \Delta)^*_{\mathrm{cont}} = \Delta$. Now, for a $\lambda_{\mathcal{C}}$-term $\Gamma \vdash t : A$ we construct a λ_μ-term $\Gamma_{\mathrm{ob}} \,|\, \Gamma_{\mathrm{cont}} \vdash t^\circ : A$ by

$$
\begin{aligned}
&x^\circ = x\\
&c^\circ = c\\
&f_\alpha^\circ = \mathsf{v}_\alpha\\
&(\lambda x\colon A.t)^\circ = \lambda x\colon A.t^\circ\\
&(t_1\ t_2)^\circ = t_1^\circ\ t_2^\circ\\
&\mathcal{C}_A^\circ = \lambda f\colon (A \Rightarrow \bot) \Rightarrow \bot.\mu\alpha.f(\mathsf{v}_\alpha)
\end{aligned}
$$

$\Diamond$

The task of continuation context Δ is to identify those variables in t which are to become continuation variables. If we merely want to map a $\lambda_{\mathcal{C}}$-term into λ_μ then we can always choose Δ to be the empty context.

We can also translate theories as follows. If $\mathcal{E}$ is a λ_μ-theory then let $\mathcal{E}^*$ stand for the set of equations

$$\{\Gamma \vdash t_1 = t_2 : A \mid \Gamma_{\mathrm{ob}} \,|\, \Gamma_{\mathrm{cont}} \vdash t_1^\circ = t_2^\circ : A \in \mathcal{E}\}$$

Conversely, if $\mathcal{E}$ is a $\lambda_{\mathcal{C}}$-theory then let $\mathcal{E}^\circ$ stand for the set of equations

$$\{\Gamma \,|\, \Delta \vdash t_1 = t_2 \mid \Gamma, \Delta^* \vdash t_1^* = t_2^* : A \in \mathcal{E}\}$$

Theorem 5.14 *If $\mathcal{E}$ be a cbv-λ_μ-theory and $\Gamma \,|\, \Delta \vdash t : A$ then $\Gamma \,|\, \Delta \vdash t = (t^*)^\circ : A$ is in $\mathcal{E}$. Moreover, $\mathcal{E}^*$ is a $\lambda_{\mathcal{C}}$-theory.*

Conversely, if $\mathcal{E}$ is a $\lambda_{\mathcal{C}}$-theory and $\Gamma \vdash t : A$ then $\Gamma \vdash t = (t^\circ)^*$ is contained in $\mathcal{E}$. Moreover, $\mathcal{E}^\circ$ is a cbv-λ_μ-theory.

Proof. The fact that the translations $*$ and $\circ$ are mutually inverse up to any theory is an immediate induction on the structure of t. The fact that $\mathcal{E}^*$ and $\mathcal{E}^\circ$ are theories amounts to checking that the basic equality axioms are mapped to theorems. For axioms $(\beta_V), (\eta_V), (E)$ this is immediate from the definition. For axiom $(\mu\text{-}\zeta)$ we notice that $(t[E\,/\,\alpha])^* = t^*[\lambda x.E^*[x]/f_\alpha]$. Equation $(\mu\text{-}\zeta)$ then follows from $(\mathcal{C}\text{-}\zeta)$ and $(\beta_V), (E)$. All other axioms are direct. $\square$

This translation allows us to transport our completeness result Thm. 5.4 to $\lambda_\mathcal{C}$ thus extending the results in [4] to arbitrary theories. This might be of interest as many applications of control operators involve general recursion and recursive datatypes both of which can be subsumed under appropriate equational theories.

We close this section by remarking that a similar translation for the original, i.e., call-by-name, λ_μ-calculus does not seem possible. The reason is that $\lambda_\mathcal{C}$ does not distinguish between the application of a variable to a term and "naming" of a term, i.e., the operation $t \mapsto [\alpha]t$. However, in call-by-name λ_μ the latter operation can be moved inside a μ-abstraction (by $(\mu\text{-}\beta)$) whereas the former cannot.

This is not in conflict with the translation described in [2] as the latter only validates the computational rules of λ_μ, i.e., (β),$(\mu\text{-}\zeta)$, but not rules $(\eta)$and $(\mu\text{-}\beta)$,$(\mu\text{-}\eta)$. Since $(\mu\text{-}\beta)$ is not part of the equations to be translated the abovementioned difficulty does not occur. We agree with de Groote that rules like (η) or $(\mu\text{-}\beta)$ are irrelevant from a computational point of view. They are, however, important logical principles for reasoning about open terms.

6 Conclusions and further work

In the first four sections we have presented a natural continuation style interpretation of call-by-name λ_μ-calculus and demonstrated that the equational theory of λ_μ is complete with respect to this interpretation.

In Section 5 we have extended our results to a call-by-value version of λ_μ and—via a back-and-forth translation—extended previous results in [4] and [13] to arbitrary theories. However, we wish to stress that the main contribution of the paper is the completeness proof for the call-by-name calculus because unlike in [13, 4] the equational theory for call-by-name predated the continuation style interpretation which provides evidence for its canonicity. Indeed, although known for a while, the present CPS translation for call-by-name using pairs does not seem to have received the attention it deserves. We hope that this paper will help to popularize it.

It should be stressed that the precise formulation of our notions of model, e.g. the restriction of exponentiation to certain rather peculiar objects and the duplication of products and terminal object, are not essential for completeness. As mentioned above, every cartesian-closed category together with a distinguished object R defines a category of continuations. Conversely, given an arbitrary category of continuations $(\mathcal{C}, R)$ the category of presheaves $\hat{\mathcal{C}}$ is cartesian closed and the full and faithful Yoneda embedding $\mathcal{Y} : \mathcal{C} \to \hat{\mathcal{C}}$ preserves existing products and exponentials. Therefore, the continuation semantics with respect to $\hat{\mathcal{C}}$ and $\mathcal{Y}(R)$ agrees up to isomorphism with the interpretation in $(\mathcal{C}, R)$ composed with $\mathcal{Y}$. Thus, the completeness result Thm. 4.1 continues to hold for the restricted class of cartesian closed λ_μ-categories. A similar construction can be carried out in the call-by-value situation; in this case a certain complication arises through the fact that the Yoneda embedding does not preserve initial object. This can be remedied by restricting $\hat{\mathcal{V}}$ to the full subcategory consisting of those presheaves F for which $F(0)$ is a singleton. See [4] for details.

Some directions for further work suggest themselves.

Thielecke [14] has described a notion of categorical model for call-by-value lambda calculi with control operators. We believe that using our methods one can show that every instance of this notion is equivalent to a model in which continuations are interpreted as functions[1], i.e., a *standard model* in the terminology of [14]. The details of such a completeness proof may be messy but we expect no intrinsic difficulty.

Although our formulation of λ_μ-syntax is very general and emcompasses, e.g., fixpoint operators and recursive types (via fold/unfold constants), it does not immediately extend to an untyped formulation of λ_μ. The reason is that equations (μ-ζ) and ($\mu_\perp$) (as well as the Convention 4.2) are type dependent and would become unsound if all types were identified.

Another interesting topic for future research might be to derive complete axiomatisations for λ_μ with additional structure such as inductively defined datatypes. More concretely, consider the class of those continuation categories which support natural numbers and lists (in a suitably formalised sense). The continuation semantics allows one to model λ_μ extended by natural numbers and lists in any such category. The task would be to find an axiomatisation of the λ_μ-theory arising from these interpretations. This might be of use for equational transformation of λ_μ-programs manipulating concrete datatypes.

References

[1] Boris Agapiev and Eugenio Moggi. Declarative Continuations and Monads. unpublished draft, July 1991.

[2] Philippe de Groote. On the relation between $\lambda\mu$-calculus and the syntactic theory of sequential control. In F. Pfenning, editor, *Proc. 5th International Conference on Logic Programming and Automated Reasoning, LPAR'94*, pages 31–43. Springer Verlag, 1994. volume 822 of Lecture Notes in Artificial Intelligence.

[3] Matthias Felleisen et al. A Syntactic Theory of Sequential Control. *TCS*, 52:205–237, 1987.

[4] Martin Hofmann. Sound and complete axiomatisations of call-by-value control operators. *Math. Struct. Comp. Sci*, 5:461–482, 1995.

[5] Jean-Louis Krivine. Classical logic, storage operators, and second-order lambda calculus. *Annals of Pure and Applied Logic*, 68:53–78, 1994.

[6] Y. Lafont, B. Reus, and T. Streicher. Continuation semantics or expressing implication by negation. Technical Report 93-21, University of Munich, 1993.

[7] C.-H. L. Ong and C. A. Stewart. A curry-howard foundation for functional computation with control. In *Proceedings of ACM SIGPLAN-SIGACT Symposium on Principle of Programming Languages, Paris, January 1997*. ACM Press, 1997.

[8] Luke Ong. A semantic view of classical proofs. In *Proceedings of LICS '96*. IEEE Press, 1996.

[9] Michel Parigot. $\lambda\mu$-calculus: an algorithmic interpretation of classical matural deduction. In *Proc. LPAR '92, vol. 624 of LNCS*. Springer, 1992.

[10] Andrew Pitts. Categorical logic. In *Handbook of Logic in Computer Science (Vol. VI)*. Oxford University Press, 1997. To appear.

[11] Gordon Plotkin. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1:125–159, 1975.

[12] B. Reus and T. Streicher. Continuation Semantics : Abstract Machines and Control Operators, 1996. submitted.

[13] Amr Sabry and Matthias Felleisen. Reasoning about programs in continuation-passing style. *Lisp and Symbolic Computation*, 6(3/4):287–358, 1993. Preliminary version in Proceedings of the ACM Conference on Lisp and Functional Programming, 1992. Also Technical Report 92-180, Rice University.

[14] Hayo Thielecke. *Categorical Structure of Continuation Passing Style*. PhD thesis, University of Edinburgh, 1997. also as LFCS technical report ECS-LFCS-97-376.