Separation and the $\Lambda\mu$-calculus

Alexis Saurin*

INRIA Futurs & École Polytechnique

Abstract. The $\lambda\mu$-calculus is an extension of the λ-calculus introduced in 1992 by Parigot [Par92] in order to generalize the Curry-Howard isomorphism to classical logic. Two versions of the calculus are usually used in the literature and considered as equivalent: Parigot's original syntax and an alternative syntax used by several authors. In 2001, David and Py [DP01] proved that the Separation Property (also referred to as Böhm theorem) fails for Parigot's $\lambda\mu$-calculus.
By analyzing David & Py's result, we exhibit an extension of Parigot's $\lambda\mu$-calculus, the $\Lambda\mu$-calculus, for which the Separation Property actually holds and which precisely corresponds to the alternative syntax. We prove the theorem adapting a technique by Joly [Jol00] and describe how $\Lambda\mu$-calculus can be considered as a calculus of terms and streams (λ being the abstraction on terms while μ abstracts on streams). We illustrate Separation in showing how in $\Lambda\mu$-calculus it is possible to separate the counter-example used by David & Py.
Finally we argue that $\lambda\mu$-calculus à la Parigot and the alternative presentation ($\Lambda\mu$-calculus) are distinct calculi, the $\Lambda\mu$-calculus having better properties.

Keywords: Pure $\lambda\mu$-calculus, Böhm Theorem, Calculus of Streams.

1 Introduction

The Separation Property. In 1968, Corrado Böhm [Böh68] proved the Separation Property, an essential syntactical property of the pure λ-calculus which states that for any two $\beta\eta$-normal forms M and N of the λ-calculus that are syntactically different, there exists a context $C[]$ that separates them, *id est* the terms $C[M]$ and $C[N]$ will reduce to two different variables. This property has consequences on the semantical level as well as on the syntactical one: on the one hand, it implies that a model of the λ-calculus cannot identify two different $\beta\eta$-normal forms without being trivial, on the other hand, it means that there is a kind of adequation between the reduction rules and the syntactical constructions of the language: it is possible to explore normal terms completely by means of the computational rules. Separation is thus desirable since it is related to the question of whether the syntax and the reduction rules fit each other well.

Böhm's original proof [Böh68,Bar84,Kri93] was achieved through the use of what is called Böhm Transformations that are elementary transformations of the terms in order to put the desired subterm in head position. In his PhD defended in 2000, Thierry Joly [Jol00] gave a new elegant and particularly short proof of the Böhm theorem. The proof that we will present in this paper is closely inspired by Joly's proof.

* LIX, École Polytechnique, 91128 Palaiseau, FRANCE email: saurin@lix.polytechnique.fr

The $\lambda\mu$-calculus. The $\lambda\mu$-calculus is an extension of the λ-calculus introduced in 1992 by Michel Parigot in order to generalize the Curry-Howard isomorphism to classical logic by developing a term assignment for a presentation of classical natural deduction.

The question of whether or not the $\lambda\mu$-calculus satisfies a kind of Separation Property has been answered negatively by René David and Walter Py [DP01] by exhibiting a pair of "normal forms" that are not equivalent but non-separable.

In this paper, we address the same question giving the opposite answer although their result is perfectly valid: as a result of a careful analysis of David & Py's result, we exhibit a straightforward extension of Parigot's $\lambda\mu$-calculus, the $\Lambda\mu$-calculus, for which the Separation Property actually holds. This is done by liberalizing the syntactical constructs of the calculus and by giving a new presentation of the $\lambda\mu$-calculus that stresses the interpretation of $\Lambda\mu$-calculus as a calculus of terms and streams of terms.

The extension obtained is not new in the sense that it is considered by several authors as a definition for $\lambda\mu$-calculus but the relations between the Parigot's syntax and the alternative presentation are not made clear.

Outline of the paper. In section 2, we briefly review the syntactical theory of the $\lambda\mu$-calculus: its origin and its definition, the reduction rules and the Curry-Howard correspondence, plus the syntactical properties of the calculus. The following section is dedicated to the study of David & Py's result, its analysis and comments about non-separation. The conclusions of section 3 lead us naturally in section 4 to consider an extension of Parigot's $\lambda\mu$-calculus for which we prove the Separation Theorem in section 5 by adapting Joly's proof technique. As an illustration of our result, we show how in $\Lambda\mu$-calculus it is possible to separate the two terms of David & Py's counter-example to the Separation Property.

Remark on notations. In the following, we will as usual consider the application to be left associative and we will consider two particular λ-terms (or $\lambda\mu$-terms or $\Lambda\mu$-terms according to the context): $1 = \lambda x, y.x$ and $0 = \lambda x, y.y$. Moreover, given an integer k, 1^k will denote the sequence of k terms all equal to 1: $(M)1^k$ corresponds to the term M applied to k arguments all equal to 1.

2 $\lambda\mu$-calculus

Extending Curry-Howard to classical logic. Parigot's $\lambda\mu$-calculus [Par92] arose from aiming at extending the Curry-Howard isomorphism to classical logic by developing a term assignment for a presentation of minimal classical natural deduction. Several propositions preexisted (for instance Felleisen's $\lambda\mathcal{C}$-calculus [FH92] designed to provide a syntactic theory of control and later linked with classical logic by Griffin's analysis [Gri90], or Girard's LC [Gir91] which gives a presentation of classical logic based on the notion of polarities for which cut-elimination is confluent), but Parigot's approach differs from both of the previously mentioned theories. Indeed it is really a calculus (LC does not have a real syntax, it is still a logic), and the deductive power of classical logic is not recovered by adding an axiom ($\neg\neg A \Rightarrow A$ in the case of the $\lambda\mathcal{C}$-calculus) but by extending the logical rules (in defining a classical version of natural

deduction). To each rule of the logic corresponds a syntactical construct on the computational side of the isomorphism, thus $\lambda\mu$-calculus has more syntactical components than the λ-calculus: μ-abstraction and naming of terms.

Since we want to analyse David & Py's result, our presentation of the $\lambda\mu$-calculus in this section closely follows the one given in their paper. Later in the paper we shall take some distance from this presentation.

Definition 1 ($\lambda\mu$-calculus). *Let $\mathcal{V}$ (denoted by $x, y, \ldots$) and $\mathcal{V}_{cont}$ (denoted by $\alpha, \beta, \ldots$) be two disjoint infinite sets of term variables and of continuation variables respectively. $\lambda\mu$-terms are inductively defined by the following syntax:*
$M ::= x \mid \lambda x.M \mid \mu\alpha.[\beta]M \mid (M)M$
We call **named terms** *the expressions of the form $[\beta]M$ (which are not $\lambda\mu$-terms).*

Two versions of the calculus are usually used in the literature: Parigot's original syntax and an alternative syntax[1] which is used by several authors (De Groote [dG94,dG98], Laurent [Lau03a], Ong and Stewart [OS97]...) even if Parigot's presentation seems to be a little more common. Those two versions are generally considered as equivalent and their relation is never made really clear. Actually, on the logical point of view their relation is clear since one corresponds to minimal classical logic while the other is for classical logic with falsity, but computationally, their relation is not clear.

We give below the type system for the simply typed $\lambda\mu$-calculus. Erasing the proof terms, we get a presentation of minimal classical natural deduction. We do not enter into more details about the Curry-Howard isomorphism for $\lambda\mu$-calculus: this paper is only concerned with the untyped calculus. If interested, one can read Ariola & Herbelin's paper [AH03] which gives an interesting analysis of the links between minimal classical logic, $\lambda\mu$-calculus and Felleisen's $\lambda\mathcal{C}$.

$$\frac{}{\Gamma, x : T \vdash x \ : T | \Delta} \ Var \qquad \frac{\Gamma \vdash M \ : B | \Delta, \beta : B}{\Gamma \vdash \mu\alpha.[\beta]M \ : A | (\Delta \cup \{\beta : B\}) \setminus \{\alpha : A\}} \ \mu$$

$$\frac{\Gamma, x : T \vdash M \ : U | \Delta}{\Gamma \vdash \lambda x.M \ : T \to U | \Delta} \ Abs \qquad \frac{\Gamma \vdash M \ : T \to U | \Delta \quad \Gamma \vdash N \ : T | \Delta}{\Gamma \vdash (M)N \ : U | \Delta} \ App$$

$\lambda\mu$-calculus reduction rules. Being interested in the Separation Property, it is necessary to look carefully at the reduction rules of the calculus.

Definition 2 ($\lambda\mu$-calculus reduction rules). *We consider six reduction rules (β, η, μ, ν, ρ and θ) for $\lambda\mu$-calculus[2]:*

$$(\lambda x.M)N \longrightarrow_\beta M[N/x] \qquad \lambda x.(Mx) \longrightarrow_\eta M \quad if\ x \notin FV(M)$$
$$[\beta]\mu\alpha.M \longrightarrow_\rho M[\beta/\alpha] \qquad \mu\alpha.[\alpha]M \longrightarrow_\theta M \quad if\ \alpha \notin FV(M)$$
$$(\mu\alpha.M)N \longrightarrow_\mu \mu\alpha.M[[\alpha](u)N/[\alpha]u]$$
$$\mu\alpha.M \longrightarrow_\nu \lambda x.\mu\alpha.M[[\alpha](u)x/[\alpha]u] \quad if\ x \notin FV(M)$$

[1] Alternative syntax: $M ::= x \mid \lambda x.M \mid (M)M \mid \mu\alpha.M \mid [\alpha]M$.

[2] The substitution written $M[[\alpha](u)N/[\alpha]u]$ must be read as "all the subterms u of M named with α are replaced by $(u)N$"; the substitution involved in the ρ-rule is just continuation variable renaming. Besides note that the ρ-reduction rule, as stated in definition 2, is not a proper rule of $\lambda\mu$-calculus since it does not involve terms of the calculus. To be precise, the ρ rule should be stated as: $\mu\alpha[\beta]\mu\gamma.[\delta]M \longrightarrow_\rho \mu\alpha.([\delta]M)[\beta/\gamma]$.

When designing the $\lambda\mu$-calculus, Parigot only considered the rules β, μ, ρ and θ. Having in mind the Separation Property, it is essential to have η-rule: without this rule the property is immediately false. Adding η to the four rules considered by Parigot is a bit problematic because it makes confluence fail[3]. Hence the need for introducing an additional rule called ν which corresponds to an η-expansion followed by a μ-reduction. Even in this case, confluence holds only for *μ-closed terms*[4]. At this point the ν-rule already deserves some comments. First, note that it makes the μ-reduction superfluous since it can be simulated by a ν-reduction followed by a β-reduction. Second, note that with this rule we loose any hope of getting normal forms: as soon as a term contains a μ-abstraction it is possible to apply an arbitrarily large number of ν-rules to it.

We will come back later to the study of reduction rules when presenting $\Lambda\mu$-calculus.

Syntactical theorems. Before going further in the analysis of Separation, we review the main syntactical theorems of $\lambda\mu$-calculus.

Parigot only looked on the reduction rules β, μ, θ and ρ and proved confluence of the calculus and strong normalization in the polymorphic typed setting [Par92,Par97]. As mentioned previously the η-rule generates some confluence problems studied by Walter Py in his PhD [Py98]: the introduction of the ν-reduction solves this for *μ-closed terms*.

The syntactical theory of the $\lambda\mu$-calculus is not much more developed than this and in a sense the last result obtained was a negative one: the failure of the Separation Property proved by David & Py in 2001. We shall study their result in the following section.

3 The failure of Separation for the $\lambda\mu$-calculus

The result of David and Py [DP01]. David & Py's result consists of two parts: they first develop tools to state the Separation Property in the $\lambda\mu$-calculus and then prove it is false.

To state Separation, David & Py introduce the reduction rules we presented in the previous section in order to have extensionality rules and confluence. After this, they introduce a notion of **canonical normal forms** which are the $\beta\eta\mu\rho\theta$-normal $\lambda\mu$-terms of the form: $M = \lambda x_1 \dots x_n.((y)N_1 \dots N_k)$ or $M = \lambda x_1 \dots x_n.\mu\alpha[\beta]((y)N_1 \dots N_k)$ with $N_1, \dots, N_k$ in canonical normal form[5].

Given confluence and a notion of normal form, they can state the Separation Property for $\lambda\mu$-calculus that they show to be false by exhibiting a counter-example, namely the $\lambda\mu$-term[6] $W = \lambda x.\mu\alpha.[\alpha]((x)\ \mu\beta.[\alpha]((x)\ U_0\ y)\ U_0)$ with $U_0 = \mu\delta.[\alpha]0$.

[3] There is a critical pair involving μ and η from the term $\lambda x.(\mu\alpha.[\beta]M)x$, with $x \notin FV(M)$, see David & Py's paper [DP01] or Py's PhD thesis [Py98] for more details.

[4] The μ-closed terms are the terms with no free continuation variable.

[5] They are not strictly speaking normal forms since a term with a μ-abstraction has always an infinite computation because of the ν-rule. They are the $\beta\eta\mu\rho\theta$-normal forms for which no ν-reduction creates a redex for β, η, μ, ρ, or θ.

[6] Actually, the term W in David & Py's paper needs two ν-reductions applied to the μ-abstraction $\mu\alpha$ and then some β-reductions in order to reduce to the term in canonical normal forum $W' = \lambda x_1, x_2, x_3.\mu\alpha.[\alpha](x_1)(\mu\beta.[\alpha]((x_1)(\mu\delta.[\alpha]x_3)yx_2x_3)(\mu\delta.[\alpha]x_3)x_2x_3)$.

Theorem 1 (Failure of Separation in $\lambda\mu$ (David & Py)). *Let us consider $W_0 = W[0/y]$ and $W_1 = W[1/y]$. W_0 and W_1 are closed, in canonical normal form and are not $\beta\eta\mu\nu\rho\theta$-equivalent* **but** *they are operationally equivalent*[7].

In other words, the Separation Property fails on W_0 and W_1.

Analysis of the result. Let us now give briefly an intuitive idea of the reasons why W_0 and W_1 cannot be separated. For additional details we refer the reader to David & Py's paper [DP01].

Thanks to a context lemma [DP01] it is possible to consider only applicative contexts. As a consequence, the result is equivalent to the fact that for no context $[]A\overrightarrow{B}$, $(W)A\overrightarrow{B}$ reduces to a head normal form for which y is the head variable. The reason for this can be summarized as follows:
$WA\overrightarrow{B} \rightarrow^{\star} \mu\alpha.[\alpha]((A)\ \mu\beta.[\alpha]((A)\ (\mu\delta.[\alpha](0)\ \overrightarrow{B})\ y\ \overrightarrow{B})\ (\mu\delta.[\alpha](0)\ \overrightarrow{B})\ \overrightarrow{B})$

The first occurrence of A has then to select its first argument and put it in head position while the second occurrence of this term has to select its second argument. Up to this point, it is the same thing as in the λ-calculus case. What differs from λ-calculus and makes the property fail is that those two behaviours of the term A have to be realized in the same context $\overrightarrow{B}$. The failure comes from this very point which is much different from the case of the λ-calculus where the arguments are consumed so that it would be possible to put the term A in two different contexts to make it compute different things. But in the $\lambda\mu$-calculus, in addition to β-reduction, we have μ-reduction which gives all the arguments directly to the subterms labelled with the correct μ-variables and this is what makes the difference.

What can we expect now? Once the failure of Separation in $\lambda\mu$ is proved, we can try to fix the problem and get the property back.

The failure of Separation can be generally stated as follows: there exists two normal forms M and N that are not equivalent for the reduction rules but that no context separates. Clearly, two parts of this proposition are of interest: on the one hand the terms are **not equivalent** and on the other hand there is **no separating context**.

Let us illustrate these points by two very simple examples taken from the λ-calculus:
• Considering λ-calculus with only β as a reduction rule, Separation fails, but adding η makes us get the property back: we have enlarged the equivalence and the result now becomes true. *The terms that were not separable are made equivalent.*
• Considering simply typed λ-calculus, Separation does not hold[8] : we do not have enough contexts to explore all parts of the term. But simply by allowing non-typable terms, that is allowing more contexts (in particular the untypable ones), one gets the Separation Property back. *The terms that were not equivalent are made separable*.

[7] In the usual sense that for all context $C[]$, $C[W_0]$ is solvable iff $C[W_1]$ is solvable.

[8] Other versions of the Separation Property can be proved in the typed setting. Joly gave a intermediate result in his PhD thesis [Jol00]: given two closed simply typed λ-terms t_1 and t_2 with same type that are not $\beta\eta$-equivalent, then by instantiating their type variables to the type of Church numerals $(o \rightarrow o) \rightarrow (o \rightarrow o)$ then one can find an applicative context in which the terms reduce to different Church numerals. These typed results are very different from the untyped Separation theorems we are interested in here.

Those two directions suggest radically different methods to get the property back. In the conclusion of their paper, David & Py suggest that it might be possible to get the Separation Property back by adding new reduction rules, that is by making more terms equivalent (particularly a family of terms related to W). Here instead, we explore the second direction which consists in trying to allow more contexts in order to get a more powerful exploration of the terms and get the Separation Property back.

To put things short, the failure of Separation is due to the fact that the μ-abstraction consumes all its arguments and that it is not really possible to make a μ-abstraction disappear[9]. The reason is that μ-abstraction and naming are a single syntactical construction in Parigot's $\lambda\mu$-calculus. In the following section, we will get rid of this constraint in order to recover Separation.

4 An extension of $\lambda\mu$-calculus: $\Lambda\mu$-calculus

$\Lambda\mu$: more liberal syntactical rules and a new syntax. We now extend our language by liberalizing the syntax in allowing a μ-abstraction on the one hand and a naming construct on the other. We will take the opportunity of this definition of a new language to change also the way naming is written: this simple change will have great impact on the writing of the proofs that will constitute the rest of the paper.

Definition 3 ($\Lambda\mu$-calculus). *Given two infinite disjoint sets $\mathcal{V}_t$ (denoted by $x, y, z, \ldots$) and $\mathcal{V}_s$ (denoted by $\alpha, \beta, \gamma, \ldots$) of variables, the set of $\Lambda\mu$-terms is inductively defined as follows:* $M ::= x \mid \lambda x.M \mid \mu\alpha.M \mid (M)\alpha \mid (M)M$

We will refer to the elements of $\mathcal{V}_t$ and $\mathcal{V}_s$ as term variables (or λ-variables) and stream variables (or μ-variables) respectively. The terminology will be justified below.

Notice that we use the same notation for the application of a term and of a stream variable, this is for pure notational convenience and because it does not make anything ambiguous, but of course those two applications are of different kinds.

Definition 4 ($\Lambda\mu$-calculus reduction rules and typing system).

$$\begin{array}{l|l} (\lambda x.M)N \longrightarrow_\beta M[N/x] & \lambda x.(Mx) \longrightarrow_\eta M \quad if\ x \notin FV(M) \\ (\mu\alpha.M)\beta \longrightarrow_{\beta_s} M[\beta/\alpha] & \mu\alpha.(M\alpha) \longrightarrow_{\eta_s} M \quad if\ \alpha \notin FV_s(M) \end{array}$$

$$\mu\alpha.M \longrightarrow_{fst} \lambda x.\mu\alpha.M[(u)x\alpha/(u)\alpha] \qquad if\ x \notin FV(M)$$

$$\frac{}{\Gamma, x:T \vdash x\ :T|\Delta}\ Var \qquad \frac{\Gamma, x:T \vdash M\ :U|\Delta}{\Gamma \vdash \lambda x.M\ :T \to U|\Delta}\ Abs \qquad \frac{\Gamma \vdash M\ :\bot|\Delta, \alpha:A}{\Gamma \vdash \mu\alpha.M\ :A|\Delta}\ \mu Abs$$

$$\frac{\Gamma \vdash M\ :T \to U|\Delta \quad \Gamma \vdash N\ :T|\Delta}{\Gamma \vdash (M)N\ :U|\Delta}\ App \qquad \frac{\Gamma \vdash M\ :A|\Delta, \alpha:A}{\Gamma \vdash (M)\alpha\ :\bot|\Delta, \alpha:A}\ \mu App$$

Remark: The only "novelty" is the change of notation for naming and the use of the *fst*-rule which is a direct translation of the ν-rule defined by David & Py [DP01].

[9] Indeed the only rule that makes a μ disappear is the ρ rule, but in fact the term stays encapsulated inside a μ: $\mu\alpha[\beta]\mu\gamma.W \longrightarrow_\rho \mu\alpha.W[\beta/\gamma]$ (the θ rule is a little particular since it is an extensionality rule...).

Moreover notice that we dropped the μ-rule since as we mentioned previously it is no more necessarily and it has no particular meaning with the stream interpretation that we develop in the following.

Definition 5 (Canonical normal forms for $\Lambda\mu$($CNF\Lambda$)). *The canonical normal forms are the $\beta\beta_s\eta\eta_s$-normal $\Lambda\mu$-terms of the form:*
$\lambda\overrightarrow{x_1}.\mu\alpha_1 \dots \lambda\overrightarrow{x_k}.\mu\alpha_k.\lambda\overrightarrow{x_{k+1}}.(y)(M_1^i)_{i\in I_1}\beta_1 \dots (M_l^i)_{i\in I_l}\beta_l(M_{l+1}^i)_{i\in I_{l+1}}$
with the M_i^j's in canonical normal form[10].

Definition 6 (Head reduction and head normal form). *The head reduction is the reduction strategy consisting of the head reduction for rules β, η, β_s and η_s, applying the fst-rule only when no other rule applies and only when it creates a head redex for one of the above mentioned rules. This reduction will be written $\rightarrow_h$.*
The head normal forms (hnf) are the terms of the form:
$\lambda\overrightarrow{x_1}.\mu\alpha_1 \dots \lambda\overrightarrow{x_k}.\mu\alpha_k.\lambda\overrightarrow{x_{k+1}}.(y)(M_1^i)_{i\in I_1}\beta_1 \dots (M_l^i)_{i\in I_l}\beta_l(M_{l+1}^i)_{i\in I_{l+1}}$ *with arbitrary applications of stream variable and arbitrary terms M_i's (but such that no η nor η_s rule can be applied).*

More terms, more space to η-expand. With the new language $\Lambda\mu$, we can build new terms (that have no equivalent in Parigot's $\lambda\mu$-calculus) such as $\mu\alpha.\mu\beta.(M)$ or $(M)\alpha x\beta$. Indeed, intuitively, η-expansion (not to mention θ-expansion) was forbidden after a μ-abstraction in $\lambda\mu$-calculus. With $\Lambda\mu$-calculus, we get the freedom to η-expand (with η or η_s) everywhere in the term: $\mu\alpha.\mu\beta.(M)\beta$ or $\mu\alpha.\lambda x.(M)x$ are now perfectly legal terms while neither $\mu\alpha.\mu\beta.[\beta]M$ nor $\mu\alpha.\lambda x.(M)x$ were allowed in $\lambda\mu$.

This change is extremely important for Separation: during the Separation process, η-expansion is critical (or to say it differently, the fact to saturate a term with arguments is essential to satisfy Separation).

Interpretation as a stream calculus. Let us now draw some comments about the links between $\Lambda\mu$-calculus and a calculus of streams in analyzing its reduction rules.

The reduction rules for $\Lambda\mu$-calculus are just the ones for $\lambda\mu$-calculus up to a change in the names (ρ becomes β_s, θ becomes η_s and ν becomes *fst*), the fact that μ disappears (as it is superfluous and has no particular meaning with the stream interpretation), and the fact that β_s is now a well-formed rule (ρ was not well formulated as mentioned in Footnote 2).

Let us now give some intuitive interpretation of these rules. Clearly, η_s is an extensional rule while β_s has more computational content in a sense (even if it is a very weak form of β-reduction since it only involves variable renaming...). Moreover, the fst-rule relates term variables with stream variables and this is what makes the calculus interesting: it is a way to access the first term of the stream. In order to make this point

[10] Notice that these are not all the $\beta\beta_s\eta\eta_s$-normal forms since for example $\mu\alpha.(\lambda x.x)\alpha$ is not in canonical normal form, but is in $\beta\beta_s\eta\eta_s$-normal form.
Notice also that these canonical normal forms directly extend the ones for $\lambda\mu$-calculus: in $\lambda\mu$-calculus we always had $k = l = 0$ or $k = l = 1$ and the abstraction $\lambda flechex_2$ as well as the sequence $(M_2^i)_{i\in I_2}$ are empty in this case.

clear, let us now look at fst-derivations from a term $\mu\alpha.M$:

$$\mu\alpha.M \rightarrow^{\star}_{fst} \lambda x_1 \dots \lambda x_n.\mu\alpha.M[(u)x_1 \dots x_n\alpha/(u)\alpha]$$

Those derivations can be arbitrarily long so that the $\mu\alpha$ can be viewed as a kind of infinite λ-abstraction but not exactly, it is closer to an abstraction over streams of terms (or lazy stacks). This had already been noticed by Parigot in his original paper about $\lambda\mu$-calculus [Par92]: "The operator μ looks like a λ having potentially infinite number of arguments". This analogy is already valid with $\lambda\mu$-calculus but only takes its strength with $\Lambda\mu$-calculus in which Stream Abstraction and Stream Application are two well identified and totally independent constructions (See definition 7 below).

Looking at the shape of the canonical normal form, we notice that the structure of $\lambda\overrightarrow{x}.\mu\alpha.M$ or of $(M)\overrightarrow{N}\alpha$ are frequent in $\Lambda\mu$-calculus. We formalize this in the following definition.

Definition 7 (Stream notation). *To shorten the writing of $\Lambda\mu$-terms, we shall adopt the following writing convention: when it is not ambiguous we abbreviate a sequence of abstractions of the form: $\lambda x_1 \dots \lambda x_n.\mu\alpha.M$ as $\Lambda\mathcal{S}.M$ and a sequence of applications of the form $(M)M_1 \dots M_m\alpha$ as $(M)\mathcal{S}$. For instance the canonical normal form of the previous definition will be written: $\Lambda\mathcal{S}_1 \dots \Lambda\mathcal{S}_k.\lambda\overrightarrow{x_{k+1}}.(y)\mathcal{S}'_1 \dots \mathcal{S}'_l(M^i_{l+1})_{i\in I_{l+1}}$.*
In the rest of the paper, we refer to these notations as **Stream abstraction** *and* **Stream application** *respectively.*

With this notation the stream interpretation becomes clearer and clearer, in particular it would be interesting to study reductions on streams like: $(\mu\alpha.M)\mathcal{S} \rightarrow M[\mathcal{S}/\alpha]$ with this we have a reduction that is more interesting than the pure variable renaming of the β_s rule and that looks better than the strange substitution of the μ-reduction.
In addition, it is possible to make even more uniform this notation: the last layer of term applications in the canonical normal forms which is not very uniform can be encapsulated in a stream just by a η_s-expansion after all the abstractions:
$\Lambda\mathcal{S}_1 \dots \Lambda\mathcal{S}_k.\lambda\overrightarrow{x_{k+1}}.(y)\mathcal{S}'_1 \dots \mathcal{S}'_l(M^i_{l+1})_{i\in I_{l+1}} \rightarrow_{\eta_s exp} \Lambda\mathcal{S}_1 \dots \Lambda\mathcal{S}_{k+1}.(y)\mathcal{S}'_1 \dots \mathcal{S}'_{l+1}$

5 Recovering the Separation Property

5.1 Scheme of the proof for $\Lambda\mu$

Before going to the technical details of the proof of Separation, we present the idea of the proof informally and its scheme.

Given two closed terms M and N that we want to separate, we must consider the open case and reason by induction on the size of the terms and by case on their structure.

The main case of the proof is for M, N of the form: $(x)\mathcal{M}_1 \dots \mathcal{M}_m M^1_{m+1} \dots M^k_{m+1}$ and $(y)\mathcal{N}_1 \dots \mathcal{N}_n N^1_{n+1} \dots N^l_{n+1}$.

When either $x \neq y$ or $(m,k) \neq (n,l)$, it is easy to separate the terms. Separation becomes more intricate in the case when $x = y$ and $m = n$ and $k = l$.

In this case, we need to find two subterms M' and N' at the same position in the trees of M and N that are not equivalent. In the λ-calculus case, it is evident but with $\Lambda\mu$, things are getting more complicated. Indeed it could happen that the terms are of the form $(x)\alpha$ and $(x)y\beta$ (or even of the form $(x)\alpha$ and $(x)\beta$). In this case, there is no position satisfying the condition even if the terms are not equivalent. We should have done a ν-reduction on the $\mu\alpha$ and $\mu\beta$ when the terms were still closed and we would have got the terms $(x)v_\alpha\alpha$ and $(x)yv_\beta\beta$ for which there is no problem. This will be formalized in the Subterm Lemma (Lemma 4) which provides a way to prevent this problem from occurring. Then we select the subterms and finally separate them thanks to the induction hypothesis. This selection is done by assigning some term P to the head variable x.

Nevertheless, it is not so easy to separate M and N since in addition to the selection of the subterms M' and N' by the term P assigned to the head variable x, we also need then to separate M' and N' and there is no reason for x not to appear also in the subterms M' and N' and thus P must be chosen to be compatible with the separation process of those terms.

In order to allow the term P assigned to x to play both roles (selection of the particular subterms M' and N' and separation of $\Lambda\mu$-terms M' and N', respectively), we will need an auxiliary lemma which is the Parametric Pairs Lemma (Lemma 3). The aim of this lemma is to guarantee that it is possible to assign a more structured term to the variable x (a kind of pair of terms)without disturbing the Separation process.

For the terms that do not conform to the shape $(z)\mathcal{P}_1 \dots \mathcal{P}_m Q_1 \dots Q_k$, we need a way to reduce them to the previous case that is compatible with the induction. To achieve that, the size that we shall define on the canonical normal forms of the $\Lambda\mu$-calculus must take into account this reduction in order not to have terms for which the size is growing during this step. The key point here is the stratification of the canonical normal forms between a layer of abstractions and a layer of applications (of streams and terms in both cases).

5.2 Definitions and lemmas

In this section we introduce some structures, definitions and lemmas that will be useful in the proof of the Separation theorem. Firstly we give elements that will deal with the structure of the proof: size of terms to reason by induction as well as a way to relate terms with abstraction at top-level and terms without.

Definition 8 (size of $\Lambda\mu$-terms). *We define inductively a size $\mathcal{T}$ on $\Lambda\mu$-calculus canonical normal forms:*
• $\mathcal{T}(x) = 0$
• $\mathcal{T}(\Lambda\mathcal{S}_1 \dots \Lambda\mathcal{S}_k.\lambda x_1 \dots \lambda x_l.M) = 1+\mathcal{T}(M)$ *if M is not an abstraction and* $k+l > 0$
• $\mathcal{T}((x)\mathcal{S}_1 \dots \mathcal{S}_m N_1 \dots N_n) = 1 + \sum_{i=1}^{m} \mathcal{T}'(\mathcal{S}_i) + \sum_{i=1}^{n} \mathcal{T}(N_i)$ *if* $(m,n) \neq (0,0)$
where the size $\mathcal{T}'(\mathcal{S})$ of a stream application $\mathcal{S} = M_1 \dots M_m\alpha$ is $\sum_{i=1}^{m} \mathcal{T}(M_i)$

Additionally to the size of a term we will need to talk about the subterms of certain terms (since separation is essentially an exploration of the terms). For this purpose,

the size of a stream, positions of subterms and minimal applicative contexts will be important notions:

Definition 9 (Size of a Stream, Position of a subterm). *The size of a stream* $\mathcal{S} = M_1 \dots M_n\alpha$ *is defined as* $\sharp\mathcal{S} = n$.
Given a term M *of the form* $\Lambda\mathcal{S}_1 \dots \Lambda\mathcal{S}_k.\lambda x_1 \dots \lambda x_{k'}.(x)\ \mathcal{S}'_1 \dots \mathcal{S}'_l M_1 \dots M_{l'}$ *its set of positions* $\mathcal{P}(M)$ *is defined to be* $\{(i,j), i \leq l, 1 \leq j \leq \sharp\mathcal{S}'_i\} \cup \{(l+1, j), 1 \leq j \leq l'\}$ *and the subterm of* M *at position* $(i,j) \in \mathcal{P}(M)$ *is the* j^{th} *element of* $\mathcal{S}'_i$ *if* $i \leq l$ *or is* M_j *if* $i = l+1$.

Definition 10 (Minimal Applicative Context). *Given two* $\Lambda\mu$*-canonical normal forms* M *and* N*, a Minimal Applicative Context (MAC) is a context* $\mathcal{C}$ *of the form* $[]x_{11} \dots x_{1k_1}$ $\alpha_1 \dots \alpha_n x_{(n+1)1} \dots x_{(n+1)k_{n+1}}$ *of minimal length such that* $\mathcal{C}[M]$ *and* $\mathcal{C}[N]$ *reduce to terms of the form* $(x)\mathcal{S}_1 \dots \mathcal{S}_s P_1 \dots P_p$ *and such that the variables in* $\mathcal{C}$ *do not interfere with the free variables of* M *and* N[11].

Lemma 1. *Consider two* $\Lambda\mu$*-terms* M *and* N *in canonical normal forms,* $\mathcal{C}$ *a MAC for this pair of terms and* M' *and* N' *the reduced forms of* $\mathcal{C}(M)$ *and* $\mathcal{C}(N)$ *respectively. Then* $\mathcal{T}(M') + \mathcal{T}(N') \leq \mathcal{T}(M) + \mathcal{T}(N)$[12].

Definition 11 (Sub-term selector $\Phi^{\mathcal{P}}_{(i,j)}$). *The sub-term selector* $\Phi^{\mathcal{P}}_{(i,j)}$ *intended to select the sub-term at position* (i,j) *in a term* $M = (x)\mathcal{S}_1 \dots \mathcal{S}_s M_1 \dots M_m$ *is:*
$\Phi^{\mathcal{P}}_{(i,j)} = \mu\alpha_1 \dots \mu\alpha_{i-1}.\lambda x_{(i,1)} \dots \lambda x_{(i,j)}.\mu\alpha_i \dots \mu\alpha_s.\lambda x_{(s+1,1)} \dots \lambda x_{(s+1,m)}.x_{(i,j)}$
when (i,j) *is in the set of positions* $\mathcal{P}$ *of the term* M[13].

In the key part of the proof we shall need to replace a term by another one that carries more information: we need to store two programs. The usual way to do so in λ-calculus is to use pairs, but we need our information no to be retrieved immediately, we need the pair to accept several arguments before choosing what component it will restore, for this a structure of parametric pairs is needed:

Definition 12 (Parametric pairs: $\langle \_, \_ \rangle_k$). *We define inductively* $\langle u, v\rangle_k$ *as follows:*
- $\langle u, v\rangle_0 = \lambda z.(z)uv \qquad z \notin FV_t((u)v)$
- $\langle u, v\rangle_{k+1} = \mu\alpha.\langle (u)\alpha, (v)\alpha\rangle_k \qquad \alpha \notin FV_s((u)v)$

This may be rephrased as: $\langle u, v\rangle_k = \mu\alpha_1 \dots \mu\alpha_k.\lambda z.(z)((u)\alpha_1 \dots \alpha_k)((v)\alpha_1 \dots \alpha_k)$

It is simply a parametric version of the usual encoding of the pair in λ-calculus, adapted to the $\Lambda\mu$-calculus. Moreover these parametric pairs are adapted from the pairs

[11] More formally, if $M = \Lambda\mathcal{S}_1 \dots \Lambda\mathcal{S}_p.\lambda x_1 \dots \lambda x_{p'}.(x)\ \mathcal{S}'_1 \dots \mathcal{S}'_q M_1 \dots M_{q'}$ and $N = \Lambda\mathcal{P}_1 \dots \Lambda\mathcal{P}_r.\lambda y_1 \dots \lambda y_{r'}.(y)\mathcal{P}'_1 \dots \mathcal{P}'_s\ N_1 \dots N_{s'}$, the MAC is of the form: $[]x_{11} \dots x_{1k_1}\alpha_1 \dots \alpha_n\ x_{(n+1)1} \dots x_{(n+1)k_{n+1}}$ and such that $n = max(p,r)$. In the case $p = r = n$, for all $i \leq n$ we have $k_i = max(\sharp\mathcal{S}_i, \sharp\mathcal{P}_i)$ and $k_{n+1} = max(p', r')$ and in the case $p \gneq r$ we have for all $i \leq r$, $k_i = max(\sharp\mathcal{S}_i, \sharp\mathcal{P}_i)$, $k_{r+1} = max(\sharp\mathcal{S}_{r+1}, r')$ and for all $r+1 \lneq i \leq n$, $k_i = \sharp\mathcal{S}_i$ and $k_{n+1} = p'$.

[12] Notice that in general the inequality is not strict even if the MAC in non-trivial. For example the terms $M = \lambda x.y$ and $N = z$ admit as a MAC the context $\mathcal{C} = []x$ which leads to terms $M' = y$ and $N' = (z)x$ and the sum of the sizes has not decreased.

[13] Of course, if $i = s+1$, then $\Phi^{\mathcal{P}}_{(i,j)} = \mu\alpha_1 \dots \mu\alpha_s.\lambda x_{(s+1,1)} \dots \lambda x_{(s+1,m)}.x_{(s+1,j)}$.

defined by Joly for the λ-calculus: $\lambda x_1 \dots \lambda x_n.\lambda z.(z)((u)x_1 \dots x_n)((v)x_1 \dots x_n)$. In a previous version of this work, we had pairs that were indexed twice: $\langle u, v\rangle_{(m,n)} = \mu\alpha_1 \dots \mu\alpha_m.\lambda x_1 \dots \lambda x_n.\lambda z.(z)((u)\alpha_1 \dots \alpha_m x_1 \dots x_n)((v)\alpha_1 \dots \alpha_m x_1 \dots x_n)$.

These pairs really extend the one by Joly and as a conclusion our proof using these pairs had the advantage to extend the proof for the λ-calculus in the sense that to separate $\Lambda\mu$-terms that were also λ-terms, the proof did exactly what is done in λ-calculus and build the same context as Joly's proof. The version of the pair we use here makes the proofs shorter but we loose this property

Proposition 1. *We notice immediately that the following relations hold:*
$\langle u, v\rangle_0 1 \to_h^\star u$ *and* $\langle u, v\rangle_{k+1} M \to_h^\star \langle (u)M, (v)M\rangle_{k+1}$
$\langle u, v\rangle_0 0 \to_h^\star v$ *and* $\langle u, v\rangle_{k+1} \alpha \to_h^\star \langle (u)\alpha, (v)\alpha\rangle_k$
More globally, using the Stream Notation, we get:
$\langle u, v\rangle_k \mathcal{S}_1 \dots \mathcal{S}_k 1 \to_h^\star (u)\mathcal{S}_1 \dots \mathcal{S}_k$ *and* $\langle u, v\rangle_k \mathcal{S}_1 \dots \mathcal{S}_k 0 \to_h^\star (v)\mathcal{S}_1 \dots \mathcal{S}_k$

At this point let us notice that the term $\langle u, v\rangle_k$ is typical of the $\Lambda\mu$-calculus: we cannot build it in $\lambda\mu$-calculus as soon as $k \geq 1$ since it is essentially a succession of Stream Abstractions on the one hand and a succession of Stream Applications on the other hand[14]. This construction will have a key role in our proof.

Notation 2 (The $\mathbb{1}^k$ Stream) *In the following, we write $\mathbb{1}^k$ for the stream $(1^k, \alpha)$. Such a stream will be used intensively.*

Lemma 3 (Parametric Pairs Lemma (PPL)) *Let τ, u be $\Lambda\mu$-terms and $x, y \in \mathcal{V}_t$ with $x \notin FV(u)$. If $\tau[u/x] \to_h^\star y$, then we have:*
$\exists K \in \mathbb{N}$ *such that* $\forall k \geq K, \forall v \in \Lambda\mu$,
  $\exists n^0$ *such that* $\forall (n_1, \dots, n_{k+1})$ *all greater or equal to* n^0,
    $\exists (l_1, \dots, l_{k+1}) \in \mathbb{N}^{k+1}$ *such that:* $\tau[\langle u, v\rangle_k/x]\mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k+1}} \to_h^\star (y)\mathbb{1}^{l_1} \dots \mathbb{1}^{l_{k+1}}$.

The lemma asserts that if $\tau[u/x]$ reduces to the variable y, then for any integer k big enough and for any term v, the parametric pair $\langle u, v\rangle_k$ will behave the same way as u does in the term τ as long as we feed it with enough streams $\mathbb{1}^n$ big enough. The intuitive idea is that one can replace the term u by a term carrying more information but that still behaves the same way.

Definition 13 (Subterm Condition). *Two $\Lambda\mu$-terms M and N are said to satisfy the Subterm Condition if applied to a MAC, they reduce to $M' = (x)\mathcal{S}_1 \dots \mathcal{S}_s M_1 \dots M_m$ and $N' = (y)\mathcal{P}_1 \dots \mathcal{P}_p N_1 \dots N_n$ which are such that:*
• *$x \neq y$ or $s \neq p$ or $m \neq n$*
• *or there is a pair (i, j) both in $\mathcal{P}(M')$ and in $\mathcal{P}(N')$ such that the subterms of M' and N' at position (i, j) satisfy the Subterm Condition.*

[14] In particular, those terms will not be typable (a term of the form $\mu\alpha.\lambda x.M$ is never typable), and it is worth being noticed that the construction $\mu\alpha[\beta]$ is in some sense edicted by typing consideration: the $\lambda\mu$-calculus has been at first designed as a typed calculus...
The non-typability of the terms $\langle u, v\rangle_k$ is not problematic for us since we are working in the untyped setting in which the separation theorems have to be looked for.

Notice that the Subterm condition implies that the terms are not equivalent, the converse being false: usually non-equivalent terms in CNFΛ do not satisfy the Subterm condition as shown by the terms $(x)y\alpha$ and $(x)\alpha$. The ν-transformation deals with this.

Definition 14 (ν-transformation). *Given a closed $\Lambda\mu$-term M we define $\lceil M \rceil^{\nu}$ as the result of the application of a ν-rule to each μ-abstraction of M:*

$\lceil x \rceil^{\nu} = x$ $\quad$ $with\ \ (\alpha \mapsto v_\alpha) : \mathcal{V}_s \to \mathcal{V}_t \setminus V_t(M)$

$\lceil \lambda x.M \rceil^{\nu} = \lambda x.\lceil M \rceil^{\nu}$ $\quad$ $\lceil (M)N \rceil^{\nu} = (\lceil M \rceil^{\nu})\lceil N \rceil^{\nu}$

$\lceil (M)\alpha \rceil^{\nu} = (\lceil M \rceil^{\nu})v_\alpha\alpha$ $\quad$ $\lceil \mu\alpha.M \rceil^{\nu} = \lambda v_\alpha.\mu\alpha.\lceil M \rceil^{\nu}$

Lemma 4 (Subterm Lemma) *Given two closed canonical normal forms that are not $\beta\eta\beta_s\eta_s fst$-equivalent, their ν-transforms satisfy the Subterm Condition.*

Proof sketch: The basic idea of the proof is that in the cases mentioned in section 5.1: $(x)M_1 \dots M_m\alpha$ and $(x)N_1 \dots N_n\beta$ with either $m \neq n$ or $\alpha \neq \beta$ then since the terms we consider are ν-transformed we have $M_m = v_\alpha$ and $N_n = v_\beta$ so that if $m \gtrsim n$ then $N_n = v_\beta \neq M_n$ and if $m = n$ then $N_n = v_\beta \neq v_\alpha = M_m$. Notice that in the lemma, the hypothesis of closed terms is only used to allow the ν-transformation of the terms. □

5.3 The main result

Theorem 5 (Separation Theorem for the $\Lambda\mu$-calculus). *Let M and N be closed $\Lambda\mu$-terms in canonical normal form which are not $\beta\eta\beta_s\eta_s fst$-equivalent. There exists an applicative context $C = []\mathcal{S}_1 \dots \mathcal{S}_k P_1 \dots P_l$ such that $C[M] \to_h^\star 1$ while $C[N] \to_h^\star 0$.*

Proof. We will in fact separate the ν-transforms $\lceil M \rceil^{\nu}$ and $\lceil N \rceil^{\nu}$ of M and N respectively which will be easier since by lemma 4 they satisfy the Subterm Condition. It is easily checked that an applicative separating context for those terms is also a separating context for M and N.

We prove the result by induction on the sum of the sizes of the terms that we consider and by case on their structure. We reason on the open case on terms satisfying the Subterm Condition. The shape of the contexts we will build is: $\mathcal{C} = []\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{\mathcal{S}}\,\overrightarrow{P}$.

First case: Both terms M and N are of the form $(x)\mathcal{P}_1 \dots \mathcal{P}_p Q_1 \dots Q_q$.
Let $M = (x)\mathcal{M}_1 \dots \mathcal{M}_m M_1 \dots M_{m'}$ and $N = (y)\mathcal{N}_1 \dots \mathcal{N}_n N_1 \dots N_{n'}$.

The only interesting case[15] is when $x = y = s$ and $(m, m') = (n, n')$. In that case, since the pair (M, N) satisfies the Subterm condition, we have a pair of integers (i, j) such that $M_{ij} \neq_{\beta\eta\beta_s\eta_s fst} N_{ij}$ and such that M_{ij} and N_{ij} satisfy the Subterm Condition. By induction hypothesis on (M_{ij}, N_{ij}), there are $(U_x)_{x \in FV(M_{ij}N_{ij})}$ and $\overrightarrow{V} = \mathcal{V}_1 \dots \mathcal{V}_v V_1 \dots V_{v'}$ such that $M_{ij}\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{V} \to_h^\star 1$ and $N_{ij}\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{V} \to_h^\star 0$ so that for all variables z_1, z_2 we have $M_{ij}\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{V}\, z_1 z_2 \to_h^\star z_1$ and $N_{ij}\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{V}\, z_1 z_2 \to_h^\star z_2$.

[15] in the other cases, separation is direct by choosing an appropriate context

If we write M' and N' for $M_{ij}\left[\overrightarrow{U}/\overrightarrow{x}, x \neq s\right]\overrightarrow{V}z_1z_2$ and $N_{ij}\left[\overrightarrow{U}/\overrightarrow{x}, x \neq s\right]\overrightarrow{V}z_1z_2$ and provided $z_1, z_2 \neq s$, we get $M'[U_s/s] \rightarrow_h^\star z_1$ and $N'[U_s/s] \rightarrow_h^\star z_2$.

At this point, we need to use the Parametric Pairs Lemma in order to pass two terms to the head variable s at once: on the one hand, we need a term Φ whose role will be to select the subterms located in position (i,j) in M and N[16] and on the other hand a term whose role will be to separate the terms M_{ij} and N_{ij}.

By the PPL applied to M' and to N' with U_s and s, we get K such that for all $k \geq K$ and for all $(k_1, \ldots, k_{k+1})$ big enough, there are $(l_1, \ldots, l_{k+1})$ and $(l'_1, \ldots, l'_{k+1})$ such that :
$M'[\langle U_s, \Phi\rangle_k/s]\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}} \rightarrow_h^\star (z_1)\mathbb{1}^{l_1}\ldots\mathbb{1}^{l_{k+1}}$
$N'[\langle U_s, \Phi\rangle_k/s]\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}} \rightarrow_h^\star (z_2)\mathbb{1}^{l'_1}\ldots\mathbb{1}^{l'_{k+1}}$
This can be also written (with U'_x equal to U_x except in the case of U'_s where it is equal to $\langle U_s, \Phi\rangle_k$):
$M_{ij}\left[\overrightarrow{U'}/\overrightarrow{x}\right]\overrightarrow{V}z_1z_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}} \rightarrow_h^\star (z_1)\mathbb{1}^{l_1}\ldots\mathbb{1}^{l_{k+1}}$ and
$N_{ij}\left[\overrightarrow{U'}/\overrightarrow{x}\right]\overrightarrow{V}z_1z_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}} \rightarrow_h^\star (z_2)\mathbb{1}^{l'_1}\ldots\mathbb{1}^{l'_{k+1}}$

Let us now define $w_1 = \mu\alpha_1\ldots\mu\alpha_{k+1}.1$ and $w_2 = \mu\alpha_1\ldots\mu\alpha_{k+1}.0$ and consider:
$M\left[\overrightarrow{U'}/\overrightarrow{x}\right]\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots 0\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}}$
$= \langle U_s, \Phi\rangle_k\mathcal{M}'_1\ldots\mathcal{M}'_m M'_1\ldots M'_{m'}\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots 0\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}}$
and the corresponding term for N.

We can now display a derivation from $\mathcal{C}(M)$ to 1 in this case: this is done below[17]. The derivation from $\mathcal{C}(N)$ to 0 is similar.

$M\left[\overrightarrow{U'}/\overrightarrow{x}\right]\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k-m-v}}0\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}}$
$\rightarrow_h^\star \langle\star, (\Phi)\mathcal{M}'_1\ldots\mathcal{M}'_m {M'}^{1}_{m+1}\ldots {M'}^{m'}_{m+1}\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k-m-v}}\rangle_0 0\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}}$
$\rightarrow_h^\star (\Phi)\mathcal{M}'_1\ldots\mathcal{M}'_m {M'}^{1}_{m+1}\ldots {M'}^{m'}_{m+1}\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}}$
$\rightarrow_h^\star (M'_{ij})\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}} \rightarrow_h^\star (w_1)\mathbb{1}^{l_1}\ldots\mathbb{1}^{l_{k+1}} \rightarrow_h^\star 1$

Second case: At least one of M and N is not of the form $(x)\mathcal{P}_1\ldots\mathcal{P}_p Q_1\ldots Q_q$.

By putting the terms in a Minimal Applicative Context, they reduce to terms M' and N' of the expected form that satisfy the Subterm Condition and $\mathcal{T}(M)+\mathcal{T}(N) \geq \mathcal{T}(M')+\mathcal{T}(N')$, so that it is possible to apply the induction hypothesis. □

5.4 Separating David & Py's counter-example

As an illustration of our result, we briefly exhibit a separating context for the terms W_0 and W_1 that induced failure of the Separation Property in David & Py's paper. We will show how it is possible to make the term $W = \lambda x.\mu\alpha.((x)\mu\gamma.((x)U_0 y\alpha)U_0\alpha)$ reduce to the variable y in a context $\mathcal{C}$, so that W_0 would reduce to 0 and W_1 to 1 in this context.

[16] That will be $\Phi^{\mathcal{P}(M)}_{(i,j)}$. Note that by hypothesis we have $\mathcal{P}(M) = \mathcal{P}(N)$.
[17] The left component of the pair is not shown.

Writing $\mathcal{P}$ for the parametric pair: $\langle \lambda z_0, z_1.\mu\alpha.\mu\beta.z_1, \lambda x.\mu\alpha.x\rangle_1$, the separating context[18] is $\mathcal{C} = []\mathcal{P}x_0x_1\alpha 0\mathbb{1}^{k_0}\mathbb{1}^{k_1}$ with k_0 and k_1 both greater than 1.

The separation process is described in the following derivation:

$\mathcal{C}(W) = (W)\mathcal{P}x_0x_1\alpha\ 0\mathbb{1}^{k_0}\ \mathbb{1}^{k_1} \rightarrow_h^\star (\mathcal{P})\ \mu\gamma.((\mathcal{P})\ U_{x_1}yx_0x_1\alpha)\ U_{x_1}x_0x_1\alpha\ 0\mathbb{1}^{k_0}\ \mathbb{1}^{k_1}$
$\rightarrow_h^\star \langle\star, (\lambda x.\mu\alpha.x)\ \mu\gamma.((\mathcal{P})\ U_{x_1}yx_0x_1\alpha)\ U_{x_1}x_0x_1\alpha\rangle_0\ 0\mathbb{1}^{k_0}\ \mathbb{1}^{k_1}$
$\rightarrow_h^\star (\lambda x.\mu\alpha.x)\ \mu\gamma.((\mathcal{P})\ U_{x_1}yx_0x_1\alpha)\ U_{x_1}x_0x_1\alpha\ \mathbb{1}^{k_0}\ \mathbb{1}^{k_1}$
$\rightarrow_h^\star (\mu\gamma.(\mathcal{P})\ U_{x_1}yx_0x_1\alpha)\ \mathbb{1}^{k_0}\ \mathbb{1}^{k_1} \rightarrow_h^\star (\mathcal{P})\ U_{x_1}yx_0x_1\alpha\ \mathbb{1}^{k_1}$
$\rightarrow_h^\star \langle(\lambda z_0, z_1.\mu\alpha.\mu\beta.z_1)\ U_{x_1}yx_0x_1\alpha, \star\rangle_0\ 1\mathbb{1}^{k_1-1}$
$\rightarrow_h^\star (\lambda z_0, z_1.\mu\alpha.\mu\beta.z_1)\ U_{x_1}yx_0x_1\alpha\ \mathbb{1}^{k_1-1}$
$\rightarrow_h^\star y$

6 Conclusion

Separation is an important property dealing both with syntax and semantics of languages. The way we went through to recover Separation means that the reductions are not responsible for the failure of Separation, the cause of the failure being that the syntax does not allow enough contexts to interact with the terms.

Moreover this result stresses the difference between $\lambda\mu$-calculus à la Parigot and the alternative syntax, $\Lambda\mu$-calculus: although these two calculi were considered by most authors to be essentially the same (some authors work with Parigot's presentation, some with the alternative one, all call this $\lambda\mu$-calculus but the relation between them is never made clear). We showed here that the difference between $\lambda\mu$-calculus and $\Lambda\mu$-calculus lies in an intrinsic lack of extensionality in $\lambda\mu$-calculus à la Parigot because of the $\mu\alpha.[\beta]$ construction. This fact is easily explained when one considers that $\lambda\mu$-calculus has been designed as a typed calculus at the beginning and that the $\mu\alpha.[\beta]$ construction can be seen as a typing constraint made syntactical: under a μ-abstraction the named term will always have type $\perp$ (if looking at classical logic with falsity).

Furthermore we exhibited a strong connection between $\Lambda\mu$-calculus and the structure of stream: the λ-abstraction abstracts over terms while the μ-abstraction abstracts over streams of terms. This presentation as a stream calculus is interesting for several reasons: first of all it gives a very clear computational interpretation to $\Lambda\mu$-calculus, in addition this structure is deeply rooted in the shape of $\Lambda\mu$-terms (and this fact is confirmed by the form the abstract machines for $\lambda\mu$-calculus [Lau03a,Bie98,dG98] usually take) as mentioned when discussing the *fst*-rule. Moreover it makes rules nicer than the ones of the $\lambda\mu$-calculus especially suggesting a rule of the form $(\mu\alpha.M)\mathcal{S} \rightarrow M[\mathcal{S}/\alpha]$ and in this case the notion of substitution is much more uniform than the one of $\lambda\mu$-calculus.

This work suggests several future developments: the study of $\Lambda\mu$ as a stream calculus with streams as first-class citizens (in the present paper we staid close to the usual presentation of $\lambda\mu$ introducing streams only as a convenient notation). Another direction of study is about confluence: confluence properties of $\lambda\mu$-calculus are highly non-trivial especially when introducing the η-reduction. This is solved by adding the

[18] This context is a simplified version of the context the can be extracted from the proof but it has the same shape. k_0 could be set to 0.

ν-rule which contain ... an η-expansion! It would be interesting to study $\Lambda\mu$-calculus with expansion rules instead of reduction rules for the extensionality rules.

Acknowledgments: I wish to thank Pierre-Louis Curien for his constant advice during this work and an anonymous referee for useful comments and suggestions of simplification. I also would like to thank Olivier Laurent, Thierry Joly and Dale Miller for valuable discussions regarding this work.

References

[AH03] Zena Ariola and Hugo Herbelin. Minimal classical logic and control operators. In *ICALP'03*, volume 2719 of *LNCS*. Springer, 2003.

[Bar84] Henk Barendregt. *The Lambda Calculus, its Syntax and Semantics*. Studies in Logic and the Foundations of Mathematics. Elsevier publishers, 1984.

[Bie98] Gavin Bierman. A computational interpretation of the $\lambda\mu$-calculus. In *MFCS'98*, volume 1450 of *LNCS*, pages 336–345, 1998.

[Böh68] Corrado Böhm. Alcune proprietà delle forme $\beta\eta$-normali nel λk-calcolo. *Publicazioni dell'Istituto per le Applicazioni del Calcolo*, 696, 1968.

[dG94] Philippe de Groote. On the relation between the lambda-mu-calculus and the syntactic theory of sequential control. In *LPAR'94*, volume 822 of *LNAI*, pages 31–43, 1994.

[dG98] Philippe de Groote. An environment machine for the $\lambda\mu$-calculus. *Math. Str. in Computer Science*, 8:637–669, 1998.

[DP01] René David and Walter Py. $\lambda\mu$-calculus and Böhm's theorem. *Journal of Symbolic Logic*, 66(1):407–413, 2001.

[FH92] Matthias Felleisen and Robert Hieb. The revised report on the syntactic theories of sequential control and state. *Theoretical Computer Science*, 103(2):235–271, september 1992.

[Gir91] Jean-Yves Girard. A new constructive logic: classical logic. *Math. Str. in Computer Science*, 1(3):255–296, 1991.

[Gri90] Timothy Griffin. A formulae-as-types notion of control. In *POPL'90*, pages 47–58, 1990.

[Jol00] Thierry Joly. *Codages, séparabilité et représentation de fonctions en λ-calcul simplement typé et dans d'autres systèmes de types*. PhD thesis, Université Paris VII, 2000.

[Kri93] Jean-Louis Krivine. *Lambda-calculus, Types and Models*. Ellis Horwood, 1993.

[Lau03a] Olivier Laurent. Krivine's abstract machine and the $\lambda\mu$-calculus (an overview). unpublished note, september 2003.

[Lau03b] Olivier Laurent. Polarized proof-nets and $\lambda\mu$-calculus. *Theoretical Computer Science*, 290(1):161–188, 2003.

[OS97] Luke Ong and Charles Stewart. A Curry-Howard foundation for functional computation with control. In *POPL'97*, pages 215–227, 1997.

[Par92] Michel Parigot. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *International Conference on Logic Programming and Automated Deduction*, volume 624 of *LNCS*, pages 190–201. Springer, 1992.

[Par97] Michel Parigot. Proofs of strong normalisation for second order classical natural deduction. *Journal of Symbolic Logic*, 62(4):1461–1479, december 1997.

[Py98] Walter Py. *Confluence en $\lambda\mu$-calculus*. PhD thesis, Université de Savoie, 1998.

7 Appendix

In this Appendix, we give the proofs of some Lemmas that we have not proved in the paper and we give a proof of the Separation theorem which includes some cases non-addressed in the main part of the paper.

7.1 Proof of the Parametric Pairs Lemma

Here is a proof of the PPL:

Proof. We prove the lemma by induction on the length of the head reduction $\tau[u/x] \rightarrow_h^\star y$ and by case on τ.

First case: τ is not in head normal form.
In this case we have $\tau \rightarrow_h \tau'$ for some $\tau' \in \Lambda\mu$, and subsequently $\tau[u/x] \rightarrow_h \tau'[u/x] \rightarrow_h^\star y$ since τ is not in *hnf*. By induction hypothesis on τ', we have the result.

Second case: τ is in head normal form.
$\tau = (z)\mathcal{S}_1 \dots \mathcal{S}_p N_1 \dots N_q$ and we have 3 sub-cases:
• $z = y$, the result is immediately true since $\tau = y$.
• $z = x$ and $\tau = x$, then we have $u \rightarrow_h^\star y$ and for any $(n_1, \dots, n_{k+1})$ all greater than 1:
$\langle u, v\rangle_k \mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k+1}} \rightarrow_h^\star \langle (u)\mathbb{1}^{n_1} \dots \mathbb{1}^{n_k}, (v)\mathbb{1}^{n_1} \dots \mathbb{1}^{n_k}\rangle_0 1 \mathbb{1}^{n_{k+1}-1} \rightarrow_h^\star$
$(u)\mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k+1}-1} \rightarrow_h^\star (y)\mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k+1}-1}$
• $z = x$ and $\tau = (x)\mathcal{S}_1 \dots \mathcal{S}_p N_1 \dots N_q$, with $(p, q) \neq (0, 0)$, and thus $\tau[u/x] \rightarrow_h^+ y$.
In the following, we will abbreviate $M[u/x]$ as M' and we extend straightforwardly this notation to stream applications so that $\tau[u/x]$ is $(u)\mathcal{S}'_1 \dots \mathcal{S}'_p N'_1 \dots N'_q$.

Notice that $\tau[u/x]$ cannot be in head normal form and let us note τ' the $\Lambda\mu$-term $(u)\mathcal{S}_1 \dots \mathcal{S}_p N_1 \dots N_q$. Since x is not among the free variables of u, the following equality holds: $\tau'[u/x] = \tau[u/x]$.
The lengths of the reductions associated to τ and τ' are the same (the reductions are the same...) but τ' is not in head normal form so that τ' corresponds to the first case and it is possible to apply the induction hypothesis to τ'.
Let us now consider K satisfying the condition for τ' and let us define K' as $max(K, p+1)$. Take $k \geq K'$ and $v \in \Lambda\mu$. Let k^0 be as required in the statement on τ' and define k'^0 as $k^0 + 1$. Consider now $(n_1, \dots, n_{k+1})$ all greater than k'^0. We have a derivation from $\tau[\langle u, v\rangle_k/x]\mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k+1}}$ to $(y)\mathbb{1}^{l_1} \dots \mathbb{1}^{l_{k+1}}$ for some $(l_1, \dots, l_{k+1})$:
$\tau[\langle u, v\rangle_k/x]\mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k+1}} = (\langle u, v\rangle_k)\mathcal{S}''_1 \dots \mathcal{S}''_p N''_1 \dots N''_q \mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k+1}}$
$\rightarrow_h^\star (\langle (u)\mathcal{S}''_1 \dots \mathcal{S}''_p N''_1 \dots N''_q, \star\rangle_{k-p})\mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k+1}}$
$\rightarrow_h^\star (\langle (u)\mathcal{S}''_1 \dots \mathcal{S}''_p N''_1 \dots N''_q \mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k-p}}, \star\rangle_0) 1 \mathbb{1}^{n_{k-p+1}-1} \dots \mathbb{1}^{n_{k+1}}$
$\rightarrow_h^\star (u)\mathcal{S}''_1 \dots \mathcal{S}''_p N''_1 \dots N''_q \mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k-p+1}-1} \dots \mathbb{1}^{n_{k+1}}$
$= \tau'[\langle u, v\rangle_k/x]\mathbb{1}^{n_1} \dots \mathbb{1}^{n_{k-p+1}-1} \dots \mathbb{1}^{n_{k+1}}$
$\rightarrow_h^\star (y)\mathbb{1}^{l_1} \dots \mathbb{1}^{l_{k+1}}$

The right component of the pair is not made explicit since it has no effect on the derivation. The last step is justified by the validity of the PPL on τ' and the fact that: $k \geq K$ and $(n_1, \ldots, n_{k-p+1} - 1, \ldots, n_{k+1})$ are all greater than k^0. The double primed terms denote the fact that the substitution is now $[\langle u, v\rangle_k/x]$ instead of $[u/x]$. □

7.2 Complete proof of the Separation theorem

We give here the complete proof of the Separation Theorem, including the cases excluded from the presentation of the proof in the paper.

Proof. We will in fact separate the ν-transforms $\lceil M \rceil^\nu$ and $\lceil N \rceil^\nu$ of M and N respectively which will be easier since by lemma 4 they satisfy the Subterm Condition. It is easily checked that an applicative separating context for those terms is also a separating context for M and N.

We prove the result by induction on the sum of the sizes of the terms that we consider and by case on their structure. We reason on the open case on terms satisfying the Subterm Condition. The shape of the contexts we will build is: $\mathcal{C} = []\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{\mathcal{S}}\,\overrightarrow{P}$.

First case: Both terms M and N are of the form $(x)\mathcal{P}_1 \ldots \mathcal{P}_p Q_1 \ldots Q_q$.
Let $M = (x)\mathcal{M}_1 \ldots \mathcal{M}_m M_1 \ldots M_{m'}$ and $N = (y)\mathcal{N}_1 \ldots \mathcal{N}_n N_1 \ldots N_{n'}$.

We reason by case analysis on M and N:

Case 1: $x \neq y$
Then, with $U_x = \mu\alpha_1 \ldots \mu\alpha_m.\lambda y_1 \ldots . \lambda y_{m'}.1$ and $U_y = \mu\alpha_1 \ldots \mu\alpha_n.\lambda y_1 \ldots . \lambda y_{n'}.0$, we have $M[U_x/x; U_y/y] \rightarrow_h^\star 1$ while $N[U_x/x; U_y/y] \rightarrow_h^\star 0$.

Case 2: $x = y = s$ **but** $(m, m') \neq (n, n')$
We must distinguish two sub-cases:

• $m = n$ and $m' \neq n'$. Let us consider the case $m' \lneq n'$.
With $U_s = \mu\alpha_1 \ldots \mu\alpha_m.\lambda z_1 \ldots . \lambda z_{n'+1}.z_{n'+1}$ and $\overrightarrow{V}$ the applicative context $\{\lambda z_1 \ldots . \lambda z_{n'-m'}.1, 0^{n'-m'}\}$, we have $M[U_s/s]\overrightarrow{V} \rightarrow_h^\star (\lambda z_1 \ldots \lambda z_{n'-m'}.1)0^{n'-m'} \rightarrow_h^\star 1$ while $N[U_s/s]\overrightarrow{V} \rightarrow_h^\star 0$ and we get the expected result.

• $m \neq n$, for instance $m \lneq n$.
With $U_s = \mu\gamma_1 \ldots \mu\gamma_n.\lambda z_1 \ldots \lambda z_{n'+1}.z_{n'+1}$ and $\overrightarrow{V} = \{\mu\gamma_1 \ldots \mu\gamma_{n-m}.\lambda z_1 \ldots \lambda z_{n'+1}.0, \gamma_1, \ldots, \gamma_{n-m}, 1^{n'+1}\}$ we get the following reduction :

$$
\begin{aligned}
&M[U_s/s]\overrightarrow{V} = (U_s)\mathcal{M}'_1 \ldots \mathcal{M}'_m {M'}_1 \ldots {M'}_{m'} \\
&\quad (\mu\gamma_1 \ldots \mu\gamma_{n-m}.\lambda z_1 \ldots \lambda z_{n'+1}.0)\gamma_1 \ldots \gamma_{n-m} 1^{n'+1} \\
&\rightarrow_h^\star (\mu\gamma_{m+1} \ldots \mu\gamma_n.\lambda z_1 \ldots \lambda z_{n'+1}.z_{n'+1}){M'}_1 \ldots {M'}_{m'} \\
&\quad (\mu\gamma_1 \ldots \mu\gamma_{n-m}.\lambda z_1 \ldots \lambda z_{n'+1}.0)\gamma_1 \ldots \gamma_{n-m} 1^{n'+1} \\
&\rightarrow_h^\star (\lambda z_1 \ldots . \lambda z_{n'+1}.z_{n'+1})1^{n'+1} \rightarrow_h^\star 1
\end{aligned}
$$

and on the other hand we have $N[U_s/s]\overrightarrow{V} \rightarrow_h^\star 0$.

Case 3: $x = y = s$ **and** $(m, m') = (n, n')$
In that case, since the pair (M, N) satisfies the Subterm condition, we have a pair of integers (i, j) such that $M_{ij} \neq_{\beta\eta\beta_s\eta_s fst} N_{ij}$ and such that M_{ij} and N_{ij} satisfy the Subterm Condition. By induction hypothesis on (M_{ij}, N_{ij}), there are $(U_x)_{x \in FV(M_{ij}N_{ij})}$ and $\overrightarrow{V} = \mathcal{V}_1 \ldots \mathcal{V}_v V_1 \ldots V_{v'}$ such that $M_{ij}\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{V} \rightarrow_h^\star 1$ and $N_{ij}\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{V} \rightarrow_h^\star 0$

so that for all variables z_1, z_2 we have $M_{ij}\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{V}z_1z_2 \to_h^\star z_1$ and $N_{ij}\left[\overrightarrow{U}/\overrightarrow{x}\right]\overrightarrow{V}z_1z_2 \to_h^\star z_2$.

If we write M' and N' for $M_{ij}\left[\overrightarrow{U}/\overrightarrow{x}, x \neq s\right]\overrightarrow{V}z_1z_2$ and $N_{ij}\left[\overrightarrow{U}/\overrightarrow{x}, x \neq s\right]\overrightarrow{V}z_1z_2$ and provided $z_1, z_2 \neq s$, we get $M'[U_s/s] \to_h^\star z_1$ and $N'[U_s/s] \to_h^\star z_2$.

At this point, we need to use the Parametric Pairs Lemma in order to pass two terms to the head variable s at once: on the one hand, we need a term whose role will be to select the subterms located in position (i,j) in M and N[19] and on the other hand a term whose role will be to separate the terms M_{ij} and N_{ij}.

By the PPL applied to M' and to N' with U_s and s, we get K such that for all $k \geq K$ and for all $(k_1, \ldots, k_{k+1})$ big enough, there are $(l_1, \ldots, l_{k+1})$ and $(l'_1, \ldots, l'_{k+1})$ such that :
$M'[\langle U_s, \Phi\rangle_k/s]\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}} \to_h^\star (z_1)\mathbb{1}^{l_1}\ldots\mathbb{1}^{l_{k+1}}$
$N'[\langle U_s, \Phi\rangle_k/s]\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}} \to_h^\star (z_2)\mathbb{1}^{l'_1}\ldots\mathbb{1}^{l'_{k+1}}$
This can be also written (with U'_x equal to U_x except in the case of U'_s where it is equal to $\langle U_s, \Phi\rangle_k$):
$M_{ij}\left[\overrightarrow{U'}/\overrightarrow{x}\right]\overrightarrow{V}z_1z_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}} \to_h^\star (z_1)\mathbb{1}^{l_1}\ldots\mathbb{1}^{l_{k+1}}$ and
$N_{ij}\left[\overrightarrow{U'}/\overrightarrow{x}\right]\overrightarrow{V}z_1z_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}} \to_h^\star (z_2)\mathbb{1}^{l'_1}\ldots\mathbb{1}^{l'_{k+1}}$

Let us now define $w_1 = \mu\alpha_1\ldots\mu\alpha_{k+1}.1$ and $w_2 = \mu\alpha_1\ldots\mu\alpha_{k+1}.0$ and consider:
$M\left[\overrightarrow{U'}/\overrightarrow{x}\right]\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots 0\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}}$
$= \langle U_s, \Phi\rangle_k\mathcal{M}'_1\ldots\mathcal{M}'_m M'_1\ldots M'_{m'}\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots 0\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}}$
and the corresponding term for N.

We can now display a derivation from $\mathcal{C}(M)$ to 1 in this case: this is done below[20]. The derivation from $\mathcal{C}(N)$ to 0 is similar.

$M\left[\overrightarrow{U'}/\overrightarrow{x}\right]\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k-m-v}}0\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}}$
$\to_h^\star \langle\star, (\Phi)\mathcal{M}'_1\ldots\mathcal{M}'_m M'^{\,1}_{m+1}\ldots M'^{\,m'}_{m+1}\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k-m-v}}\rangle_0 0\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}}$
$\to_h^\star (\Phi)\mathcal{M}'_1\ldots\mathcal{M}'_m M'^{\,1}_{m+1}\ldots M'^{\,m'}_{m+1}\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k+1}}$
$\to_h^\star (M'_{ij})\overrightarrow{V}w_1w_2\mathbb{1}^{k_1}\ldots\mathbb{1}^{k_{k-m-v+1}}\ldots\mathbb{1}^{k_{k+1}} \to_h^\star (w_1)\mathbb{1}^{l_1}\ldots\mathbb{1}^{l_{k+1}} \to_h^\star 1$

Second case: At least one of M and N is not of the form $(x)\mathcal{P}_1\ldots\mathcal{P}_p Q_1\ldots Q_q$.

By putting the terms in a Minimal Applicative Context, they reduce to terms M' and N' of the expected form that satisfy the Subterm Condition and $\mathcal{T}(M) + \mathcal{T}(N) \geq \mathcal{T}(M') + \mathcal{T}(N')$, so that it is possible to apply the induction hypothesis. □

[19] That will be $\Phi_{(i,j)}^{\mathcal{P}(M)}$. Note that by hypothesis we have $\mathcal{P}(M) = \mathcal{P}(N)$.

[20] The left component of the pair is not shown.