The untyped stack calculus and Böhm's theorem

Alberto Carraro

PPS, Université Paris Diderot, France

`acarraro@pps.univ-paris-diderot.fr`

The stack calculus is a functional language in which is in a Curry-Howard correspondence with classical logic. It enjoys confluence but, as well as Parigot's $\lambda\mu$, does not admit the Böhm Theorem, typical of the λ-calculus. We present a simple extension of stack calculus which is for the stack calculus what Saurin's $\Lambda\mu$ is for $\lambda\mu$.

1 Introduction

In [3] Corrado Böhm proved the so-called *Böhm's theorem*, a fundamental syntactical feature of the pure λ-calculus which states that if M and N are two distinct $\beta\eta$-normal terms, then for each pair of terms P, Q there exists a context $C[\cdot]$ such that $C[M]$ is β-equivalent to P and $C[N]$ is β-equivalent to Q. If moreover M and N are closed, the context $C[\cdot]$ can have shape $[\cdot]\vec{L}$, for a suitable sequence $\vec{L} = L_1, \ldots, L_k$ of λ-terms. The original issue motivating this result was the quest for solutions of systems of equations between λ-terms: given closed λ-terms $M_1, N_1, \ldots, M_n, N_n$, is there a λ-term S such that $SM_1 =_\beta N_1 \wedge \cdots \wedge SM_n =_\beta N_n$ holds? The answer is trivial for $n = 1$ (just take $S = \lambda z.N_1$ for a fresh variable z) and Böhm's theorem gives a positive answer for $n = 2$ when M_1, M_2 are distinct $\beta\eta$-normal forms (apply the theorem to M_1 and M_2 and then set $S = \lambda f.f\vec{L}N_1N_2$). The result has been then generalized (and this step is non-trivial) in [4] to treat every finite family $M_1, \ldots, M_n$ of pairwise distinct $\beta\eta$-normal forms.

The notion of *operational equivalence* has been a subject of many research works in the literature. Essentially one considers as "equivalent" two λ-terms M and N when for every possible context $C[\cdot]$ the head reduction process of $C[M]$ halts iff the head reduction process of $C[N]$ halts. It is common to think of two non-operationally equivalent terms as programs that can be distinguished one another by making them interact with all possible environments, observing termination of head reduction. From Böhm's theorem it follows that given two distinct $\beta\eta$-normal forms one can choose a term P that admits a head normal form and a term Q does not have a head normal form, having the guarantee that there exists a context $C[\cdot]$ such that $C[M] =_\beta P$ and $C[N] =_\beta Q$. In this sense M and N would be "separated" (or distinguished) by the context $C[\cdot]$ witnessing that M and N are not operationally equivalent. For this reason Böhm's theorem is also known as the *separation theorem* and it is said to prove the *separation property* for the untyped λ-calculus.

The separation property has consequences both on the semantical and on the syntactical side. For example it implies that $\beta\eta$-equivalence is the maximal non-trivial congruence on normalizable terms extending the β-equivalence, so that any model of the λ-calculus cannot identify two different $\beta\eta$-normal forms without being trivial. A possible reading of Böhm's theorem is that the λ-calculus is powerful enough to "inspect" itself and the syntax and the reduction rules fit each other well. Nonetheless Böhm's result gives an alternative characterization of operational equivalence for normalizable terms. The complete characterization operational equivalence (also for non-normalizable terms) was then achieved by Hyland [17] and Wadsworth [28]. In general two λ-terms are operationally equivalent iff they have the same Böhm tree, up to possibly infinite η-expansion iff they have the same denotation in Scott's D_∞

model (see also [2]). Böhm's proof shows how to produce the separating context with an algorithm whose inputs are M, N ($\beta\eta$-normal terms) P and Q (arbitrary terms). Following this observation Huet [16] shows an ML implementation of Böhm's algorithm and poses the problem of formalizing a proof of Böhm's theorem for the purpose of mechanical checking. The combinatorial core of Böhm's algorithm is called the *Böhm-out technique* and it is at the basis of the implementation, presented in [6], of the CUCH-machine, a λ-calculus interpreter introduced by Böhm and Gross in [5]. Various generalizations/extensions of Böhm's theorem have been studied. The λ-calculus has been immersed in other languages in order to obtain finer observations on the behaviour of λ-terms. Sangiorgi [22] considers the encoding of the λ-calculus in the π-calculus with the addition of a unary non-deterministic operator. In [10] and [11] Dezani et al. add a binary parallel operator and a non-deterministic choice. Manzonetto and Pagani [19] give a proof of Böhm's theorem for a resource-sensitive extension of the λ-calculus.

Curry-Howard correspondence [15] was first stated as the isomorphism between natural deduction for minimal intuitionistic logic [21] and the simply typed λ-calculus. Later Griffin [13] proposed that natural deduction for classical logic could be viewed as a type system for a λ-calculus with a certain control operator introduced by Felleisen [12]. Several other proposals have been made for a computational interpretation of classical logic, among which Parigot's $\lambda\mu$-calculus [20] received a lot of attention. In [8] David and Py proved that the $\lambda\mu$-calculus does not satisfy the separation property. Saurin [23] exhibited an extension of $\lambda\mu$-calculus, the $\Lambda\mu$-calculus, in which the separation property does hold. In [7] the authors introduce the stack calculus, a finitary functional language in which the $\lambda\mu$-calculus can be faithfully translated.

In the present paper we prove, using David and Py's [8] counterexample, that the separation property does not hold for the stack calculus, i.e., there are (extensionally) different normal forms which are operationally equivalent. We introduce the extended stack calculus which is a calculus that contains the stack calculus (as Saurin's $\Lambda\mu$ extends Parigot's $\lambda\mu$). We show that operational equivalence is maximally consistent, i.e. it cannot be properly extended to another consistent equational theory both in the stack calculus and in the extended stack calculus. We work out the details of a Böhm-out technique for the extended stack calculus. A nice feature of the extended stack calculus is that, having only one binder (instead of two as in $\lambda\mu$), it admits a simpler proof of Böhm's theorem, which is similar to the one for the λ-calculus.

The treatment of Böhm's theorem deserves a prominent place in classical monographs on the λ-calculus (Hindley–Seldin [26], Hankin [14], Barendregt [2]). Besides the applications of Böhm's theorem, there has always been interest around the proof itself and the algorithmic content of Böhm-out technique. In [9] Dezani et al. give a thorough account of Böhm's theorem, together with an overview of the impressive research activity which originated from it. To the best of our knowledge since Huet's challenge [16], no mechanical proof of Böhm's theorem has been produced yet. Instead Aehlig and Joachimski [1] provide a different proof of Böhm's theorem that does not use the Böhm-out technique. In view of the interest in proofs of Böhm's theorem for various calculi, we believe useful to contribute in the present work with a direct proof of Böhm's theorem (i.e. with a Böhm-out technique) for the extended stack calculus, even if the mere separation result would follow by a suitable mutual translation with the $\Lambda\mu$-calculus.

2   The untyped stack calculus

We report the presentation of stack calculus from [7]. The language has three syntactic categories: *terms* that are in functional position, *stacks* that are in argument position and represent streams of arguments,

processes that are terms applied to stacks.

The basis for the definition of the stack calculus language is a countably infinite set of *stack variables*, ranged over by the initial small letters $\alpha, \beta, \gamma, \ldots$ of the greek alphabet. The language is then given by the following grammar:

$$
\begin{array}{lll}
\text{stacks} & \pi, \varpi & ::= \quad \mathsf{nil} \mid \alpha \mid \mathsf{cdr}(\pi) \mid M \textbf{.} \pi \\
\text{terms} & M, N & ::= \quad \mathsf{car}(\pi) \mid \mu\alpha.P \\
\text{processes} & P, Q & ::= \quad M \star \pi
\end{array}
$$

We use letters E, E' to range over *expressions* which are either stacks, terms or processes. The operator μ is a binder. An occurrence of a variable α in an expression E is *bound* if it is under the scope of a $\mu\alpha$; the set $\mathrm{FV}(E)$ of *free variables* is made of those variables having a non-bound occurrence in E.
Stacks represent lists of terms: nil is the empty stack. A stack $M_1 \textbf{.} \cdots \textbf{.} M_k \textbf{.} \mathsf{nil}$, stands for a finite list while a stack $M_1 \textbf{.} \cdots \textbf{.} M_k \textbf{.} \alpha$ stands for a non-terminated list that can be further extended. *Terms* are entities that represent the "active part" of computations. A term $\mu\alpha.P$ is the μ-*abstraction* of α in P.
Processes result from the *application* $M \star \pi$ of a term M to a stack π. This application, unlike in the λ-calculus, has to be thought as *exhaustive*: an application of a stack to a term is an evolving entity that does not have any outcome.

As usual, the calculus involves a substitution operator. By $E\{\pi/\alpha\}$ we denote the substitution of the stack π for all free occurrences of α in E (paying attention to avoid capture of free variables). A basic but useful fact about substitutions is the *substitution lemma* (see [7]): for all $E \in \Sigma^{\mathsf{e}}$ and all $\pi, \varpi \in \Sigma^{\mathsf{s}}$ with $\alpha \notin \mathrm{FV}(\varpi)$ and $\alpha \not\equiv \beta$ we have $E\{\pi/\alpha\}\{\varpi/\beta\} \equiv E\{\varpi/\beta\}\{\pi\{\varpi/\beta\}/\alpha\}$ (the symbol '$\equiv$' stands for syntactic equality).

The reduction rules characterizing the stack calculus are the following ones:

$$
\begin{array}{lll}
(\mu) & (\mu\alpha.P) \star \pi \rightarrow_\mu P\{\pi/\alpha\} \\
(\mathsf{car}) & \mathsf{car}(M \textbf{.} \pi) \rightarrow_{\mathsf{car}} M \\
(\mathsf{cdr}) & \mathsf{cdr}(M \textbf{.} \pi) \rightarrow_{\mathsf{cdr}} \pi
\end{array}
$$

Adding the following rules we obtain the *extensional* stack calculus:

$$
\begin{array}{lll}
(\eta_1) & \mu\alpha.M \star \alpha \rightarrow_{\eta_1} M & \quad \text{if } \alpha \notin \mathrm{FV}(M) \\
(\eta_2) & \mathsf{car}(\pi) \textbf{.} \mathsf{cdr}(\pi) \rightarrow_{\eta_2} \pi
\end{array}
$$

We simply write $\rightarrow_{\mathsf{s}}$ for the contextual closure of the relation $(\rightarrow_\mu \cup \rightarrow_{\mathsf{car}} \cup \rightarrow_{\mathsf{cdr}})$. Moreover we write $\rightarrow_\eta$ for the contextual closure of the relation $(\rightarrow_{\eta_1} \cup \rightarrow_{\eta_2})$ and finally we set $\rightarrow_{\mathsf{s}\eta} = (\rightarrow_{\mathsf{s}} \cup \rightarrow_\eta)$. We denote by $\twoheadrightarrow_{\mathsf{s}}$ (resp. $\twoheadrightarrow_{\mathsf{s}\eta}$) the reflexive and transitive closure of $\rightarrow_{\mathsf{s}}$ (resp. $\rightarrow_{\mathsf{s}\eta}$) and we denote by $=_{\mathsf{s}}$ (resp. $=_{\mathsf{s}\eta}$) the reflexive, symmetric, and transitive closure of $\rightarrow_{\mathsf{s}}$ (resp. $\rightarrow_{\mathsf{s}\eta}$).

An example of term is $\mathbf{I} := \mu\alpha.\mathsf{car}(\alpha) \star \mathsf{cdr}(\alpha)$ (the symbol ':=' stands for definitional equality). For example $\mathbf{I} \star \mathbf{I} \textbf{.} \mathsf{nil} \rightarrow_{\mathsf{s}} \mathbf{I} \star \mathsf{nil} \rightarrow_{\mathsf{s}} \mathsf{car}(\mathsf{nil}) \star \mathsf{cdr}(\mathsf{nil})$ and the reduction does not proceed further. If $\omega := \mu\alpha.\mathsf{car}(\alpha) \star \alpha$, then $\omega \star \omega \textbf{.} \mathsf{nil} \rightarrow_{\mathsf{s}} \omega \star \omega \textbf{.} \mathsf{nil}$; this is an example of a non-normalizing process. The stack calculus enjoys confluence, even in its extensional version, as stated in the following theorem.

Theorem 2.1 ([7]). *The reductions $\rightarrow_{\mathsf{s}}$ and $\rightarrow_{\mathsf{s}\eta}$ are both Church-Rosser.*

It seems natural to define a meta-language with constructions $\pi[n] := \mathsf{cdr}(\cdots \mathsf{cdr}(\pi) \cdots)$ (n times) and $\pi[n] := \mathsf{car}(\pi[n])$. It is easily checkable that every expression E has a $\twoheadrightarrow_{\mathsf{car},\mathsf{cdr}}$-normal form, that we will refer to as the *canonical form* of E. If $\vec{N} = N_1, \ldots, N_m$ is a sequence of terms, we write $\vec{N} \textbf{.} \gamma[k]$ and $\vec{N} \textbf{.} \mathsf{nil}[k]$ for the obvious corresponding stacks. In an expression in canonical form the stacks have either shape $\vec{N} \textbf{.} \gamma[k]$ or $\vec{N} \textbf{.} \mathsf{nil}[k]$ and the non-abstraction terms have shape either $\alpha[n]$ or $\mathsf{nil}[n]$.

2.1   Operational equivalence and failure of Böhm's theorem

We provide a notion of *outer reduction* for the stack calculus, obtained by performing the contraction of outer-most redexes only. The one-step *outer-reduction* on terms is given by the following rule:

$$M \to_o \mu\alpha.P\{\pi/\beta\} \qquad \text{if } \mu\alpha.(\mu\beta.P) \star \pi \text{ is the canonical form of } M$$

Note that outer-reduction is *deterministic* and we don't take any contextual closure for $\to_o$; its reflexive and transitive closure is denoted by $\twoheadrightarrow_o$. A term *M is in outer-normal form* (*onf*, for short) if it is not $\to_o$-reducible. It is straightforward to see that a term is in onf iff it has the form $\mu\alpha.H \star \vec{N}.\tau$, where the terms in the sequence $\vec{N}$ are arbitrary, H is either $\beta[n]$ (in which case we say that the onf is *proper*) or $\text{nil}[n]$ (in which case we say that the onf is *improper*) and τ is either $\gamma(n)$ or $\text{nil}(n)$; H is the head of the onf and τ is the tail of the onf in question. A term *has* a onf if it $\twoheadrightarrow_o$-reduces to a term in onf. Of course the outer reduction strategy is complete for finding onf's of terms, i.e., if $M \twoheadrightarrow_s$-reduces to a term N' in onf, then M reduces to some onf N'' via outer reduction.

A *head context* is a context generated by the grammar $C[\cdot] ::= [\cdot] \mid \mu\alpha.C[\cdot] \star \pi$.

Definition 2.1 (Operational equivalence). *Two terms M, N are* operationally equivalent, *notation $M \approx N$, if for every head context $C[\cdot]$ we have that $C[M]$ has a proper onf iff $C[N]$ has a proper onf.*

Operational equivalence is a fairly common notion. In Definition 2.1 we only quantify over head contexts, but it can be shown (as it is done for the λ-calculus [2] and the $\lambda\mu$-calculus [8]) that a quantification over *all* contexts gives as equivalent definition.

We now define some important terms: $\mathbf{T} := \mu\alpha.\alpha[0] \star \alpha(2)$, $\mathbf{F} := \mu\alpha.\alpha[1] \star \alpha(2)$, and $\Omega := \mu\gamma.\omega \star \omega.\gamma$, where $\omega := \mu\alpha.\alpha[0] \star \alpha$. Clearly Ω is an example of term without onf and we have $\text{nil}[0] \approx \Omega \not\approx \mathbf{T} \not\approx \mathbf{F}$.

We conclude the first part of the paper showing that in the stack calculus there exist different $\to_{s\eta}$-normal forms which are operationally equivalent. This situation is in contrast with the λ-calculus: the original Böhm's theorem [3] implies that two different $\beta\eta$-normal λ-terms M and N are never operationally equivalent because there exists a context $C[\cdot]$ such that $C[M]$ has a head normal form and $C[N]$ does not have a head normal form. However an analogous situation occurs for the $\lambda\mu$-calculus: David and Py [8] exhibited two extentionally different normal $\lambda\mu$-terms which are operationally equivalent. In fact the counterexample given in the next theorem is obtained by translating David and Py's terms into the stack calculus.

Theorem 2.2. *Let $U := \mu\gamma.\alpha[0] \star \alpha$ and $W[\cdot] := \mu\alpha.\alpha[0] \star (\mu\beta.\alpha[0] \star U.[\cdot].\alpha).U.\alpha$. Then for all terms M, N we have $W[M] \approx W[N]$.*

Proof. Let M be a term and let $C[\cdot] := \mu\delta.(\mu\delta_1.\cdots(\mu\delta_m.[\cdot] \star \pi) \star \pi_m \cdots) \star \pi_1$ be a term context. Let

$$M' := M\{\pi_1/\delta_1\}\cdots\{\pi_m/\delta_m\} \qquad\qquad \pi' := \pi\{\pi_1/\delta_1\}\cdots\{\pi_m/\delta_m\}$$
$$U' := U\{\pi'/\alpha\} \qquad\qquad\qquad\qquad \pi'' := (\mu\beta.\pi'[0] \star U'.M'.\pi').U'.\pi'$$

Then $C[W[M]] \twoheadrightarrow_o \mu\delta.\pi'[0] \star \pi''$. At this point we distinguish six possible cases:

(1) $\pi'[0]$ is not an abstraction;

(2) $\pi'[0] \equiv \mu\varepsilon.\text{nil}[n] \star \varpi$;

(3) $\pi'[0] \equiv \mu\varepsilon.\varepsilon'[n] \star \varpi$, with $\varepsilon \not\equiv \varepsilon'$;

(4) $\pi'[0] \equiv \mu\varepsilon.\varepsilon[0] \star \varpi$;

(5) $\pi'[0] \equiv \mu\varepsilon.\varepsilon[1] \star \varpi$;

(6) $\pi'[0] \equiv \mu\varepsilon.\varepsilon[n] \star \varpi$, with $n \geq 2$.

According to the above cases, the outer reduction of $C[W[M]]$ proceeds as:

(1) $\dots \twoheadrightarrow_o \mu\delta.\pi'[0] \star \pi'' \not\twoheadrightarrow_o$.

(2) $\dots \twoheadrightarrow_o \mu\delta.\text{nil}[n] \star \varpi\{\pi''/\varepsilon\} \not\twoheadrightarrow_o$.

(3) $\dots \twoheadrightarrow_o \mu\delta.\varepsilon'[n] \star \varpi\{\pi''/\varepsilon\} \not\twoheadrightarrow_o$.

(4) $\dots \twoheadrightarrow_o \mu\delta.(\mu\varepsilon.\varepsilon[0] \star \varpi) \star \pi'' \twoheadrightarrow_o \mu\delta.\pi'[0] \star \pi'$.

(5) $\dots \twoheadrightarrow_o \mu\delta.(\mu\varepsilon.\varepsilon[1] \star \varpi) \star \pi'' \twoheadrightarrow_o \mu\delta.\pi'[0] \star \pi'$.

(6) $\dots \twoheadrightarrow_o \mu\delta.(\mu\varepsilon.\varepsilon[n] \star \varpi) \star \pi'' \twoheadrightarrow_o \mu\delta.\pi'[n-2] \star \varpi\{\pi''/\varepsilon\}$.

Suppose $C[W[M]]$ has a proper onf, say, Z. In each of the above cases there is no step in the outer reduction path $C[W[M]] \twoheadrightarrow_o Z$ such that a substitution instance of the occurrence of M put into the hole is active part of a contracted redex. Therefore an isomorphic outer-reduction path takes any other term $C[W[N]]$ to its onf, which must be proper too. $\qquad\square$

Failure of Böhm's Theorem. Theorem 2.2 implies that Böhm's theorem fails (and quite violently) in the stack calculus. Every pair M, N of distinct $\to_{s\eta}$-normal forms yields a pair $W[M], W[N]$ of distinct $\to_{s\eta}$-normal forms which are operationally equivalent.

A *stack-theory* (resp. *extensional stack-theory*) for the stack calculus is any set $\mathscr{T}$ of equalities between stack-expressions containing $=_s$ (resp. $=_{s\eta}$) and closed under context formation and replacement of $\mathscr{T}$-equal sub-expressions. We indicate by $=_{\mathscr{T}}$ the congruence associated to the theory $\mathscr{T}$.

A stack-theory $\mathscr{T}$ is *inconsistent* if for every pair of terms M, N and every variable $\gamma \notin \text{FV}(M) \cup \text{FV}(N)$ we have that $\mu\gamma.M \star \gamma =_{\mathscr{T}} \mu\gamma.N \star \gamma$; $\mathscr{T}$ is *consistent* otherwise. Since there are distinct $\to_{s\eta}$-normal forms, Theorem 3.1 implies that $=_s$ and $=_{s\eta}$ are consistent equational theories, of which $=_{s\eta}$ is extensional. Also the relation $\approx$ is an extensional equational theory which is consistent, because for example $\mathbf{T} \not\approx \Omega$.

A theory is *Hilbert–Post complete* (HP-complete, for short) if it is maximally consistent (cannot be properly extended to a consistent theory). The next theorem shows that operational equivalence is maximally consistent.

Theorem 2.3. *The relation $\approx$ is an HP-complete equational theory for the stack calculus.*

Proof. The relation $\approx$ is indeed an equational theory for the stack calculus, because it is closed w.r.t. context formation. Suppose, by contradiction, that $\mathscr{T}$ is a consistent theory that contains $\approx$ properly and let M, N be terms such that $M \not\approx N$ and $M =_{\mathscr{T}} N$. Then there exists a head context $C[\cdot]$ such that, say, $C[M]$ has a proper onf and $C[N]$ does not have a proper onf. Assume $C[M] =_s \mu\alpha.\beta[n] \star \vec{N} \cdot \gamma[k]$ and define the context $C'[\cdot] := \mu\varepsilon.(\mu\beta.[\cdot] \star \alpha) \star \underbrace{\mathbf{I} \cdots \mathbf{I}}_{n} \cdot (\mu\delta.\mathbf{I} \star \varepsilon) \cdot \varepsilon$. Then $C'[C[M]] =_s \mathbf{I}$. On the other hand $C'[C[N]]$ cannot have a proper onf. Let $u := \mu x.f[0] \star (\mu\beta.x[0] \star x[0] \cdot \beta) \cdot x[1]$, $U := \mu\gamma.u \star u \cdot \gamma$ and $Y := \mu f.U \star f[1]$. Finally set $\mathbf{T}_\infty := \mu\delta.Y \star \mathbf{T}\cdot\delta$.

Since $\mathbf{T}_\infty$ does not have an onf we have $C'[C[N]] \approx \mathbf{T}_\infty$ and from $\mathbf{I} =_s C'[C[M]] =_{\mathscr{T}} C'[C[N]]$ and the fact that the congruence $=_{\mathscr{T}}$ extends both $=_s$ and $\approx$, we get that $\mathbf{I} =_{\mathscr{T}} \mathbf{T}_\infty$. Now take an arbitrary term Z and a variable $\gamma \notin \text{FV}(Z)$. We have $\mu\gamma.Z \star \gamma =_s \mu\gamma.\mathbf{I} \star Z \cdot \gamma =_{\mathscr{T}} \mu\gamma.\mathbf{T}_\infty \star Z \cdot \gamma =_s \mathbf{T}_\infty$. Since the congruence $=_{\mathscr{T}}$ extends both $=_s$ and $\approx$, we can conclude that $\mu\gamma.Z \star \gamma =_{\mathscr{T}} \mu\gamma.Z' \star \gamma$ for all terms Z, Z' and every variable $\gamma \notin \text{FV}(Z) \cup \text{FV}(Z')$. Therefore $\mathscr{T}$ is inconsistent. $\qquad\square$

3 The extended stack calculus

The *extended stack calculus* is a super-language of the stack calculus. Formally, it is obtained by incorporating the syntactic category of processes into that of terms. Therefore the grammar for the extended language is the following one:

$$
\begin{array}{lll}
\text{stacks} & \pi, \varpi & ::= \quad \mathsf{nil} \mid \alpha \mid \mathsf{cdr}(\pi) \mid M.\pi \\
\text{terms} & M, N & ::= \quad \mathsf{car}(\pi) \mid \mu\alpha.M \mid M \star \pi
\end{array}
$$

We still use letters E, E' to range over expressions which are either stacks or terms. An example of term which belongs to the extended language but not to the original one is $\mu\beta.\mu\alpha.\mathsf{car}(\alpha) \star \mathsf{cdr}(\alpha)$. Application associates to the left, so that $M \star \pi \star \varpi$ stands for $(M \star \pi) \star \varpi$ and application has precedence over μ-abstraction. If $\vec{\pi} = \pi_1, \ldots, \pi_m$ and $\vec{\alpha} = \alpha_1, \ldots, \alpha_n$, we shall abbreviate the term $\mu\alpha_1 \ldots \mu\alpha_n.(M \star \pi_1 \star \cdots \star \pi_m)$ as $\mu\vec{\alpha}.M \star \vec{\pi}$. We denote by Σ^{t}, Σ^{s} and Σ^{e} the sets of all terms, stacks and expressions respectively. We still use $\to_{\mathsf{s}}$, $\to_{\mathsf{s}\eta}$, $=_{\mathsf{s}}$ and $=_{\mathsf{s}\eta}$ to indicate the straightforward extensions of the corresponding relations defined for the stack calculus. Also the notion of canonical form extends straightforwardly to the extended stack calculus.

Theorem 3.1. *The reductions $\to_{\mathsf{s}}$ and $\to_{\mathsf{s}\eta}$ in the extended stack calculus are both Church-Rosser.*

The proof of Theorem 3.1 is an easy modification of that of Theorem 2.1 (see [7]).

Also the definitions of (consistent) equational theories and HP-completeness extend straightforwardly to the larger calculus and once again Theorem 3.1 guarantees the consistency of the theories $=_{\mathsf{s}}$ and $=_{\mathsf{s}\eta}$.

In this paper we are not concerned with semantics. We just mention that the works of Streicher and Reus [27] and Nakazawa and Katsumata [18] already provide sound models for the extended stack calculus, which are the same as those for the $\Lambda\mu$-calculus.

3.1 Head-reduction and operational equivalence for the extended calculus

We now provide a notion of *head-reduction* for the stack calculus, which is performed by contracting the left-most redex only. The one-step *head-reduction* on terms is given by the following rule:

$$
M \to_h \mu\vec{\alpha}.N\{\varpi/\beta\} \star \vec{\pi} \qquad \text{if } \mu\vec{\alpha}.(\mu\beta.N) \star \varpi \star \vec{\pi} \text{ is the canonical form of } M
$$

Note that head-reduction is *deterministic* and we don't take any contextual closure for $\to_h$; its reflexive and transitive closure is denoted by $\twoheadrightarrow_h$. A term *M is in head-normal form* (*hnf*, for short) if it is not $\to_h$-reducible. It is straightforward to see that a term is in hnf iff it has the form $\mu\vec{\alpha}.H \star \vec{\pi}$, where the stacks in the sequence $\vec{\pi}$ are arbitrary, H is either $\beta[n]$ (in which case we say that the hnf is *proper*) or $\mathsf{nil}[n]$ (in which case we say that the hnf is *improper*); H is the head of the hnf in question. A term *has a* hnf if it $\twoheadrightarrow_h$-reduces to a term in hnf. Of course the head-reduction strategy is complete for finding hnf's of terms, i.e., if $M \twoheadrightarrow_{\mathsf{s}}$-reduces to a term N' in hnf, then M reduces to some hnf N'' via head-reduction.

For convenience we define a partial function $\mathsf{Hnf}(\cdot) : \Sigma^{\mathsf{t}} \rightharpoonup \Sigma^{\mathsf{t}}$ which returns the $\to_h$-normal form of a term, if it exists.

The *head contexts* of the extended stack calculus are produced by the following grammar:

$$
C[\cdot] ::= [\cdot] \mid C[\cdot] \star \pi \mid \mu\alpha.C[\cdot]
$$

The next definition is the analogue of Definition 2.1 for the extended stack calculus.

Definition 3.1 (Operational equivalence (extended)). *Two terms M, N are* operationally equivalent, *notation $M \approx N$, if for every head context $C[\cdot]$ we have that $C[M]$ has a proper hnf iff $C[N]$ has a proper hnf.*

We use the same symbol as in Definition 2.1 because there will be no ambiguity: from now on we are only concerned with the extended stack calculus. The relation $\approx$ is a consistent extensional theory which is HP-complete, because the proof of Theorem 2.3 works also for the larger calculus.

Theorem 3.2. *The relation $\approx$ is an HP-complete equational theory for the extended stack calculus.*

3.2 Similarity and separability

The following definition introduces an important concept, somewhat orthogonal to operational equivalence, which is very typical of functional calculi (as the λ-calculus).

Definition 3.2 (Separability). *We say that M and N are* separable *if there exists a head context $C[\cdot]$ such that $C[M] =_s \mathbf{T}$ and $C[N] =_s \mathbf{F}$.*

The following theorem says that separability and operational equivalence are somewhat orthogonal to each other.

Theorem 3.3. *If M and N are separable, then $M \not\approx N$.*

Proof. Suppose M, N separable. Then there exists a context $C[\cdot]$ such that $C[M] =_s \mathbf{T}$ and $C[N] =_s \mathbf{F}$ and setting $C'[\cdot] := \mu \varepsilon . C[\cdot] \star \Omega . \mathbf{I} . \varepsilon$ we obtain that $C'[M] =_s \Omega$ and $C'[N] =_s \mathbf{I}$, thus showing that $M \not\approx N$. $\square$

The converse of Theorem 3.3 does not hold. For example $\Omega \not\approx \mathbf{I}$ but it is also true that Ω and $\mathbf{I}$ are not separable. In fact if a term M does not have a hnf, then $C[M]$ does not have a hnf too for every head context $C[\cdot]$. This means that for no head context $C[\cdot]$ we can have $C[\Omega] =_s \mathbf{T}$ (or $C[\Omega] =_s \mathbf{F}$).

Next we introduce the notion of *similarity* between stacks and terms which, for terms having an hnf, is weaker than inseparability.

Definition 3.3 (Similarity for stacks). *We define the* similarity relation $\overset{s}{\sim}$ *on Σ^s as the smallest equivalence relation closed under $=_s$ satisfying the following conditions:*

(1) $\pi \overset{s}{\sim} M_1 \bullet \ldots \bullet M_m \bullet \mathsf{nil}[k]$

(2) if $k - m = k' - m'$, then $M_1 \bullet \ldots \bullet M_m \bullet \gamma[k] \overset{s}{\sim} N_1 \bullet \ldots \bullet N_{m'} \bullet \gamma[k']$

Definition 3.4 (Similarity for terms). *We define the* similarity relation $\sim$ *on Σ^t as the smallest equivalence relation closed under $=_s$ satisfying the following conditions:*

(1) if $k - m = k' - m'$, $\pi_i \overset{s}{\sim} \varpi_i$ for all $i = 1, \ldots, min\{m, m'\}$, and $\varpi_{min\{m,m'\}+j} \overset{s}{\sim} \alpha_{min\{k,k'\}+j}$ for all $j = 1, \ldots, (max\{k, k'\} - min\{k, k'\})$ then

$$\mu \alpha_1 \ldots \alpha_k . \beta[n] \star \pi_1 \star \cdots \star \pi_m \sim \mu \alpha_1 \ldots \alpha_{k'} . \beta[n] \star \varpi_1 \star \cdots \star \varpi_{m'}$$

(2) if $\mathsf{Hnf}(M)$ and $\mathsf{Hnf}(N)$ are both defined and improper, then $M \sim N$

(3) if $\mathsf{Hnf}(M)$ and $\mathsf{Hnf}(N)$ are both undefined, then $M \sim N$

The following theorems show that dissimilarity of terms having a hnf implies their separability. We start by treating a particular case, which is nevertheless non-trivial.

It will occur frequently to use stacks of the form $\overbrace{M \bullet \ldots \bullet M}^{n} . \pi$. Therefore we set the special notation $M^n . \pi$ for these stacks.

Theorem 3.4. *Let $M \equiv \mu\alpha.\beta[n] \star \pi$ and $N \equiv \mu\alpha.\beta'[n'] \star \pi'$. If $M \not\sim N$, then M and N are separable.*

Proof. Assume $M \not\sim N$. We analyze the different reasons for this fact and each time we build a context $C[\cdot]$ such that $C[M] =_s \mathbf{T}$ and $C[N] =_s \mathbf{F}$. In the rest of the proof we let ε be a fresh variable. Since M and N are not similar, we have the following possible cases:

(1) $\beta \not\equiv \beta'$;

(2) $\beta \equiv \beta'$ but $n \neq n'$;

(3) $\beta \equiv \beta'$, $n = n'$ but $\pi \not\overset{s}{\sim} \pi'$.

(1) Define $\pi := \mathbf{I}^n \cdot (\mu\delta.\mathbf{T} \star \varepsilon).\varepsilon$, where $\delta \not\equiv \varepsilon$, $\pi' := \mathbf{I}^{n'} \cdot (\mu\delta.\mathbf{F} \star \varepsilon).\varepsilon$, where $\delta \not\equiv \varepsilon$, and $C[\cdot] := \mu\varepsilon.(\mu\beta'.(\mu\beta.[\cdot] \star \alpha) \star \pi) \star \pi'$. Then $C[M] =_s \mathbf{T}$ and $C[N] =_s \mathbf{F}$.

(2) We can assume w.l.o.g. that $n > n'$ since the opposite case can be treated symmetrically. Define $\pi := \mathbf{I}^{n'} \cdot (\mu\delta.\mathbf{F} \star \varepsilon).\mathbf{I}^{n-n'-1} \cdot (\mu\delta.\mathbf{T} \star \varepsilon).\varepsilon$, where $\delta \not\equiv \varepsilon$, and $C[\cdot] := \mu\varepsilon.(\mu\beta.[\cdot] \star \alpha) \star \pi$. Then $C[M] =_s \mathbf{T}$ and $C[N] =_s \mathbf{F}$.

In order to treat case (3), we need to explicit the possible forms of π and π'. By assumption there some are terms $M_1, \ldots, M_m, N_1, \ldots, N_{m'}$ and numbers $k, k' \in \mathbb{N}$ such that:

- $M \equiv \mu\alpha.\beta[n] \star M_1 \cdot \ldots \cdot M_m \cdot \gamma[k]$
- $N \equiv \mu\alpha.\beta'[n'] \star N_1 \cdot \ldots \cdot N_{m'} \cdot \gamma'[k']$

Case (3) can be exhaustively splitted in the following sub-cases:

(3.1) $\beta \equiv \beta'$, $n = n'$ but $\gamma \not\equiv \gamma'$ (no assumptions on m, k, m', k') and

    (3.1.1) $\gamma \not\equiv \beta$;

    (3.1.2) $\gamma \equiv \beta$.

(3.2) $\beta \equiv \beta'$, $n = n'$, $\gamma \equiv \gamma'$ but $m - k \neq m' - k'$ and

    (3.2.1) $\gamma \not\equiv \beta$;

    (3.2.2) $\gamma \equiv \beta$.

We now show how to reduce the cases (3.1.1), (3.1.2) and (3.2.1) to the case (3.2.2), for which we show how to build the separating context.

(3.1.1) If $\gamma' \equiv \beta$, then case (3.1.2) applies (changing the roles of γ and γ'). If $\gamma' \not\equiv \beta$, then define $C[\cdot] := \mu\alpha.(\mu\gamma.[\cdot] \star \alpha) \star \beta$. Now case (3.1.2) applies to $C[M]$ and $C[N]$.

(3.1.2) Define $C[\cdot] := \mu\alpha.(\mu\gamma'.[\cdot] \star \alpha) \star \gamma'[m' + m + k + 1]$. Now case (3.2.2) applies to $C[M]$ and $C[N]$.

(3.2.1) Define $C[\cdot] := \mu\alpha.(\mu\gamma.[\cdot] \star \alpha) \star \beta$. Now case (3.2.2) applies to $C[M]$ and $C[N]$.

(3.2.2) Let $\partial m = |m - m'|$, $\partial k = |k - k'|$, $d = \partial k + \partial m$, and $e = min\{\partial k, \partial m\}$. Note that $m - k = m' - k'$ if, and only if, $m - m' = k - k'$. Therefore, under the hypothesis of this case, $m - m' \neq k - k'$ and $d > 0$.

$$X := \begin{cases} \mu\delta.\delta[0] \star \varepsilon[2] & \text{if } k < k' \\ \mu\delta.\delta[e] \star \varepsilon[2] & \text{if } k \geq k' \end{cases}$$

Moreover define $\pi := \mathbf{I}^n.\,(\mu\delta.X \star \delta[n+1+max\{m,m'\}]).\,\varepsilon[1]^{\partial m+max\{k,k'\}}.\,\varepsilon[0].\,\varepsilon$ and $C[\cdot] := \mu\varepsilon.(\mu\beta.[\cdot] \star \alpha) \star \pi$. We can assume w.l.o.g. that $m \geq m'$ (the other case can be treated symmetrically) so that we have

$$C[M] =_{\mathsf{s}} \begin{cases} \mu\varepsilon.X \star \varepsilon[1]^d.\,\varepsilon[0].\,\varepsilon & \text{if } k < k' \\ \mu\varepsilon.X \star \varepsilon[1]^{\partial m}.\,\varepsilon[0].\,\varepsilon & \text{if } k \geq k' \end{cases} \quad \text{and} \quad C[N] =_{\mathsf{s}} \begin{cases} \mu\varepsilon.X \star \varepsilon[0].\,\varepsilon & \text{if } k < k' \\ \mu\varepsilon.X \star \varepsilon[1]^{\partial k}.\,\varepsilon[0].\,\varepsilon & \text{if } k \geq k' \end{cases}$$

Concluding the computation we obtain

$$C[M] =_{\mathsf{s}} \begin{cases} \mathbf{F} & \text{if } k < k' \\ \mathbf{T} & \text{if } k \geq k' \end{cases} \quad \text{and} \quad C[N] =_{\mathsf{s}} \begin{cases} \mathbf{T} & \text{if } k < k' \\ \mathbf{F} & \text{if } k \geq k' \end{cases}$$

$\square$

Now we can now prove the following more general statement.

Theorem 3.5. *Let M, N be terms having a proper hnf. Then $M \not\sim N$ implies that M and N are separable.*

Proof. Suppose M and N be terms with a proper hnf and assume $M \not\sim N$. We analyze the different cases in which $M \not\sim N$ and each time we build a context $C[\cdot]$ such that $C[M] =_{\mathsf{s}} \mathbf{T}$ and $C[N] =_{\mathsf{s}} \mathbf{F}$. Since M, N both have a hnf, let's say we have

- $\mathrm{Hnf}(M) = \mu\alpha_1 \ldots \alpha_k.\beta[n] \star \pi_1 \star \cdots \star \pi_m$
- $\mathrm{Hnf}(N) = \mu\alpha_1 \ldots \alpha_{k'}.\beta'[n'] \star \varpi_1 \star \cdots \star \varpi_{m'}$

In the rest of the proof we let ε be a fresh variable and we suppose, w.l.o.g. that $k' \geq k$. Since M and N are not similar, we have the following possible cases:

(1) $\beta \not\equiv \beta'$;

(2) $\beta \equiv \beta'$ but $n \neq n'$;

(3) $\beta \equiv \beta'$, $n = n'$ but $m - k \neq m' - k'$;

(4) $\beta \equiv \beta'$, $n = n'$, $m - k = m' - k'$ but there is some $i = 1, \ldots, m$ such that $\pi_i \not\overset{\mathsf{s}}{\sim} \varpi_i$ (note that $m' \geq m$);

(5) $\beta \equiv \beta'$, $n = n'$, $m - k = m' - k'$, $\pi_i \overset{\mathsf{s}}{\sim} \varpi_i$ for all $i = 1, \ldots, m$ but there is some $j = 1, \ldots, k' - k$ such that $\varpi_{m+j} \not\overset{\mathsf{s}}{\sim} \alpha_{k+j}$ (note that $m' \geq m$).

We now show how to build in each case a separating context.

(1) Define $\pi := \mathbf{I}^n.\,(\mu\delta_1 \ldots \delta_m \alpha_{k+1} \ldots \alpha_{k'}.\mathbf{T}).\,\varepsilon$, $\pi' := \mathbf{I}^{n'}.\,(\mu\delta_1 \ldots \delta_{m'}.\mathbf{F}).\,\varepsilon$, and $C[\cdot] := (\mu\beta\beta'.[\cdot] \star \alpha_1 \star \cdots \star \alpha_{k'}) \star \pi \star \pi'$. Then $C[M] =_{\mathsf{s}} \mathbf{T}$ and $C[N] =_{\mathsf{s}} \mathbf{F}$.

(2) Suppose that $n' > n$ (all other cases can be treated similarly). Define $\pi := \mathbf{I}^n.\,(\mu\delta_1 \ldots \delta_m \alpha_{k+1} \ldots \alpha_{k'}.\mathbf{T}).\,\mathbf{I}^{n'-n-1}.\,(\mu\delta_1 \ldots \delta_{m'}.\mathbf{F}).\,\varepsilon$ and $C[\cdot] := (\mu\beta.[\cdot] \star \alpha_1 \star \cdots \star \alpha_{k'}) \star \pi$. Then $C[M] =_{\mathsf{s}} \mathbf{T}$ and $C[N] =_{\mathsf{s}} \mathbf{F}$.

(3) Note that, having assumed $m - k \neq m' - k'$, we have $m + k' - k \neq m'$. Suppose $m + k' - k > m'$ (the opposite case can be treated similarly). Now let $p = m + k' - k$, $h = p - m'$ and define $\pi := \mathbf{I}^n.\,(\mu\alpha_1 \ldots \alpha_{p+1}.\alpha_{p+1}[0]).\,\varepsilon$, $C[\cdot] := (\mu\beta.[\cdot] \star \alpha_1 \star \cdots \star \alpha_{k'} \star \delta \star \varepsilon_1 \star \cdots \star \varepsilon_h) \star \pi$, and $C'[\cdot] := (\mu\delta\varepsilon_h.[\cdot]) \star ((\mu\alpha_1 \ldots \alpha_h.\mathbf{T}).\,\varepsilon) \star (\mathbf{F}.\,\varepsilon)$. Then $C'[C[M]] =_{\mathsf{s}} \mathbf{T}$ and $C'[C[N]] =_{\mathsf{s}} \mathbf{F}$.

(4) Define $X := \mu\alpha_1 \ldots \alpha_{m'}\beta''.\beta''[0] \star \alpha_i$, $\pi' := \mathbf{I}^n.X.\varepsilon$, and $C[\cdot] := (\mu\beta.[\cdot] \star \alpha_1 \star \cdots \star \alpha_{k'}) \star \pi'$. Then we have

$$C[M] =_{\mathsf{s}} \mu\beta''.\beta''[0] \star \pi_i\{\pi'/\beta\} \text{ and } C[N] =_{\mathsf{s}} \mu\beta''.\beta''[0] \star \varpi_i\{\pi'/\beta\}$$

Finally since $\pi_i \overset{\mathsf{s}}{\not\sim} \varpi_i$ and $\overset{\mathsf{s}}{\not\sim}$ is closed under substitution, we also have $\pi_i\{\pi'/\beta\} \overset{\mathsf{s}}{\not\sim} \varpi_i\{\pi'/\beta\}$, which in turn implies $C[M] \not\sim C[N]$. At this point we can apply Theorem 3.4 to show that $C[M]$ and $C[N]$ are separable, and hence M and N are separable.

(5) Define $X := \mu\alpha_1 \ldots \alpha_{m'}\beta''.\beta''[0] \star \alpha_{m+j}$, $\pi' := \mathbf{I}^n.X.\varepsilon$, and $C[\cdot] := (\mu\beta.[\cdot] \star \alpha_1 \star \cdots \star \alpha_{k'}) \star \pi'$. Then we have

$$C[M] =_{\mathsf{s}} \mu\beta''.\beta''[0] \star \alpha_{k+j}\{\pi'/\beta\} \text{ and } C[N] =_{\mathsf{s}} \mu\beta''.\beta''[0] \star \varpi_{m+j}\{\pi'/\beta\}$$

Finally since $\varpi_{m+j} \overset{\mathsf{s}}{\not\sim} \alpha_{k+j}$ and $\overset{\mathsf{s}}{\not\sim}$ is closed under substitution, we also have $\varpi_{m+j}\{\pi'/\beta\} \overset{\mathsf{s}}{\not\sim} \alpha_{k+j}\{\pi'/\beta\}$, which in turn implies $C[M] \not\sim C[N]$. At this point we can apply Theorem 3.4 to show that $C[M]$ and $C[N]$ are separable, and hence M and N are separable.   □

The converse of Theorem 3.5 does not hold; for example $\mu\gamma.\gamma[0] \star \mathbf{T}.\gamma$ and $\mu\gamma.\gamma[0] \star \mathbf{F}.\gamma$ are separable but it is also true that $\mu\gamma.\gamma[0] \star \mathbf{T}.\gamma \sim \mu\gamma.\gamma[0] \star \mathbf{F}.\gamma$. The point is that the relation $\sim$ only looks at the "surface" of terms, while separation may require to unravel terms by iteratively computing head normal forms, i.e., looking at their Böhm trees.

3.3 Böhm out technique and Böhm's theorem

As for the λ-calculus, Böhm trees can be defined for stack terms too. While the nodes of Böhm trees of λ-terms are indexed by sequences of natural numbers, the nodes of Böhm trees of stack terms should be indexed by sequences of pairs of natural numbers. The reason for this choice is that a child of the node corresponding to a hnf $\mu\vec{\alpha}.\beta[n] \star \vec{\pi}$ must be selected with two coordinates (j, j'), saying that the child is the root of the Böhm tree of the j'-th term of the canonical form of the stack π_j.

The Böhm trees for the λ-calculus (see [2, § 10]) are partial functions mapping sequences of natural numbers either to a special constant $\bot$ or to a λ-term o shape $\lambda\vec{x}.y$. We will still speak of Böhm trees for the extended stack calculus, but strictly speaking we mean partial functions mapping pairs of natural numbers to non-necessarily normal terms, called *nodes*. Letters $\sigma, \tau, \rho, \ldots$ range over the set Seq of all finite sequences of pairs of strictly positive natural numbers. We define the order $<$ on these sequences as follows: $\tau < \sigma$ iff τ is a proper prefix of σ. We denote by $\sharp\sigma$ the length of the sequence σ. Moreover if $\vec{N}$ is a sequence of terms, then $\sharp\vec{N}$ indicates the lenght of $\vec{N}$.

Definition 3.5. *Given a term M we define a partial map $M(\cdot) : \mathsf{Seq} \rightharpoonup \Sigma^{\mathsf{t}}$ as follows:*

$$M(\sigma) := M \qquad \text{if } \sigma \text{ is the empty sequence}$$

$$M(\tau \cdot (j, j')) := \begin{cases} N_{j'} & \text{if } M(\tau) \text{ is defined, } \mathsf{Hnf}(M(\tau)) = \mu\vec{\alpha}.\beta[n] \star \pi_1 \star \cdots \star \pi_m \text{ and} \\ & j \leq m \text{ and } \pi_j \text{ has canonical form } \vec{N}.\gamma[k] \text{ or } \vec{N}.\mathsf{nil}[k], \text{ with } j' \leq \sharp\vec{N} \\ \\ \text{undefined} & \text{otherwise} \end{cases}$$

The map $M(\cdot) : \mathsf{Seq} \rightharpoonup \Sigma^{\mathsf{t}}$ for us *is the Böhm tree of M*. We let $\mathsf{dom}(M) = \{\sigma \in \mathsf{Seq} : M(\sigma) \text{ is defined }\}$.

In the present section we prove a theorem which is the analogue of Böhm's Theorem for λ-calculus. Such theorem is supported fundamentally by the forthcoming Lemma 3.7 (analogous to what is called

the Böhm out Lemma for the λ-calculus [2, § 10]) which shows how to extract substitution instances of nodes of the Böhm tree of a term, in such a way that some important properties are preserved.

For technical reasons we need to introduce the set $\mathrm{dom}(M,n) = \{\sigma \in \mathrm{dom}(M) : \sharp\sigma \leq n\}$. The following definitions 3.6, 3.7 and 3.8 are all auxiliary for the statement and proof of the subsequent Lemma 3.6.

Definition 3.6. *We define the* breadth $\mathrm{b}(M)$ *and the* weight $\mathrm{w}(M)$ *of* $M \in \Sigma^{\mathrm{t}}$ *as*

$$
\mathrm{b}(M) = \begin{cases} 0 & \text{if } \mathrm{Hnf}(M) \text{ is undefined} \\ 0 & \text{if } \mathrm{Hnf}(M) \text{ is defined but improper} \\ m & \text{if } \mathrm{Hnf}(M) = \mu\vec{\alpha}.\beta[n] \star \pi_1 \star \cdots \star \pi_m \end{cases}
\qquad
\mathrm{w}(M) = \begin{cases} 0 & \text{if } \mathrm{Hnf}(M) \text{ is undefined} \\ 0 & \text{if } \mathrm{Hnf}(M) \text{ is defined but improper} \\ n & \text{if } \mathrm{Hnf}(M) = \mu\vec{\alpha}.\beta[n] \star \pi_1 \star \cdots \star \pi_m \end{cases}
$$

Definition 3.7. *The* n-bounded breadth $\mathrm{b}(M,n)$ *and the* n-bounded weight $\mathrm{w}(M,n)$ *of a term* M *are defined as* $\mathrm{b}(M,n) = \max\{\mathrm{b}(M(\sigma)) : \sigma \in \mathrm{dom}(M,n)\}$ *and* $\mathrm{w}(M,n) = \max\{\mathrm{w}(M(\sigma)) : \sigma \in \mathrm{dom}(M,n)\}$, *respectively.*

Definition 3.8. *Let* q,p *be natural numbers. We define the expressions*

$$
\mathbf{A}^{(q)} := \mu\varepsilon_1 \ldots \varepsilon_q \delta.\delta[0] \star \varepsilon_1 \star \cdots \star \varepsilon_q
\qquad
\pi^{(\varepsilon,q,p)} := \underbrace{\mathbf{A}^{(q)} \ldots \mathbf{A}^{(q)}}_{p \text{ times}} \cdot \varepsilon
$$

Clearly by the common conventions about bound variables, in the above definition $\varepsilon_1, \ldots, \varepsilon_q, \delta$ are all distinct from each other and from ε.

The next lemma is the combinatorial core of the forthcoming Lemma 3.7, and is the analogue for the extended stack calculus of what is called the *Böhm-out technique* for the λ-calculus (see [2, § 10]). For a sequence $\sigma = (j_1, j_1') \cdots (j_l, j_l')$ and a natural number $i \leq l$ we indicate with $\sigma \restriction i$ the sequence $(j_1, j_1') \cdots (j_i, j_i')$ (so for example $\sigma \restriction 0$ is the empty sequence).

Lemma 3.6 (Böhm out)**.** *Let* M *be a term, let* n *a natural number and let* $q \geq \mathrm{b}(M,n)$, $p \geq \mathrm{w}(M,n)$. *Then for every sequence* $\sigma \in \mathrm{dom}(M,n)$ *there exists a context* $C[\cdot]$ *such that* $C[M] \twoheadrightarrow_{\mathrm{s}} M(\sigma)\{\pi^{(\varepsilon_1,q,p)}/\beta_1\} \cdots \{\pi^{(\varepsilon_l,q,p)}/\beta_l\}$ *where*

(i) $l = \sharp\sigma$ *and for each* $i = 1, \ldots, l$, β_i *is the head variable of*
$M(\sigma \restriction i-1)\{\pi^{(\varepsilon_1,q,p)}/\beta_1\} \cdots \{\pi^{(\varepsilon_{i-1},q,p)}/\beta_{i-1}\}$

(ii) $\varepsilon_1, \ldots, \varepsilon_l$ *is a sequence of pairwise distinct variables disjoint from* $\beta_1, \ldots, \beta_l$ *and not occurring in* M

(iii) $C[M]$ *has a proper hnf iff* $M(\sigma)$ *has a proper hnf.*

Proof. We proceed at the same time to define the context $C[\cdot]$ and prove its properties by induction on the length of σ.

If σ is the empty sequence, then $C[\cdot] := [\cdot]$ and the statement is trivially satisfied.

Let $\sigma = (j_1, j_1') \cdots (j_l, j_l') \in \mathrm{dom}(M,n)$ and $\tau = \sigma \restriction l-1$. Now $\tau \in \mathrm{dom}(M,n)$, since $\tau < \sigma$, so by induction hypothesis applied to τ we have a context $C[\cdot]$ such that $C[M] \twoheadrightarrow_{\mathrm{s}} M(\tau)\{\pi^{(\varepsilon_1,q,p)}/\beta_1\} \cdots \{\pi^{(\varepsilon_{l-1},q,p)}/\beta_{l-1}\}$ where

(i) for each $i = 1, \ldots, l-1$, β_i is the head variable of $M(\tau \restriction i-1)\{\pi^{(\varepsilon_1,q,p)}/\beta_1\} \cdots \{\pi^{(\varepsilon_{i-1},q,p)}/\beta_{i-1}\}$

(ii) $\vec{\varepsilon} = \varepsilon_1, \ldots, \varepsilon_{l-1}$ is a sequence of pairwise distinct variables disjoint from $\vec{\beta} = \beta_1, \ldots, \beta_{l-1}$ and not occurring in M.

For simplicity, we write E^* for $E\{\pi^{(\varepsilon_1,q,p)}/\beta_1\}\cdots\{\pi^{(\varepsilon_{l-1},q,p)}/\beta_{l-1}\}$, so that $C[M] \twoheadrightarrow_{\mathsf{s}} (M(\tau))^*$. The definition of the new context for the longer sequence σ is based on the shape of the hnf of $C[M]$ (and therefore on its existence). Since $\tau \in \mathrm{dom}(M,n)$ and it is not maximal, we have that $M(\tau)$ does have a proper hnf, say, $\mathsf{Hnf}(M(\tau)) \equiv \mu\vec{\alpha}.\beta[h] \star \pi_1\ldotp\ldots\ldotp\pi_m$. Then we have

$$\mathsf{Hnf}(M(\tau))^* \quad \equiv \quad \begin{cases} \mu\vec{\alpha}.\mathbf{A}^{(q)} \star \pi_1^* \star \cdots \star \pi_m^* & \text{if } \beta \in \vec{\beta} \\ \mu\vec{\alpha}.\beta[h] \star \pi_1^* \star \cdots \star \pi_m^* & \text{if } \beta \notin \vec{\beta} \end{cases}$$

$$\twoheadrightarrow_{\mathsf{s}} \quad \begin{cases} \mu\vec{\alpha}\varepsilon_{m+1}\ldots\varepsilon_q\delta.\delta[0] \star \pi_1^* \star \cdots \star \pi_m^* \star \varepsilon_{m+1} \star \cdots \star \varepsilon_q & \text{if } \beta \in \vec{\beta} \\ \mu\vec{\alpha}.\beta[h] \star \pi_1^* \star \cdots \star \pi_m^* & \text{if } \beta \notin \vec{\beta} \end{cases}$$

$$\equiv \quad \mathsf{Hnf}((M(\tau))^*)$$

because by hypothesis $p \geq h$ and $q \geq m$. The computations above give, according to the different cases, the head variable of $C[M]$, since $C[M] \twoheadrightarrow_{\mathsf{s}} (M(\tau))^*$. Now let $\varepsilon \notin \mathrm{FV}(\mathsf{Hnf}((M(\tau))^*))$ and set

- $C'[\cdot] := \begin{cases} [\cdot] \star \vec{\alpha} & \text{if } \beta \in \vec{\beta} \\ (\mu\beta.[\cdot] \star \vec{\alpha}) \star \pi^{(\varepsilon,q,p)} & \text{if } \beta \notin \vec{\beta} \text{ and } \beta \notin \vec{\alpha} \\ [\cdot] \star \alpha_1 \star \cdots \star \alpha_{r-1} \star \pi^{(\varepsilon,q,p)} \star \alpha_{r+1} \star \cdots \star \alpha_t & \text{if } \beta \notin \vec{\beta} \text{ and } \beta \equiv \alpha_r \in \vec{\alpha} = \alpha_1,\ldots,\alpha_t \end{cases}$

- $C''[\cdot] := [\cdot] \star \varepsilon_{m+1} \star \cdots \star \varepsilon_q \star ((\mu\alpha_1\ldots\alpha_q.\alpha_{j_l}[j'_l - 1])\ldotp\varepsilon)$

We claim that the context $D[\cdot] := C''[C'[C[\cdot]]]$ satisfies the statement of the lemma. By all the definitions and results above we have that

$$C'[\mathsf{Hnf}((M(\tau))^*)] \twoheadrightarrow_{\mathsf{s}} \mu\varepsilon_{m+1}\ldots\varepsilon_q\delta.\delta[0] \star \pi_1^*\{\pi^{(\varepsilon,q,p)}/\beta\} \star \cdots \star \pi_m^*\{\pi^{(\varepsilon,q,p)}/\beta\} \star \varepsilon_{m+1} \star \cdots \star \varepsilon_q$$

because if $\beta \in \vec{\beta}$, then the sequences of stacks $\pi_1^*,\ldots,\pi_m^*$ and $\pi_1^*\{\pi^{(\varepsilon,q,p)}/\beta\},\ldots,\pi_m^*\{\pi^{(\varepsilon,q,p)}/\beta\}$ coincide since in that case the variable β does not occur free in $\pi_1^*,\ldots,\pi_m^*$. Therefore

$$D[M] \twoheadrightarrow_{\mathsf{s}} C''[C'[(M(\tau))^*]] \twoheadrightarrow_{\mathsf{s}} C''[C'[\mathsf{Hnf}((M(\tau))^*)]] \twoheadrightarrow_{\mathsf{s}} (\pi_{j_l}^*\{\pi^{(\varepsilon,q,p)}/\beta\})[j'_l - 1] \twoheadrightarrow_{\mathsf{s}} (M(\sigma))^*\{\pi^{(\varepsilon,q,p)}/\beta\}$$

Finally we remark that $M(\sigma)$ has a proper hnf iff $(M(\sigma))^*\{\pi^{(\varepsilon,q,p)}/\beta\}$ has a proper hnf. This concludes the proof. $\qquad\square$

Note that if $\sigma \in \mathrm{dom}(M)$ is non-empty, then for every proper prefix $\tau < \sigma$, the term $M(\tau)$ must have a proper hnf. This fact allows the "navigation" of the Böhm tree of M implemented in Lemma 3.6. The improper hnf's do not play the same role that head normal forms have in the λ-calculus.

Now we want to look at terms as maps which are defined also at nodes reachable by the suitable amount of η-expansions. To this end the following concept of *path expansion* will be used to define these maps.

Definition 3.9 (Path expansion). *Let σ be a sequence and let M be a term. We define the* path expansion

$\langle M \| \sigma \rangle$ of M by σ *by induction on the length of* σ *as follows:*

$\langle M \| \sigma \rangle := M$ *if* σ *is the empty sequence*

$$\langle M \| (j, j') \cdot \tau \rangle := \begin{cases} \mu \vec{\alpha}.\beta[n] \star \pi_1 \star \cdots \star (\vec{N} \centerdot \gamma[k] \centerdot \ldots \centerdot \langle \gamma[k+j'-1] \| \tau \rangle \centerdot \gamma[k+j']) \star \cdots \star \pi_m \\ \quad \textit{if } \mathsf{Hnf}(M) = \mu \vec{\alpha}.\beta[n] \star \pi_1 \star \cdots \star \pi_m, \ j \le m \textit{ and} \\ \quad \vec{N} \centerdot \gamma[k] \textit{ is the canonical form of } \pi_j \textit{ and } j' > \sharp \vec{N} \\ \\ \mu \vec{\alpha} \gamma_1 \ldots \gamma_{j-m}.\beta[n] \star \pi_1 \star \cdots \star \pi_m \star \gamma_1 \star \cdots \star (\gamma_{j-m}[0] \centerdot \ldots \centerdot \langle \gamma_{j-m}[j'-1] \| \tau \rangle \centerdot \gamma_{j-m}[j']) \\ \quad \textit{if } \mathsf{Hnf}(M) = \mu \vec{\alpha}.\beta[n] \star \pi_1 \star \cdots \star \pi_m \textit{ and } j > m \\ \\ \textit{undefined} \quad \textit{otherwise} \end{cases}$$

Definition 3.10. *Given a term M we define a partial map $M_{(.)} : \mathsf{Seq} \rightharpoonup \Sigma^{\mathsf{t}}$ (extending that of Definition 3.5) as follows:*

$$M_\sigma := \begin{cases} \langle M(\tau) \| \tau' \rangle (\tau') & \textit{if } \tau \textit{ is the longest prefix of } \sigma \textit{ such that } \tau \in \mathsf{dom}(M), \ \sigma = \tau \cdot \tau' \textit{ and} \\ & \langle M(\tau) \| \tau' \rangle \textit{ is defined} \\ \\ \textit{undefined} & \textit{otherwise} \end{cases}$$

The map $M_{(.)} : \mathsf{Seq} \rightharpoonup \Sigma^{\mathsf{t}}$ contains information about all possible η-expansions of the Böhm tree of M. We let $\mathsf{vir}(M) = \{\sigma \in \mathsf{Seq} : M_\sigma \text{ is defined }\}$. Note that $\mathsf{dom}(M) \subseteq \mathsf{vir}(M)$ and for every $\sigma \in \mathsf{dom}(M)$, the values M_σ and $M(\sigma)$ coincide. The elements of $\mathsf{vir}(M)$ are the *virtual sequences* of M. The map $M_{(.)}$ extends $M(\cdot)$ by giving also *virtual nodes*, which are intuitively nodes of some η-expansion of the Böhm tree of M, but still $M_{(.)}$ cannot return the unreachable nodes, that correspond to sequences in $\mathsf{Seq} - \mathsf{vir}(M)$, which do not belong to any of the η-expansions of the Böhm tree of M. Note that for the maximal sequences $\sigma \in \mathsf{dom}(M)$, the term M_σ may have an improper hnf or not have an hnf at all, while for non-maximal $\sigma \in \mathsf{dom}(M)$, the term M_σ must have a proper hnf.

Lemma 3.7. *Let M, N be terms. If $\sigma \in \mathsf{vir}(M) \cap \mathsf{vir}(N)$ is minimal such that $M_\sigma \not\sim N_\sigma$, then there exists a head context $C[\cdot]$ such that $C[M] \not\sim C[N]$. Moreover $C[M]$ has a proper hnf iff M_σ has a proper hnf and $C[N]$ has a proper hnf iff N_σ has a proper hnf.*

Proof. If σ is the empty sequence then the result is trivial. Now assume $\sigma = \sigma'(j, j')$, so that $M_{\sigma'} \sim N_{\sigma'}$ and the similarity also holds for all prefixes of σ'. Let τ be the longest prefix of σ' contained in $\mathsf{dom}(M)$ and let τ' be such that $\sigma' = \tau \tau'$. Let ρ be the longest prefix of σ' contained in $\mathsf{dom}(N)$ and let ρ' be such that $\sigma' = \rho \rho'$. We assume w.l.o.g. that $\tau \ge \rho$. Let $n = \sharp \sigma$, let p be greater than $\mathsf{w}(M, n)$, $\mathsf{w}(N, n)$ and of all the second components of the pairs occurring in σ. Let q be greater than $\mathsf{b}(M, n)$, $\mathsf{b}(N, n)$ and of all the first component of the pairs occurring in σ. Let $C[\cdot]$ be the context produced by Lemma 3.6 applied to M, n, ρ, q and p. Then $C[M]$ and $C[N]$ reduce, respectively, to substitution instances $(M(\rho))^*$ and $(N(\rho))^*$ where the same substitutions have been applied. Therefore $(M(\rho))^* \sim (N(\rho))^*$ and $C[M] \sim C[N]$. Now let $C'[\cdot]$ be the context produced by Lemma 3.6 applied to $\langle (N(\rho))^* \| \rho' \rangle$, $n' = \sharp \rho'$, τ', q and p. Once again $C'[C[N]]$ reduces to a substitution instance $(\langle N(\rho)^* \| \rho' \rangle (\rho'))^o$ and, because $M_{\sigma'} \sim N_{\sigma'}$, we have $C'[C[M]] \sim C'[C[N]]$. Finally let $C''[\cdot]$ be the context produced by Lemma 3.6 applied to $(\langle N(\rho)^* \| \rho' \rangle (\rho'))^o$, $n'' = 1$, $\tau'' = (j, j')$, q and p. Then the context $C''[C'[C[\cdot]]]$ has the properties required in the statement, since $C''[C'[C[M]]] \not\sim C''[C'[C[N]]]$, $C''[C'[C[M]]]$ has a proper hnf iff M_σ has a proper hnf and $C''[C'[C[N]]]$ has a proper hnf iff N_σ has a proper hnf. $\qquad \square$

Theorem 3.8 (Böhm's theorem for the extended stack calculus). *Let M, N be two distinct $\to_{s\eta}$-normal forms without subterms which are improper hnf's. Then M and N are separable.*

Proof. Under the hypotheses of the statement about M and N, there has to be a minimal sequence $\sigma \in$ vir$(M) \cap$ vir(N) such that $M_\sigma \not\sim N_\sigma$ so that by Lemma 3.7 there exists a head context $C[\cdot]$ such that $C[M] \not\sim C[N]$. Now by hypothesis M_σ and N_σ are proper hnf's and hence $C[M]$ and $C[N]$ have proper hnf's. Therefore, applying Theorem 3.5 we have that $C[M]$ and $C[N]$ are separable, which trivially implies that M and N are separable. □

Supppose that there is a $\sigma \in$ dom$(M) \cap$ dom(N) such that $M(\sigma) = \mu\vec{\alpha}.\text{nil}[h] \star \vec{\pi}$ and $N(\sigma) = \mu\vec{\alpha}'.\text{nil}[h'] \star \vec{\omega}$. Assume P is the j'-th term of the j-th stack of $\vec{\pi}$ and Q is the j'-th term of the j-th stack of $\vec{\omega}$ and that $P \not\sim Q$. If P and Q are the only dissimilar subterms then no separating context can be built with the technique described in Lemma 3.6. As a matter of fact the requirement, appearing in the statement of Theorem 3.8, that M and N do not have subterms which are improper hnf's is more strict then necessary. In fact we only need that, among the dissimilar subterms, there is a $\sigma \in$ vir$(M) \cap$ vir(N) such that $M_\sigma \not\sim N_\sigma$ and M_σ, N_σ are proper hnf's.

We decided to study the extension of the stack calculus as defined in [7], thus including nil in the language. However nil received a special treatment throughout this paper, in the sense that the improper hnf's are kept do not play a role similar to the proper hnf's (see for example Definition 3.1, Definition 3.4 and Definition 3.5). The reason is that improper hnf's and terms without hnf are, in the extended stack calculus, in some sense comparable to what unsolvable terms are in the λ-calculus. In any case, we do not treat a notion of solvability for the extended stack calculus.

3.4 Characterization of operational equivalence

In this section we give a concrete characterization of operational equivalence that does not involve any universal quantification over head contexts (see Definition 2.1). Rather, this characterization is based on Böhm trees.

Definition 3.11. *We define a binary relation $\overset{\infty}{\sim}$ on Σ^{t} as follows: $M \overset{\infty}{\sim} N$ iff vir$(M) =$ vir(N) and for all $\sigma \in$ vir(M) we have that $M_\sigma \sim N_\sigma$ (up to rename of variables which are bound in some M_τ, N_τ, where $\tau < \sigma$, but free in M_σ, N_σ).*

Theorem 3.9. *$M \approx N$ iff $M \overset{\infty}{\sim} N$.*

Proof. ($\Rightarrow$) We prove the contrapositive. Suppose $M \overset{\infty}{\not\sim} N$. Then there exists a sequence σ such that $M \not\sim_\sigma N$. Let σ be minimal w.r.t. this property. Then $\sigma \in$ vir$(M) \cap$ vir(N), so that in fact $M_\sigma \not\sim N_\sigma$.

Since M_σ and N_σ are not similar, at least one among them must have a proper hnf: say it is M_σ. Now suppose N_σ does not have a proper hnf. By Lemma 3.7 there exists a head context $C[\cdot]$ such that $C[M]$ has a proper hnf while $C[N]$ does not have a proper hnf. This proves that $M \not\approx N$.

If N_σ has a proper hnf, then by Lemma 3.7 there exists a head context $C'[\cdot]$ such that $C'[M] \not\sim C'[N]$ and both $C'[M], C'[N]$ have a proper hnf; therefore applying Theorem 3.5 we obtain that $C'[M]$ and $C'[N]$ are separable and consequently by Theorem 3.3, we have $C'[M] \not\approx C'[N]$. This trivially implies $M \not\approx N$.

($\Leftarrow$) Immediate, because by Theorem 3.2, the relation $\approx$ is an HP-complete equational theory which, by Theorem 3.9, is contained in $\overset{\infty}{\sim}$. □

4   Conclusions

The stack calculus [7] is a finitary functional language in which the $\lambda\mu$-calculus can be faithfully translated, in the sense that conversion (and typing, for the typed versions) is preserved by the translation. As it happens for the $\lambda\mu$-calculus, the stack calculus fails to have the separation property and in this paper we introduce the extended stack calculus which, as Saurin's $\Lambda\mu$-calculus [23], does have this property. The separation property proved in this paper for the extended stack calculus has consequences both on the semantical and on the syntactical side. For example it implies that $=_{s\eta}$ is the maximal consistent congruence on normalizable terms extending $=_s$, so that any model of the extended stack calculus cannot identify two different $s\eta$-normal forms without being trivial.

Nonetheless the definition of operational equivalence involves a universal quantification over contexts but the problem of checking operational equivalence between nil-free normalizable terms reduces to the problem of finding their $s\eta$-normal forms (Theorem 3.8) (with a leftmost strategy, for example). The complete characterization of operational equivalence (also for non-normalizable terms) is achieved: two terms of the extended stack calculus are operationally equivalent iff they have the same Böhm tree, up to possibly infinite η-expansion (Theorem 3.9), a condition that does not involve a quantification over all head contexts. We showed that operational equivalence is maximally consistent, i.e. it cannot be properly extended to another consistent equational theory both in the stack calculus and in the extended stack calculus. We work out the details of a Böhm-out technique for the extended stack calculus (Lemma 3.6). A nice feature of the extended stack calculus is that, having only one binder, it admits a simpler proof of Böhm's theorem, which is similar to the one for the λ-calculus. Besides the applications of Böhm's theorem, there has always been interest around the proof itself and the algorithmic content of the Böhm-out technique: from Huet's [16] interest in the implementation and mechanical formalization of Böhm's proof, Aehlig and Joachimski [1] alternative proof and to Dezani et al.'s account [9]. Saurin in [24] establishes a standardization theorem for the $\Lambda\mu$-calculus, and studies Böhm-like trees for the $\Lambda\mu$-calculus, strengthening the separation results that he obtained in [23]. In view of the interest in proofs of Böhm's theorem for various calculi, we believe useful to contribute in the present work with a direct proof of Böhm's theorem (i.e. with a Böhm-out technique) for the extended stack calculus, even if the mere separation result would follow by a suitable mutual translation with the $\Lambda\mu$-calculus.

It is out of the scope of this paper to analyze the typed extended stack calculus. In the case of Saurin's $\Lambda\mu$ the Curry–Howard isomorphism carries through via the straightforward extension of the type system. However Saurin [25] and Nakazawa and Katsumata [18] noted that with this approach many "interesting" $\Lambda\mu$-terms are not typeable (for example those used in the Böhm out technique). For this reason Saurin [25] studies an alternative type system for the $\Lambda\mu$-calculus. This latter approach could also be adapted to the extended stack calculus.

References

[1] K. Aehlig & F. Joachimski (2002): *A note on Böhm's theorem*. Available at `http://www.mathematik.uni-muenchen.de/~joachski/boehm.pdf.gz`.

[2] H.P. Barendregt (1984): *The Lambda calculus: its syntax and semantics*. North-Holland.

[3] C. Böhm (1968): *Alcune proprietà delle forme $\beta\eta$-normali nel λK-calcolo. Pubblicazioni dell'IAC* 696.

[4] C. Böhm, M. Dezani-Ciancaglini, P. Peretti & S. Ronchi Della Rocca (1979): *A discrimination algorithm inside lambda-beta-calculus*. Theoretical Computer Science 8, pp. 265–292, doi:10.1016/0304-3975(79)90014-8.

[5] C. Böhm & W. Gross (1966): *Introduction to the CUCH*. In E. Caianiello, editor: *Automata Theory*, Academic Press, London, UK, pp. 35–65.

[6] C. Böhm, A. Piperno & S. Guerrini (1994): *Lambda-Definition of Function(al)s by Normal Forms*. In D. Sannella, editor: *ESOP, Lecture Notes in Computer Science* 788, Springer-Verlag, Berlin, Germany, pp. 135–149, doi:10.1007/3-540-57880-3_9.

[7] A. Carraro, T. Ehrhard & A. Salibra (2012): *The Stack calculus*. Submitted to LSFA12.

[8] R. David & W. Py (2001): *λµ-Calculus and Böhm's Theorem*. *J. Symb. Log.* 66(1), pp. 407–413, doi:10.2307/2694930.

[9] M. Dezani-Ciancaglini, S. Guerrini & A. Piperno (2009): *Böhm's Theorem*, pp. 1–15. World Scientific Pub. Co. Inc. Available at `http://www.di.unito.it/~dezani/papers/dgp.pdf`.

[10] M. Dezani-Ciancaglini, U. De' Liguoro & A. Piperno (1996): *Filter Models for Conjunctive-disjunctive λ-calculi*. *Theoretical Computer Science* 170(1-2), pp. 83–128, doi:10.1016/S0304-3975(96)00235-6.

[11] M. Dezani-Ciancaglini, J. Tiuryn & P. Urzyczyn (1999): *Discrimination by parallel observers: the algorithm*. *Information and computation* 150(2), pp. 153–186, doi:10.1006/inco.1998.2773.

[12] M. Felleisen (1990): *On the expressive power of programming languages*. In: *ESOP*, LNCS, pp. 134–151, doi:10.1007/3-540-52592-0_60.

[13] T. Griffin (1990): *A Formulae-as-Types Notion of Control*. In: *POPL*, pp. 47–58, doi:10.1145/96709.96714.

[14] C. Hankin (1995): *Lambda Calculi: a guide for computer scientists*. Oxford University Press.

[15] W.A. Howard (1980): *The formulas-as-types notion of construction*. In J.R. Hindley & J.P. Seldin, editors: *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pp. 479–490.

[16] G.P. Huet (1993): *An Analysis of Böhm's Theorem*. *Theor. Comput. Sci.* 121(1&2), pp. 145–167, doi:10.1016/0304-3975(93)90087-A.

[17] J.M.E. Hyland (1976): *A syntactic characterization of the equality in some models of the λ-calculus*. *J. London Math. Soc.* 2(12), pp. 361–370, doi:10.1112/jlms/s2-12.3.361.

[18] S. Katsumata & K. Nakazawa (2012): *Extensional models of untyped λµ-calculus*. In: *CL& C*, doi:10.4204/EPTCS.97.3.

[19] G. Manzonetto & M. Pagani (2011): *Böhm's theorem for resource lambda calculus through Taylor expansion*. In: *TLCA*, pp. 153–168, doi:10.1007/978-3-642-21691-6_14.

[20] M. Parigot (1991): *Free Deduction: An Analysis of "Computations" in Classical Logic*. In: *RCLP*, pp. 361–380, doi:10.1007/3-540-55460-2_27.

[21] D. Prawitz (1965): *Natural Deduction - a proof theoretical study*. Almqvist & Wiksell, Stokholm.

[22] D. Sangiorgi (1994): *The lazy lambda calculus in a concurrency scenario*. *Information and computation* 111(1), pp. 120–153, doi:10.1006/inco.1994.1042.

[23] A. Saurin (2005): *Separation with streams in the Λµ-calculus*. In: *LICS*, pp. 356–365, doi:10.1109/LICS.2005.48.

[24] A. Saurin (2010): *Standardization and Böhm Trees for Λµ-calculus*. In: *FLOPS*, pp. 134–149, doi:10.1007/978-3-642-12251-4_11.

[25] A. Saurin (2010): *Typing streams in the Λµ-calulus*. *ACM Trans. Comput. Log.* 11(4), doi:10.1145/1805950.1805958.

[26] J.P. Seldin & J. Hindley (1986): *Introduction to combinators and λ-calculus*. Cambridge University Press.

[27] T. Streicher & B. Reus (1998): *Classical logic, continuation semantics and abstract machines*. *J. Funct. Program.* 6(8), pp. 543–572, doi:10.1017/S0956796898003141.

[28] C.P. Wadsworth (1976): *The relation between computational and denotational properties for Scott's D_∞-models of the λ-calculus*. *SIAM Journal on Computing* 5(3), pp. 488–521, doi:10.1137/0205036.