ELSEVIER

Theoretical Computer Science 212 (1999) 233–246

Theoretical Computer Science

An algebraic view of the Böhm-out technique[1]

Adolfo Piperno *

Dipartimento di Scienze dell'Informazione, Università di Roma "La Sapienza", Via Salaria 113, 00198 Roma, Italy

Abstract

Using an algebraic representation of closed β-normal forms in λ-calculus, the Böhm's theorem is rephrased as an equality predicate between elements of a term algebra. The presented algebraic interpretation gives new insight into the Böhm-out technique and allows for original applications of the method. © 1999—Elsevier Science B.V. All rights reserved

Keywords: Lambda-calculus; Böhm trees; Combinatory Equations; Separability

1. Introduction

The Böhm-out technique is perhaps one of the most applied results in λ-calculus, since it is used in studies about several different aspects of the system. In effect, while stated as a syntactical property of closed normal forms [3], the Böhm's theorem finds its relevance in its main semantical consequence: two different terms having normal form are not identifiable in a nontrivial model of the λ-calculus.

The method used in the proof of the Böhm's theorem, named Böhm-out by Barendregt in his book [1], roughly consists in extracting subterms (or substitution instances of them) from normal forms. It can be considered as a tool for analyzing the information content of a λ-term. This has in its turn applications in the analysis of both syntactical and semantical aspects of λ-calculus: as examples, consider the solvabilty problem for systems of combinatory equations and the problem of comparing the behaviour of applicative programs, respectively. Indeed, when a combinator is searched for some purposes, it is often found using some variant of the Böhm-out technique.

Different presentations of the Böhm-out technique appear in the literature [1, 10–12], each of which focuses on particular aspects of the method. Moreover, Böhm's theorem has been extended in various directions [9, 7, 13].

[1] This work has been partially supported by grants from ESPRIT-BRA 7232 "Gentzen".

* E-mail: piperno@dsi.uniroma1.it.

PII: S0304-3975(98)00142-X

In this paper, an algebraic interpretation of the Böhm-out technique will be given. To introduce and motivate it, let us first recall the Böhm's theorem:

Let M, N be two closed β-normal forms, and u, v any two different objects. There exists a λ-term Δ (*a discriminator* for M and N) such that $\Delta M =_\beta u$, $\Delta N =_\beta v$ iff M and N are not η-convertible

Taking u, v to be (any pair of terms λ-defining) the Booleans `true` and `false`, one can view the discriminator Δ as the λ-definition of the (partial, namely defined for M and N only) predicate "to be equal to M".

On the other hand, assume that there exists a term ∇ (a veritable equality predicate between normal forms) such that

$$\nabla MN = \begin{cases} u \text{ if } M =_\eta N, \\ v \text{ otherwise.} \end{cases} \tag{1}$$

Then clearly $\Delta = \nabla M$ discriminates between M and any other term, hence it is a discriminator for M and N.

Some questions naturally arise here: up to what extent can the discriminator be considered to λ-define an equality predicate between normal forms? More precisely, given a finite set $\mathcal{N}$ of β-normal forms, does there always exist a term ∇ satisfying (1) over $\mathcal{N}$? Moreover, are there infinite sets of normal forms for which ∇ can be defined?

This paper answers affirmatively all these questions, giving a representation of the Böhm's theorem as the process of identifying an equality predicate for sets of normal forms. To this aim, an algebraic representation of normal forms is used, first introduced in [6], which mimics the construction of Böhm trees. The presented method also allows to extract subterms from normal forms in a "clean" way. In effect, the Böhm-out technique allows for extracting substitution instances of subterms, since during the extraction process some variables might be replaced by terms (see [1, Section 10.3.1]). In this paper it will be proved that it is possible to extract subterms from normal forms (in fact for infinite Böhm trees having a bound on the number of sons, too) without performing any substitution over them, provided that the system is equipped with the η-rule.

This work is inspired by similar results obtained (for a restricted class of normal forms, namely for *proper* combinators) by Corrado Böhm [5] and presented, *in nuce*, during the evening lecture of the LICS conference in Paris, 1994.

The paper is organized as follows. In the Section 2 the λ-calculus will be extended with algebraic features and some useful results will be proved for the extended calculi. In Section 3, the algebraic representation of closed normal forms will be introduced and the main technical result of the paper will be proved, namely the definability of a translation from normal forms to their algebraic counterpart. The Böhm-out lemma and the Böhm's theorem will be reinterpreted in Section 4 as straightforward consequences of the definability theorem. Technical details of proofs are deferred to Section 5.

2. Extended λ-calculi

The reader will be assumed to be familiar with the basic notions and properties of λ-calculus (see [1]). In particular, conventional notations will be used for β- and η-reduction and equality, while the symbol $\equiv$ will be used for syntactical identity.

As usual, we shall consider the set Λ of terms of the λ-calculus to be described by the following BNF, where x ranges over a denumerable set of variables:

$$L ::= x \mid (\lambda x.L) \mid (L_1 L_2). \tag{2}$$

Let Σ be a set of function symbols from a given signature. $\Lambda(\Sigma)$ denotes the set of *extended lambda terms* with symbols from the signature Σ. To be precise $\Lambda(\Sigma)$ can be defined by adding the following clause to the clauses (2) for the formation of lambda terms: if $t_1,\dots,t_n \in \Lambda(\Sigma)$ and $f \in \Sigma$ is an n-ary function symbol, then $f(t_1,\dots,t_n) \in \Lambda(\Sigma)$. Note that $\mathsf{Ter}(\Sigma) \subseteq \Lambda(\Sigma)$ where $\mathsf{Ter}(\Sigma)$ is the set of terms over the signature Σ.

A *λ-interpretation* of Σ is a function $\phi : \Sigma \to \Lambda$. Any such λ-interpretation ϕ induces a map $(\cdot)^\phi : \Lambda(\Sigma) \to \Lambda$ in the obvious way, namely

$$\begin{aligned}
\text{for any variable } x,\ x^\phi &= x,\\
(\lambda x.M)^\phi &= \lambda x.M^\phi,\\
(MN)^\phi &= M^\phi N^\phi,\\
f(M_1,\dots,M_n)^\phi &= f^\phi M_1^\phi \dots M_n^\phi.
\end{aligned}$$

Definition 1. Let $\mathscr{E} = \{a_i = b_i | i \in J\}$ be a set of equations between extended λ-terms $a_i, b_i \in \Lambda(\Sigma)$.

A λ-interpretation ϕ *satisfies* (or *solves*) $\mathscr{E}$ if $a_i^\phi =_\beta b_i^\phi$, for each equation $a_i = b_i$ in $\mathscr{E}$. If there exists a λ-interpretation ϕ which satisfies $\mathscr{E}$, then $\mathscr{E}$ can be *solved* inside λ-calculus and ϕ is a *solution* for $\mathscr{E}$.

Definition 2. A λ-interpretation $\phi : \Sigma \to \Lambda$ of a signature Σ is *adequate* for $\mathsf{Ter}(\Sigma)$ iff there exists a term $It_{\Sigma,\phi} \in \Lambda(\Sigma)$ such that $It_{\Sigma,\phi}\, T^\phi =_\beta T$, for any $T \in \mathsf{Ter}(\Sigma)$.

Definition 3. (i) Let $\mathscr{E}$ be a set of equations in an extended λ-calculus $\Lambda(\Sigma)$. $\mathscr{E}$ is *canonical* if the function symbols in Σ can be partitioned into two disjoint subsets $\Sigma = \Sigma_0 \cup \Sigma_1$ so that, letting $\Sigma_0 = \{c_1,\dots,c_r\}$ and $\Sigma_1 = \{f_1,\dots,f_k\}$, each equation $t = t'$ of $\mathscr{E}$ has the form

$$f_i(c_j(x_1,\dots,x_m), y_1,\dots,y_n) = b_{i,j}, \tag{3}$$

where $f_i \in \Sigma_1$, $c_j \in \Sigma_0$, $b_{i,j} \in \Lambda(\Sigma)$ is a term depending on i and j, $n,m \geqslant 0$ and the variables $x_1,\dots,x_m$, $y_1,\dots,y_n$ are all distinct.

(ii) The elements of Σ_0 are called *data constructors* and those of Σ_1 *programs*.

Example 4 (*Useful data structures*). Using superscripts to indicate arities of function symbols, let

$$\Sigma_{\mathbb{N}} = \{zero^{(0)}, succ^{(1)}\},$$

$$\Sigma_{string(r)} = \{\#^{(0)}, char_1^{(1)}, \ldots, char_r^{(1)}\},$$

$$\Sigma_{bool} = \{tt^{(0)}, ff^{(0)}\}.$$

Clearly, each element of $\mathsf{Ter}(\Sigma_{\mathbb{N}})$ can be interpreted as a natural number, each element of $\mathsf{Ter}(\Sigma_{string(r)})$ can be interpreted as a string over a r-character alphabet (with # corresponding to the empty string), while each element of $\mathsf{Ter}(\Sigma_{bool})$ can be interpreted as a Boolean.

Now let $\Sigma = \Sigma_{\mathbb{N}} \cup \Sigma_{string(r)} \cup \{length^{(1)}\}$. The definition of the function which associates to any above considered string its length can be expressed by the following canonical system of equations:

$$\mathscr{E} = \{length(\#) = zero\} \cup \{length(char_i(y)) = succ(length(y)) \mid 1 \leqslant i \leqslant r\}.$$

The following example shows that binary functions can be defined by means of canonical systems of equations, provided that auxiliary function symbols are considered: let $\Sigma = \Sigma_{bool} \cup \{And^{(2)}, And_1^{(1)}\}$; the following system is a canonical one and it clearly expresses the definition of the Boolean function *AND*.

$$\mathscr{E}' = \left\{ \begin{array}{ll} And(tt, y) = And_1(y), & And_1(tt) = tt, \\ And(ff, y) = ff, & And_1(ff) = ff \end{array} \right\}.$$

2.1. More on canonical systems

The following examples are needed to express the Böhm's theorem as the equality predicate over elements of a term algebra and the Böhm-out lemma.

The equality predicate for elements of $\mathsf{Ter}(\Sigma)$: Let $\Sigma = \{c_1^{(a_1)}, \ldots, c_r^{(a_r)}\}$ and $\Sigma' = \Sigma \cup \Sigma_{bool} \cup \{Eq^{(2)}, Eq_1^{(a_1+1)}, \ldots, Eq_r^{(a_r+1)}, And\}$. The equality predicate between elements of $\mathsf{Ter}(\Sigma)$ can be expressed by means of the following system of equations ($1 \leqslant i, j \leqslant r$):

$$Eq(c_i(x_1, \ldots, x_{m_i}), y) = Eq_i(y, x_1, \ldots, x_{m_i}),$$

$$Eq_i(c_j(y_1, \ldots, y_{m_j}), x_1, \ldots, x_{m_i})$$

$$= \begin{cases} ff & \text{if } i \neq j, \\ tt & \text{if } i = j \text{ and } m_i = m_j = 0, \\ And(\ldots(And(Eq(x_1, y_1), Eq(x_2, y_2))), \ldots, Eq(x_{m_i}, y_{m_j})) & \\ \quad \text{otherwise (here } m_i = m_j). & \end{cases}$$

The subterm extraction for elements of $\mathsf{Ter}(\Sigma)$: Let $\Sigma = \{c_1, \ldots, c_r\}$ and $\Sigma' = \Sigma \cup \Sigma_{string(k)} \cup \{Xtr^{(2)}, Xtr_1^{(2)}, \ldots, Xtr_r^{(2)}, error^{(0)}\}$. The subterm extraction for elements of $\mathsf{Ter}(\Sigma)$ can be expressed by means of the following system of equations, where a string describes the path identifying the subterm to be extracted:

$$
\begin{aligned}
&Xtr(\#, y) &&= y,\\
&Xtr(char_j(x), y) &&= Xtr_j(y,x),\\
&Xtr_j(c_i(y_1,\dots,y_{m_i}),x) &&= \begin{cases} error & \text{if } j > m_i,\\ Xtr(x, y_j) & \text{otherwise.}\end{cases}
\end{aligned}
$$

2.2. Solution of canonical systems

Any canonical system has a solution inside λ-calculus.

Theorem 5. *Let $\Lambda(\Sigma)$ be an extended λ-calculus; then every canonical set of equations $\mathscr{E}$ has a solution $\phi : \Sigma \to \Lambda$ inside λ-calculus. Furthermore, ϕ can be chosen in a way that the restriction $\phi|\Sigma_0$ depends only on Σ_0 and not on $\mathscr{E}$, namely there is a fixed representation of the constructors.*

Proof. The proof of this theorem, which appeared in [2, 8], is reported in Section 5. Observe that the proposed solution is a term in normal form. □

Corollary 6. *There exist two β-normal forms Eq_Σ and Xtr_Σ which solve the systems of equations defining the equality predicate and the subterm extraction function, respectively. Clearly, these terms depend on Σ and on the representation of Σ_0.*

Proof. Directly from Theorem 5. □

Lemma 7. *Let $\Sigma = \{c_1,\dots,c_r\}$. The λ-interpretation $\phi_\Sigma : \Sigma \to \Lambda$ such that, for $1 \leqslant j \leqslant r$,*

$$\phi_\Sigma(c_j) = \lambda x_1 \dots x_{m_j} e.e\,\mathbf{U}_j^r x_1 \dots x_{m_j}, \tag{4}$$

where m_j is the arity of c_j and $\mathbf{U}_j^r \equiv \lambda x_1 \dots x_r.x_j$, is adequate for $\mathsf{Ter}(\Sigma)$.

Proof. Follows directly from the proof of Theorem 5. Consider the canonical system defining the identity function over $\mathsf{Ter}(\Sigma)$:

$$\mathscr{E} = \{Id(c_j(x_1,\dots,x_{m_j})) = c_j(Id(x_1),\dots,Id(x_{m_j})) \mid 1 \leqslant j \leqslant r\}$$

and replace the constructors appearing in the right-hand sides of the equations with fresh free variables $v_1,\dots,v_r$, turning the algebraic notation into the usual λ notation for application. The system thus becomes

$$\mathscr{E}' = \{Id'(\phi_\Sigma(c_j)(x_1,\dots,x_{m_j})) = v_j(Id'(x_1))\dots(Id'(x_{m_j})) \mid 1 \leqslant j \leqslant r\}.$$

Such a system can be solved following the pattern of the proof of Theorem 5. The solution is therefore a pair $\langle t_1, t_1\rangle$ where $t_1 = \langle t_{1,1}, \ldots t_{1,r}\rangle$. It comes out that

$$\begin{array}{ll} & \langle t_1,t_1\rangle((\lambda x_1 \ldots x_{m_j} e.e\,\mathbf{U}^r_j x_1 \ldots x_{m_j})x_1 \ldots x_{m_j}) \\ \xrightarrow{\beta}\!\!\!\!\rightarrow & (\lambda e.e\,\mathbf{U}^r_j x_1 \ldots x_{m_j})t_1 t_1 \\ \xrightarrow{\beta}\!\!\!\!\rightarrow & t_{1,j} x_1 \ldots x_{m_j} t_1 \\ & \text{take now } t_{1,j} = \lambda x_1 \ldots x_{m_j} t.v_j(x_1 tt) \ldots (x_{m_j} tt) \\ \xrightarrow{\beta}\!\!\!\!\rightarrow & v_j(x_1 t_1 t_1) \ldots (x_{m_j} t_1 t_1) = v_j(\langle t_1,t_1\rangle x_1) \ldots (\langle t_1,t_1\rangle x_{m_j}). \end{array}$$

The solution of the system $\mathscr{E}'$ is therefore a λ term (a normal form, indeed) in which the variables $v_1, \ldots, v_j$ occur free. The lemma follows taking such a term, substituting in it every variable v_j with

$$\lambda x_1 \ldots x_{m_j}.c_j(x_1, \ldots, x_{m_j}),$$

and β-reducing to the normal form. □

3. Algebraic representation of closed normal forms

Let $\mathsf{NF}^0 \subset \Lambda$ be the set of closed normal forms. The following algebraic representation of closed normal forms has been first proposed in [6] and then reprised in [4].

Definition 8. (i) Define $\Sigma_\pi = \{p_i^{m,n} \mid m \geqslant 0, n > 0, 0 < i \leqslant n\}$, where $p_i^{m,n}$ is an m-ary function symbol.

(ii) Define the interpretation map $\phi_\pi : \Sigma_\pi \to \Lambda$ as follows:

$$\phi_\pi(p_i^{m,n}) = \boldsymbol{\pi}_i^{m,n} \equiv \lambda y_1 \ldots y_m x_1 \ldots x_n.x_i(y_1 x_1 \ldots x_n) \ldots (y_m x_1 \ldots x_n).$$

(iii) Define the representation map $f : \mathsf{NF}^0 \to \mathsf{Ter}(\Sigma_\pi)$ as follows:

$$f(\lambda x_1 \ldots x_n.x_i M_1 \ldots M_m) = p_i^{m,n}(f(\lambda x_1 \ldots x_n.M_1), \ldots, f(\lambda x_1 \ldots x_n.M_m)).$$

Theorem 9. *For any* $M \in \mathsf{NF}^0$,

$$(f(M))^{\phi_\pi} \xrightarrow{\beta}\!\!\!\!\rightarrow M.$$

Proof. It is an easy induction on the depth of the Böhm tree of M. □

Example 10. Let $M \equiv \lambda xyz.xz(\lambda a.y(\lambda b.x)a)(\lambda c.c)$, whose Böhm tree $(BT(M))$ is shown in Fig. 1 (left). The representation of M in $\mathsf{Ter}(\Sigma_\pi)$ can be obtained as follows:

- Consider the tree whose structure is the same as $BT(M)$ and labels are pairs of integers (n, i), where (see Fig. 1 (center))
  - n is the total amount of λ's from the root to the corresponding node of $BT(M)$;

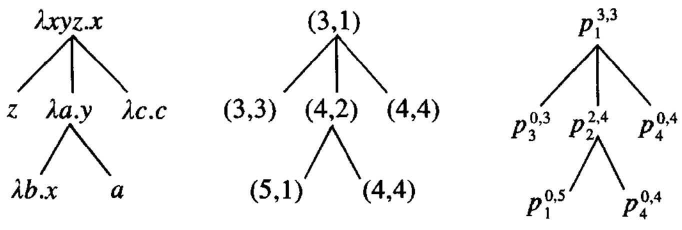

Fig. 1. A closed normal form and its algebraic counterpart.

- ∘ the head variable of the label of the corresponding node in $BT(M)$ is bound by the ith λ starting from the root of $BT(M)$.
- For any node in the obtained tree, substitute the label (n,i) with $p_i^{m,n}$, where m is the amount of immediate subtrees of the tree rooted in the considered node (see Fig. 1 (right)).

It comes out that the representation of M in $\mathsf{Ter}(\Sigma_\pi)$ is

$$p_1^{3,3}\, p_3^{0,3}(p_2^{2,4}\, p_1^{0,5}\, p_4^{0,4})\, p_4^{0,4} \equiv f(M)$$

and it is easy to verify that

$$(f(M))^{\phi_\pi} \equiv \pi_1^{3,3}\pi_3^{0,3}(\pi_2^{2,4}\pi_1^{0,5}\pi_4^{0,4})\pi_4^{0,4} \xrightarrow{\beta}\!\!\!\!\rightarrow M.$$

A canonical system clearly consists of a finite number of equations. In the following definitions, a finite amount of constructors of normal forms is isolated, together with the subset of normal forms they are able to represent.

Definition 11. For any $\mu, \nu \in \mathbb{N}$, define $\Sigma_\nu^\mu \subseteq \Sigma_\pi$:

$$\Sigma_\nu^\mu = \{p_i^{\mu,n} \mid 0 < i \leqslant n \leqslant \nu\} \cup \{p_i^{0,n} \mid 0 < i \leqslant n \leqslant \nu\}.$$

Definition 12. Let $M \in \mathsf{NF}^0$ and $\Sigma_\nu^\mu \subseteq \Sigma_\pi$.

(i) We say that M is *representable* in Σ_ν^μ iff there exists $M' \in \mathsf{NF}^0$ such that

$$M' \xrightarrow{\eta}\!\!\!\!\rightarrow M \quad \wedge \quad f(M') \in \mathsf{Ter}(\Sigma_\nu^\mu).$$

(ii) Define $\mathsf{Rep}(\Sigma_\nu^\mu)$ to be the set of closed normal forms representable in Σ_ν^μ:

$$\mathsf{Rep}(\Sigma_\nu^\mu) = \{M \in \mathsf{NF}^0 \mid \exists M' \in \mathsf{NF}^0 . M' \xrightarrow{\eta}\!\!\!\!\rightarrow M \;\wedge\; f(M') \in \mathsf{Ter}(\Sigma_\nu^\mu)\}.$$

Given any finite set of normal forms, there exists a suitable set of constructors which represents at least all of them.

Fact 13. *For any $\mathscr{M} = \{M_1, \ldots, M_n\} \subset \mathsf{NF}^0$, there exist μ, ν such that*

$$\mathscr{M} \subseteq \mathsf{Rep}(\Sigma_\nu^\mu).$$

Proof. All the elements of $\mathscr{M}$ can be η-expanded in a way that any node of their Böhm trees is either a leaf or has a constant amount of branches μ. Call $\mathscr{M}'$ the resulting set of normal forms. The parameter ν can be computed out of $\mathscr{M}'$, taking the maximum n used in the $p_i^{m,n}$'s needed for representing the elements of $\mathscr{M}'$. $\square$

The representation map can be λ-defined into an extended λ-calculus.

Theorem 14 (Definability theorem). *For any $\Sigma_\nu^\mu \subseteq \Sigma_\pi$ there exists $\mathbf{F}_\nu^\mu \in \Lambda(\Sigma_\nu^\mu)$ such that*

(i) *for any $M \in \mathsf{Rep}(\Sigma_\nu^\mu)$,*

$$\mathbf{F}_\nu^\mu M \xrightarrow{\beta}\!\!\!\!\rightarrow f(M') \in \mathsf{Ter}(\Sigma_\nu^\mu),$$

with $M' \xrightarrow{\eta}\!\!\!\!\rightarrow M$;

(ii) *for any $M, N \in \mathsf{Rep}(\Sigma_\nu^\mu)$,*

$$M =_\eta N \;\Rightarrow\; \mathbf{F}_\nu^\mu M, \mathbf{F}_\nu^\mu N \xrightarrow{\beta}\!\!\!\!\rightarrow f(P),$$

with $P \xrightarrow{\eta}\!\!\!\!\rightarrow M, N$.

Proof. This proof is of a very technical nature, hence it is deferred to the Section 5. $\square$

4. An agebraic vew of the Böhm-out technique

Let M be a closed normal form. A subterm N of M is identified by a path γ in the Böhm tree of M. Now let $f(M)$ (see Definition 8) be the representation of M in $\mathsf{Ter}(\Sigma_\nu^\mu)$, where μ, ν are chosen according to Fact 13. The path γ identifies in $f(M)$ a term whose interpretation is

$$N' \equiv \lambda t_1 \ldots t_k.N,$$

where $t_1, \ldots, t_k$ are the free variables of N, which are bound along γ in M.

N' can be extracted out of M as follows (thus, to obtain N, just apply N' to $t_1, \ldots, t_k$):

$$N' =_{\beta\eta} \left(It_{\Sigma_\nu^\mu, \phi_{\Sigma_\nu^\mu}} \left(Xtr_{\Sigma_\nu^\mu} \mathbf{String}\, (\mathbf{F}_\nu^\mu(M))^{\phi_{\Sigma_\nu^\mu}} \right) \right)^{\phi_\pi}.$$

Indeed:

- $\mathbf{F}_\nu^\mu(M)$ β-reduces to an element of $\mathsf{Ter}(\Sigma_\nu^\mu)$;
- $(\mathbf{F}_\nu^\mu(M))^{\phi_{\Sigma_\nu^\mu}} \in \Lambda$ is obtained interpreting the symbols in Σ_ν^μ (which appear in $\mathbf{F}_\nu^\mu(M)$) as in (4);
- $\mathbf{String} \in \Lambda$ is the interpretation of the string identifying the extraction path as it comes out from (4) again;

- $Xtr_{\Sigma^\mu_\nu}\mathbf{String}(\mathbf{F}^\mu_\nu(M))^{\phi_{\Sigma^\mu_\nu}} \in \Lambda$ is the result of the application of the λ-interpretation of the extraction function, defined in Section 2.1. This is a λ-term, precisely the one obtained from the algebraic representation of N' interpreting the constructors as in (4). Such an interpretation, as proved in Theorem 5 allows for the definition of the extraction function.
- $It_{\Sigma^\mu_\nu,\phi_{\Sigma^\mu_\nu}}\left(Xtr_{\Sigma^\mu_\nu}\mathbf{String}(\mathbf{F}^\mu_\nu(M))^{\phi_{\Sigma^\mu_\nu}}\right) \in \mathsf{Ter}(\Sigma^\mu_\nu)$ is therefore the algebraic representation of N', so that N' is finally obtained by means of the interpretation map ϕ_π.

Observe that

$$\mathbf{X} = \lambda xy.\left(It_{\Sigma^\mu_\nu,\phi_{\Sigma^\mu_\nu}}\left(Xtr_{\Sigma^\mu_\nu}y(\mathbf{F}^\mu_\nu(x))^{\phi_{\Sigma^\mu_\nu}}\right)\right)^{\phi_\pi}$$

is not an extended λ-term, but a pure one. More precisely, given any $M \in \mathsf{Rep}(\Sigma^\mu_\nu)$ and any λ-term S interpreting a path in M, $\mathbf{X}MS$ β-reduces to the term

$$N' = \lambda t_1 \ldots t_k.N,$$

where N is the subterm of M identified by the path coded in S, and $t_1,\ldots,t_k$ are the free variables of N.

Similarly, the Böhm's theorem can be expressed as the equality predicate between normal forms in the following way:

$$\mathbf{E}MN =_\beta \left(It_{\Sigma^\mu_\nu,\phi_{\Sigma^\mu_\nu}}\left(Eq_{\Sigma^\mu_\nu}(\mathbf{F}^\mu_\nu(M))^{\phi_{\Sigma^\mu_\nu}}(\mathbf{F}^\mu_\nu(N))^{\phi_{\Sigma^\mu_\nu}}\right)\right)^{\{\phi(tt)=u,\phi(ff)=v\}}$$

so that $\mathbf{E}$ is such that, given any $M,N \in \mathsf{Rep}(\Sigma^\mu_\nu)$, $\mathbf{E}MN$ returns u if $M =_\eta N$ and v otherwise. Observe that $\mathsf{Rep}(\Sigma^\mu_\nu)$ might be an infinite set of terms. Therefore, all the questions raised in Section 1 have been affirmatively answered.

5. Technical details

5.1. Theorem 5

Let $\Lambda(\Sigma)$ be an extended λ-calculus; then every canonical set of equations $\mathscr{E}$ has a solution $\phi : \Sigma \to \Lambda$ inside λ-calculus. Furthermore, ϕ can be chosen in a way that the restriction $\phi|\Sigma_0$ depends only on Σ_0 and not on $\mathscr{E}$, namely there is a fixed representation of the constructors.

Proof. Let $\Sigma_0 = \{c_1, c_2, \ldots, c_r\}$.

For $1 \leqslant j \leqslant r$, we define $\vartheta = \phi|\Sigma_0 : \Sigma_0 \to \Lambda$ as in (4):

$$\vartheta(c_j) = \lambda x_1 \ldots x_m e.e\mathbf{U}^r_j x_1 \ldots x_m, \tag{5}$$

where m is the arity of c_j and $\mathbf{U}^r_j \equiv \lambda x_1 \ldots x_r.x_j$.

It remains to define $\zeta = \phi|\Sigma_1 : \Sigma_1 \to \Lambda$, namely the representation of programs. Without loss of generality, we can assume that $\mathscr{E}$ is complete (i.e., it contains an equation for any pair (i,j)), otherwise add more equations to make it complete.

Let $\Sigma_1 = \{f_1, \ldots, f_k\}$. Consider $k \times r$ lambda terms $t_{i,j}$, $1 \leqslant i \leqslant k$, $1 \leqslant j \leqslant r$ to be defined later. Recall the definition of *Church n-tuple*:

$$\langle M_1, \ldots, M_n \rangle \equiv \lambda x.xM_1 \ldots M_n.$$

For $1 \leqslant i \leqslant k$, let $t_i \equiv \langle t_{i,1}, \ldots, t_{i,r} \rangle$ and define

$$\zeta(f_i) \equiv \langle t_i, t_1, t_2, \ldots, t_k \rangle.$$

Thus, $\zeta(f_i)$ is a Church $k+1$-tuple of Church r-tuples of terms. The lambda terms $t_{i,j}$ are chosen in the only natural way which makes ζ a solution of the canonical system of equations $\mathscr{E}$. More precisely, consider the equation

$$f_i(c_j(x_1, \ldots, x_m), y_1, \ldots, y_n) = b_{i,j}$$

belonging to $\mathscr{E}$ ($b_{i,j} \in \Lambda(\Sigma)$). After applying $\phi = \zeta \circ \vartheta$ the equation becomes

$$\langle t_i, t_1, \ldots, t_k \rangle (c_j^{\phi} x_1 \ldots x_m) y_1 \ldots y_n = b_{i,j}^{\phi}.$$

By definition of Church tuple, this simplifies to

$$c_j^{\phi} x_1 \ldots x_m t_i t_1 \ldots t_k y_1 \ldots y_n = b_{i,j}^{\phi}.$$

Recalling the definition of c_j^{ϕ} we have

$$c_j^{\phi} x_1 \ldots x_m t_i = t_i \mathbf{U}_j^r x_1 \ldots x_m = t_{i,j} x_1 \ldots x_m.$$

Hence, the equation becomes

$$t_{i,j} x_1 \ldots x_m t_1 \ldots t_k y_1 \ldots y_n = b_{i,j}^{\phi}.$$

We can now solve this equation for $t_{i,j}$ by replacing on both sides all the occurrences of $t_1, \ldots, t_k$ by fresh variables $v_1, \ldots, v_k$ and abstracting with respect to all variables present in the left-hand side. More precisely, define

$$t_{i,j} \equiv \lambda x_1 \ldots x_m v_1 \ldots v_k y_1 \ldots y_n.(b_{i,j}^{\vartheta})^{\psi},$$

where $\psi : \Sigma_1 \to \Lambda$ is defined by

$$\psi(f_i) = \langle v_i, v_1, \ldots, v_k \rangle. \tag{6}$$

Note that, for any $V \in \Lambda(\Sigma)$, $V^{\zeta} = V^{\psi}[t_h/v_h]_{1 \leqslant h \leqslant k}$.

With this definition

$$\begin{aligned} t_{i,j} x_1 \ldots x_m t_1 \ldots t_k y_1 \ldots y_n &\overset{\beta}{\twoheadrightarrow} (b_{i,j}^{\vartheta})^{\psi}[t_h/v_h]_{1 \leqslant h \leqslant k} \\ &= b_{i,j}^{\zeta \circ \vartheta} = b_{i,j}^{\phi} \end{aligned}$$

and all the equations will be satisfied. □

5.2. Theorem 14 (*Definability theorem*)

For any $\Sigma^\mu_\nu \subseteq \Sigma_\pi$ there exists $\mathbf{F}^\mu_\nu \in \Lambda(\Sigma^\mu_\nu)$ such that
(i) for any $M \in \mathsf{Rep}(\Sigma^\mu_\nu)$,

$$\mathbf{F}^\mu_\nu M \xrightarrow{\beta}\!\!\!\!\rightarrow f(M') \in \mathsf{Ter}(\Sigma^\mu_\nu),$$

with $M' \xrightarrow{\eta}\!\!\!\!\rightarrow M$;
(ii) for any $M, N \in \mathsf{Rep}(\Sigma^\mu_\nu)$,

$$M =_\eta N \;\Rightarrow\; \mathbf{F}^\mu_\nu M, \mathbf{F}^\mu_\nu N \xrightarrow{\beta}\!\!\!\!\rightarrow f(P),$$

with $P \xrightarrow{\eta}\!\!\!\!\rightarrow M, N$.

Proof. Let $M \equiv \lambda x_1 \ldots x_n.x_j M_1 \ldots M_m$, where either $m = \mu$ or $m = 0$ and $n \leqslant \nu$. We set

$$\mathbf{F}^\mu_\nu \equiv \lambda x.xA_1 \ldots A_{\nu+1}B_1 \ldots B_\mu C_1 \ldots C_{\mu+\nu},$$

where $A_1, \ldots, A_{\nu+1}, B_1, \ldots, B_\mu, C_1, \ldots, C_{\mu+\nu}$ will be specified later. We have

$$\mathbf{F}^\mu_\nu M \xrightarrow{\beta}\!\!\!\!\rightarrow A_j M'_1 \ldots M'_m A_{n+1} \ldots A_{\nu+1} B_1 \ldots B_\mu C_1 \ldots C_{\mu+\nu},$$

where for $1 \leqslant k \leqslant m$,

$$M'_k = M_k[A_1/x_1, \ldots, A_n/x_n].$$

We set, for $1 \leqslant i \leqslant \mu+\nu$,

$$C_i = \langle C^{(i)}_1, \ldots, C^{(i)}_\nu \rangle \equiv \lambda t.t\, C^{(i)}_1 \ldots C^{(i)}_\nu$$

and, for $1 \leqslant j \leqslant \nu$,

$$A_j = \lambda z_1 \ldots z_{2\mu+\nu+1}.z_{2\mu+\nu+1} \mathbf{U}^\nu_j z_1 \ldots z_{2\mu+\nu+1}, \qquad \text{where } \mathbf{U}^\nu_j \equiv \lambda u_1 \ldots u_\nu.u_j,$$

while

$$A_{\nu+1} = \lambda z_1 \ldots z_{2\mu+\nu+1}.z_{2\mu+\nu+1} \overbrace{\star \ldots \star}^{2\mu+\nu+2},$$

where $\star$ is any term. We then have

$$A_j M'_1 \ldots M'_m A_{n+1} \ldots A_{\nu+1} B_1 \ldots B_\mu C_1 \ldots C_{\mu+\nu}$$

$$\xrightarrow{\beta}\!\!\!\!\rightarrow \begin{cases} C_n \mathbf{U}^\nu_j M'_1 \ldots M'_m A_{n+1} \ldots A_{\nu+1} B_1 \ldots B_\mu C_1 \ldots C_{\mu+\nu} & \text{if } m = \mu, \\ C_{n+\mu} \mathbf{U}^\nu_j A_{n+1} \ldots A_{\nu+1} B_1 \ldots B_\mu C_1 \ldots C_{\mu+\nu} & \text{if } m = 0, \end{cases} \tag{7}$$

$$\xrightarrow{\beta}\!\!\!\!\rightarrow \begin{cases} C^{(n)}_j M'_1 \ldots M'_m A_{n+1} \ldots A_{\nu+1} B_1 \ldots B_\mu C_1 \ldots C_{\mu+\nu} & \text{if } m = \mu, \\ C^{(n+\mu)}_j A_{n+1} \ldots A_{\nu+1} B_1 \ldots B_\mu C_1 \ldots C_{\mu+\nu} & \text{if } m = 0. \end{cases}$$

Now, we set, for $1 \leqslant j \leqslant \nu, 1 \leqslant i \leqslant \nu+\mu$,

$$C_j^{(i)} = G\langle D_j^{(i)}, E_j^{(i)}\rangle,$$

where

$$G \equiv \lambda a t_1 \ldots t_{\mu+1}.a(t_{\mu+1} \overbrace{\star \ldots \star}^{2\mu+\nu} (\mathbf{U}_{2\mu+\nu+4}^{2\mu+\nu+5})\mathbf{U}_2^2)t_1 \ldots t_{\mu+1}.$$

We observe that, for $1 \leqslant i \leqslant \nu+1, 1 \leqslant j \leqslant \mu$, putting

$$B_j \equiv \mathbf{U}_{2\mu+\nu+2}^{2\mu+\nu+2},$$

and with $\mathbf{K} \equiv \mathbf{U}_1^2, \mathbf{O} \equiv \mathbf{U}_2^2$, we have

$$A_i \overbrace{\star \ldots \star}^{2\mu+\nu} (\mathbf{U}_{2\mu+\nu+4}^{2\mu+\nu+5})\mathbf{O} \xrightarrow{\beta}\!\!\!\!\rightarrow \mathbf{K},$$

$$B_j \overbrace{\star \ldots \star}^{2\mu+\nu} (\mathbf{U}_{2\mu+\nu+4}^{2\mu+\nu+5})\mathbf{O} \xrightarrow{\beta}\!\!\!\!\rightarrow \mathbf{O}.$$

Furthermore, recall that $\langle a,b\rangle\mathbf{K} \xrightarrow{\beta}\!\!\!\!\rightarrow a$ and $\langle a,b\rangle\mathbf{O} \xrightarrow{\beta}\!\!\!\!\rightarrow b$. Thus, we have, continuing (7) and using $\bar{M}', \bar{A}, \bar{B}, \bar{C}$ as shorthands for $M_1' \ldots M_m', A_{n+1} \ldots A_{\nu+1}, B_1 \ldots B_\mu$ and $C_1 \ldots C_{\mu+\nu}$, respectively,

$$
\begin{aligned}
& A_j \bar{M}' \bar{A} \bar{B} \bar{C} \\
& \xrightarrow{\beta}\!\!\!\!\rightarrow \begin{cases} G\langle D_j^{(n)}, E_j^{(n)}\rangle \bar{M}' \bar{A} \bar{B} \bar{C} & (m = \mu), \\ G\langle D_j^{(n+\mu)}, E_j^{(n+\mu)}\rangle \bar{A} \bar{B} \bar{C} & (m = 0), \end{cases} \\
& \xrightarrow{\beta}\!\!\!\!\rightarrow \begin{cases} \langle D_j^{(n)}, E_j^{(n)}\rangle (A_{n+1} \overbrace{\star \ldots \star}^{2\mu+\nu} (\mathbf{U}_{2\mu+\nu+4}^{2\mu+\nu+5})\mathbf{O}) \bar{M}' \bar{A} \bar{B} \bar{C} & (m = \mu), \\ \langle D_j^{(n+\mu)}, E_j^{(n+\mu)}\rangle (A_{n+\mu+1} \overbrace{\star \ldots \star}^{2\mu+\nu} (\mathbf{U}_{2\mu+\nu+4}^{2\mu+\nu+5})\mathbf{O}) \bar{M}' \bar{A} \bar{B} \bar{C} & (m = 0, \\ & \nu - n \geqslant \mu), \\ \langle D_j^{(n+\mu)}, E_j^{(n+\mu)}\rangle (B_{\mu-\nu+n} \overbrace{\star \ldots \star}^{2\mu+\nu} (\mathbf{U}_{2\mu+\nu+4}^{2\mu+\nu+5})\mathbf{O}) \bar{M}' \bar{A} \bar{B} \bar{C} & (m = 0, \\ & \nu - n < \mu), \end{cases} \qquad (8) \\
& \xrightarrow{\beta}\!\!\!\!\rightarrow \begin{cases} \langle D_j^{(n)}, E_j^{(n)}\rangle \mathbf{K} \bar{M}' \bar{A} \bar{B} \bar{C} & (m = \mu), \\ \langle D_j^{(n+\mu)}, E_j^{(n+\mu)}\rangle \mathbf{K} \bar{M}' \bar{A} \bar{B} \bar{C} & (m = 0 \wedge \nu - n \geqslant \mu), \\ \langle D_j^{(n+\mu)}, E_j^{(n+\mu)}\rangle \mathbf{O} \bar{M}' \bar{A} \bar{B} \bar{C} & (m = 0 \wedge \nu - n < \mu), \end{cases} \\
& \xrightarrow{\beta}\!\!\!\!\rightarrow \begin{cases} D_j^{(n)} \bar{M}' \bar{A} \bar{B} \bar{C} & (m = \mu), \\ D_j^{(n+\mu)} \bar{A} \bar{B} \bar{C} & (m = 0 \wedge \nu - n \geqslant \mu), \\ E_j^{(n+\mu)} \bar{A} \bar{B} \bar{C} & (m = 0 \wedge \nu - n < \mu), \end{cases}
\end{aligned}
$$

Notice that the first subcase in which case $(m = 0)$ is split is treated as the case $(m = \mu)$. Indeed, if $(m = \mu)$ the head variable has arity μ, while if $m = 0$ and $v - n \geqslant \mu$ the head variable can be η-expanded μ times without exceeding v.

So we can say that $D_j^{(n)}$ (or $D_j^{(n+\mu)}$) comes into the functional position when the head variable has actually arity μ, while $E_j^{(n+\mu)}$ comes into the functional position when the head variable has arity 0. It follows that we can finally set, for $1 \leqslant k \leqslant v, 1 \leqslant j \leqslant k$,

$$D_j^{(k)} \equiv \lambda w_1 \ldots w_{3\mu+2v-k+1}.p_j^{\mu,k}(w_1 w_{\mu+1} \ldots w_{3\mu+2v-k+1}, \ldots, w_\mu w_{\mu+1} \ldots w_{3\mu+2v-k+1})$$

and, for $\mu+1 \leqslant k \leqslant \mu+v, 1 \leqslant j \leqslant k$,

$$E_j^{(k)} \equiv \lambda w_1 \ldots w_{3\mu+2v-k+1}.p_j^{0,k-\mu}.$$

Continuing (8), we finally obtain

$$A_j M_1' \ldots M_m' A_{n+1} \ldots A_{v+1} B_1 \ldots B_\mu C_1 \ldots C_{\mu+v}$$
$$\xrightarrow{\beta}\!\!\!\!\rightarrow \begin{cases} p_j^{\mu,n}(M_1'\bar{A}\bar{B}\bar{C}, \ldots, M_m'\bar{A}\bar{B}\bar{C}) & (m = \mu), \\ p_j^{\mu,n+\mu}(A_{n+1}\bar{A}\bar{B}\bar{C}, \ldots, A_{n+\mu}\bar{A}\bar{B}\bar{C}) & (m = 0 \wedge v - n \geqslant \mu), \\ p_j^{0,n} & (m = 0 \wedge v - n < \mu). \end{cases} \tag{9}$$

But

$$p_j^{\mu,n}(M_1'\bar{A}\bar{B}\bar{C}, \ldots, M_m'\bar{A}\bar{B}\bar{C}) = p_j^{\mu,n}(\mathbf{F}_v^\mu \lambda x_1 \ldots x_n.M_1, \ldots, \mathbf{F}_v^\mu \lambda x_1 \ldots x_n.M_m)$$

and

$$p_j^{\mu,n+\mu}(A_{n+1}\bar{A}\bar{B}\bar{C}, \ldots, A_{n+\mu}\bar{A}\bar{B}\bar{C})$$
$$= p_j^{\mu,n+\mu}(\mathbf{F}_v^\mu \lambda x_1 \ldots x_{n+\mu}.x_{n+1}, \ldots, \mathbf{F}_v^\mu \lambda x_1 \ldots x_{n+\mu}.x_{n+\mu}),$$

so that (9) shows exactly what is needed to prove both *(i)* and *(ii)* by induction on the depth of the Böhm tree of M. □

Acknowledgements

I am grateful to Corrado Böhm for helpful discussions about the topics of this paper, and to the referees for their suggestions.

References

[1] H. Barendregt, The Lambda Calculus; Its Syntax and Semantics (revised edition). North-Holland, Amsterdam, 1984.

[2] A. Berarducci, C. Böhm, A self-interpreter of lambda calculus having a normal form, in: E. Börger, G. Jäger, H. Kleine Büning, S. Martini, M.M. Richter (Eds.), Proceedings Computer Science Logic, San Miniato, 1992, Lecture Notes in Computer Science, vol. 702, pp. 85–99.

[3] C. Böhm, Alcune proprietà delle forme β–η-normali nel λ–κ-calcolo, Pubblicazioni dell'Istituto per le Applicazioni del Calcolo 696 (Roma, 1968), 19.

[4] C. Böhm, Functional Programming and Combinatory Algebras, in: M.P. Chytil, L. Januga, V. Koubek (Eds.), Proc. Mathematical Foundations of Computer Science, (Carlsbad, 1988), Lecture Notes in Computer Science, vol. 324, pp. 14–26.
[5] C. Böhm, Fixed point equations inside the algebra of normal forms, Fundamenta Informaticae, to appear.
[6] C. Böhm, M. Dezani-Ciancaglini, Combinatorial problems,combinator equations and normal forms, in: J. Loeckx (Ed.), Proceedings Automata, Languages and Programming 2th. Colloquium, (Saarbrücken, 1974), Lecture Notes in Computer Science, vol. 14, pp. 185-199.
[7] C. Böhm, M. Dezani-Ciancaglini, P. Peretti, S. Ronchi della Rocca, A discrimination algorithm inside λ–β-calculus, Theoret. Comput. Sci. 8 (1979) 271-291.
[8] C. Böhm, A. Piperno, S. Guerrini, λ-definition of function(al)s by normal norms, in: D. Sannella (Ed.), Proc. European Symp. on Programming, Edinburgh, 1994, Lecture Notes in Computer Science, vol. 788, pp. 135-149.
[9] M. Coppo, M. Dezani-Ciancaglini, S. Ronchi della Rocca, (Semi)-separability of finite sets of terms in Scott's D_∞ models of the λ-calculus, in: G. Ausiello, C. Böhm (Eds.), Proceedings Automata, Languages and Programming 5th. Colloquium, Udine, 1978, Lecture Notes in Computer Science, vol. 62, pp. 142-164.
[10] G. Huet, An analysis of Böhm's theorem, Theoret. Comput. Sci. 121 (1993) 145-167.
[11] J.L. Krivine, Lambda-Calcul: Types et Modèles, Masson, 1990.
[12] G.F. Mascari, M. Pedicini, Head linear reduction and pure proof net extraction, Theoret. Comput. Sci. 135 (1994) 111-137.
[13] A. Piperno, E. Tronci, Regular systems of equations in the λ-calculus, Internat. J. Found. Comput. Sci. 1(3) (1994) 325-339.