International Journal of Foundations of Computer Science Vol. 1 No. 3 (1990) 325-339
©World Scientific Publishing Company

REGULAR SYSTEMS OF EQUATIONS IN λ-CALCULUS

ADOLFO PIPERNO
Dipartimento di Matematica, Istituto "G.Castelnuovo"
Università degli Studi di Roma "La Sapienza"
Piazzale Aldo Moro 5, I-00185 Rome, Italy
and
ENRICO TRONCI
Department of Mathematics, Carnegie Mellon University
Pittsburgh, PA 15213, USA

Received 15 December 1989
Revised 15 November 1990

ABSTRACT

Many problems arising in equational theories like Lambda-calculus and Combinatory Logic can be expressed by combinatory equations or systems of equations. However, the solvability problem for an arbitrarily given class of systems is in general undecidable. In this paper we shall focus our attention on a decidable class of systems, which will be called *regular systems*, and we shall analyse some classical problems and well-known properties of Lambda-calculus that can be described and solved by means of regular systems. The significance of such class will be emphasized showing that for slight extensions of it the solvability problem turns out to be undecidable.

Keywords: Combinatory equations and systems of equations; (X)-separability, invertibility and strong normalization in λ-calculus; Fixed point of combinators; Numeral sytems; Functional programming.

0. Introduction

Many problems arising in equational theories like λ-calculus and Combinatory Logic can be expressed by equations or systems of equations. This motivates the study of classes of combinatory equation systems for which it is possible to provide uniform methods of solution, and the study of classes of systems whose solvability can be proved to be undecidable.

Furthermore, from a different standpoint, a system of equations can be viewed as the specification of a problem in a declarative language, the system's solution being an executable program satisfying the equations, as e.g. in Ref. 1. From such a perspective, since λ-calculus can be considered as the prototype of any (higher-order) functional programming language, it seems natural to consider the study of combinatory equations and systems of equations as a theoretical foundation for a kind of synthesis of functional programs and machines.

As an example, we consider the problem of constructing a functional machine $\mathcal{M}$ s.t.:

(i) $\mathcal{M}$ has a single instruction F;

(ii) Any recursive function can be programmed in $\mathcal{M}$, i.e. any recursive function can be expressed as an applicative combination of F's;

(iii) The instruction F itself implements a particular given recursive function;

(iv) Some given programs $A_1,\dots,A_n$ can be "easily" expressed using F.

Table A. Solvability of equations and systems of equations.

$\exists X\ MX =_T NX$	UNDECIDABLE for semisensible T	(see 1.1)
$\exists X\ MX =_T y\ (y \notin M)$:		(see 1.2)
• $M =_\beta \lambda x.xN_1 \ldots N_m$	ALWAYS SOLVABLE for semisensible T	
• $M =_\beta \lambda xt_1 \ldots t_h.xN_1 \ldots N_m$	OPEN for extensional T, NEVER SOLVABLE otherwise	
• $M =_\beta \lambda xt_1 \ldots t_h.\xi N_1 \ldots N_m\ (\xi \neq x)$	NEVER SOLVABLE for semisensible T	
$\exists X\ XM_i =_T y_i\ (i \in \{1,\ldots,t\}$, M_i closed, $y_i \notin M)$	CHARACTERIZED [a]	(see 1.3)
$\exists X\ XM_i =_T y_i\ (i \in \{1,\ldots,t\}, y_i \notin M_i)$	CHARACTERIZED [b]	(see 1.4)
Regular Systems:		
$\exists X_1,\ldots,X_n\ M_iX_1 \ldots X_n =_T y_i\ (i \in \{1,\ldots,t\}, y_i \notin M_i)$	OPEN for extensional T, when M has more than n initial λ's CHARACTERIZED for a suitable distribution of right hand sides and UNDECIDABLE otherwise, when M has at most n initial λ's	(see §2.1)
$\exists X_1,\ldots,X_n\ M_iX_1 \ldots X_n =_T y_iX_1 \ldots X_nP_{i,1} \ldots P_{i,r}\ (i \in \{1,\ldots,t\})$, where for $k=1,\ldots,r$: $P_{i,k}$ is a suitable subterm of M_i	As above	

[a] Böhm's theorem[6] in $\beta\eta$-calculus, separability[7] in $\beta\eta\Omega$-calculus

[b] Reducible to the separability problem.

Such a problem can be formulated by the following system, in the unknown F:

$$\begin{cases} F\ \mathbf{zero} = y_0 & \text{Eq 0} \\ F\ (\mathbf{succ}\ z) = y_1\, z\, F & \text{Eq 1} \\ F\,F = A_1 & \text{Eq 2} \\ F\,A_1 = A_2 & \text{Eq 3} \\ \ldots & \ldots \\ F\,A_{n-1} = A_n & \text{Eq n+1} \\ F\,A_n = \mathbf{K} & \text{Eq n+2} \\ F\,\mathbf{K} = \mathbf{S} & \text{Eq n+3} \end{cases} \qquad (0.1)$$

where **zero** and **succ** are respectively the zero and the successor function of some adequate numeral system (Ref.2, §6.4.1). In effect, we have:

- (i) is obviously satisfied;
- (ii) is enforced by the eqs n+2, n+3;
- (iii) is enforced by the eqs 0, 1;
- (iv) is enforced by the eqs 2, 3, ...,n+1.

Table A summarizes the *state of the art* in the study of combinatory equations. In Section 1 we shall illustrate the contents of that table and we shall prove the undecidability of some classes of equations and systems. We shall then introduce (Section 2) a class of systems, which will be called *regular systems*, as a natural extension of the decidable classes already appearing in the literature. The significance of the class of regular systems shall be emphasized showing how it is difficult to generalize it without running into undecidable problems (Section 4).

Furthermore, we will show (Section 3) how problems and properties of λ -calculus can be described and solved by means of regular systems; for instance, system (0.1), under certain assumptions over left-hand side members of the equations, turns out to be regular,

but we shall also examine more classical properties of λ -calculus, concerning e.g. invertibility of terms, numeral systems and fixed point combinators.

We assume the reader is familiar with the basic notions of λ-calculus not explicitly cited in this paper; for a complete treatment of them, see e.g. Refs. 2 and 5.

We shall adopt the following notation: $=$ $(=_\beta, =_{\beta\eta})$ denotes convertibility, $\equiv$ denotes identity of objects; FV(T) denotes the set of free variables of a λ -term T; we define

- $\mathrm{HNF} = \{\lambda z_1 \ldots z_n.\xi T_1 \ldots T_m \mid (\xi \text{ is a variable}) \wedge (T_1,\ldots,T_m \in \Lambda, n,m \geq 0)\}$,
- $\mathrm{SOL} = \{M \in \Lambda \mid \exists N \in \mathrm{HNF} \text{ s.t. } M =_\beta N\}$.

If $T = \lambda z_1 \ldots z_n.\xi T_1 \ldots T_m \in \mathrm{SOL}$, we define the *order*, *degree*, *head* and the *Böhm tree* of T to be respectively:

- $\mathrm{ord}(T) = n$,
- $\deg(T) = m$,
- $\mathrm{head}(T) = \xi$
- $\mathrm{BT}(T) = \lambda z_1 \ldots z_n.\xi$ with subtrees $\mathrm{BT}(T_1) \ldots \mathrm{BT}(T_m)$

otherwise, if $T \notin \mathrm{SOL}$: $\mathrm{BT}(T) = \Omega$ (head(T), ord(T) and deg(T) are not defined here).

Roughly speaking, we say that a λ-theory T is *semisensible* (sms) if it never equates two terms A and B, where $A \in \mathrm{SOL}$ and $B \notin \mathrm{SOL}$.

A *path* $\gamma = \langle a_1,\ldots,a_k \rangle$ $(k \geq 0)$ starting from the root of BT(T) is a (possibly empty) finite sequence of positive integers uniquely identifying a (poss.non-proper) subtree of BT(T); roughly speaking, a path γ is *defined* if it does not meet any Ω; we will then write $\gamma \in \mathrm{BT}(T)$ and we will denote by T_γ the subterm of T whose Böhm tree is identified by γ .

Systems of combinatory equations: some definitions

A *system of equations* in λ-calculus will be considered as a pair $\mathbf{S} = (\Gamma, \mathcal{X})$, where Γ is a finite set of formulas of Λ (the equations) and $\mathcal{X} \equiv \{x_1,\ldots,x_n\}$ is a finite set of variables of Λ (the unknowns).

We will denote by $\mathcal{L}_\Gamma$ $(\mathcal{R}_\Gamma)$ the set of left-hand side (lhs) (right-hand side (rhs)) members of the equations in Γ .

A solution for the system $\mathbf{S}$ in the λ-theory T is a set $\mathcal{V} \equiv \{V_1,\ldots,V_n\}$ of terms (with $FV(\mathcal{V}) \cap FV(\mathcal{L}_\Gamma) = \varnothing$) such that the substitution $[V_1/x_1,\ldots,V_n/x_n]\,\Gamma$ makes the equations of $\mathbf{S}$ theorems in T. For simplicity's sake we will denote by the pair $\mathcal{E} = (U{=}V, \mathcal{X})$ a single combinatory equation.

From now onwards the contents of Table A shall be rephrased according to the previous definitions, e.g. the first line of the table will be expressed asking for the solvability of $\mathcal{E} = (M\,x = N\,x, \{x\})$, where $x \notin M,N$. Though this might appear as a mere change of notation, it will allow us to simplify the description of important properties of systems.

1. Decidable Combinatory Equations and Systems

In this section we will analyse equations and systems of equations, whose solvability has been studied in the literature, and we will emphasize the relationships existing among them. Moreover, we will identify some classes of systems whose solvability we will prove (Section 4) to be undecidable and some problems which are still open.

1.1. Combinatory Equations

Being not restricted to consider a single unknown, the most general shape for combinatory equations is:

$\mathcal{E} = (Mx = Nx, \{x\})$, where $x \notin M,N$. (1.1)

Statman[8] and Dezani[9] respectively proved the undecidability of the equations, belonging to that class (see also Ref.10): $\mathcal{E}_1 = (Mx = \mathbf{S}, \{x\})$; $\mathcal{E}_2 = (Mx = \mathbf{I}, \{x\})$.

In Section 4 we will extend their proofs to the most general case.

These results force the assumption of stronger conditions for rhs members of equations. Indeed, one may start asking for the solvability of the equation

$\mathcal{E} = (Mx = y, \{x\})$, (1.2)

where $x \notin M$ and y is a variable not occurring in M. In other words, (1.2) is equivalent to asking whether, given M, for every term T there exists a term X_T such that

$M X_T = T$. (1.2)*

To be more explicit: (1.2) is a special case of (1.2)* with $T \equiv y$ and if X is a solution of (1.2) then, for every T, $X_T = (\lambda y.X)T$ is a solution of (1.2)*.

The solvability of (1.2) corresponds to the existence of a right inverse for M, i.e. a term M^R s.t. $\mathbf{B}\, M\, M^R = \mathbf{I}$, since if X is s.t. MX = y, then $M^R = \lambda y.X$ and vice versa, if M^R is s.t. $\mathbf{B}\, M\, M^R = \mathbf{I}$, then $X = M^R y$ is s.t. MX = y. Such a problem (together with the existence of a left inverse for M, i.e. a term M^L s.t. $\mathbf{B}\, M^L\, M = \mathbf{I}$) has been characterized for λ-β-calculus in Ref. 11 (see also Ref. 12). More precisely,

the equation (1.2) is β-solvable ⇔ for some $N_1,\ldots,N_m$, $Mx =_\beta x N_1 \ldots N_m$.

The right/left invertibility problem in λ–βη-calculus has been studied in Ref. 13, where sufficient conditions for its solvability have been given, while full invertibility in λ–βη-calculus has been characterized in Ref. 14.

1.2. Combinatory Equations in Extensional Theories: A Remark

The characterization of the solvability of (1.2) can be easily extended to any non-extensional semisensible (sms) theory, but it is still an open problem in the presence of extensionality in the case where Mx does not reduce to a λ-free term. For such a reason, while considering systems, we will assume the lhs members of the equations to be λ-free. With this restriction, we will obtain characterizations which hold in *any* sms theory. The corresponding unrestricted problems, whenever the shape of the considered systems allows their formulation, do not admit any solution in non-extensional theories, but must be considered still open in the presence of extensionality.

1.3. Systems of Equations

The first equation systems for which a method of solution was given have been studied in Böhm's theorem (Ref.6 for t = 2, Ref.15 for t ≥ 2):

$\mathbf{S} = (\Gamma,\{x\})$, with $\Gamma = \{x\mathbf{M}_i = y_i \mid i \in \{1,\ldots,t\}\}$, (1.3)

where the $\mathbf{M}_i$'s are closed βη-normal forms and the y_i's are arbitrary variables.

More precisely, it was proved that (1.3) is solvable iff, modulo α-conversion, for i,k=1,...,t:

$\mathbf{M}_i =_{\beta\eta} \mathbf{M}_k \Rightarrow y_i \equiv y_k$.

It comes out that (1.3) can be solved, whenever possible, substituting a Church s-tuple $(\lambda a.a\, A_1 \ldots A_s)$ of terms for the unknown x, with s suitably large. In Ref. 16 it has been proved that it is sufficient to bound s to μ+1 where μ is the maximum of the orders of the $\mathbf{M}_i$'s.

It is important to note that it is the hypothesis for the $\mathbf{M}_i$'s to be closed normal forms that allows stating Böhm's theorem in terms of βη-convertibility. This is no more sufficient in the general case, as shown by the following examples:

- $\mathbf{S}_1 = (\Gamma,\{x\})$, with $\Gamma = \{xz = y_1, x\mathbf{I} = y_2\}$, not solvable while $z \neq_{\beta\eta} \mathbf{I}$;
- $\mathbf{S}_2 = (\Gamma,\{x\})$, with $\Gamma = \{x(\omega\,\omega) = y_1, x\,\mathbf{I} = y_2\}$, not solvable while $\omega\,\omega \neq_{\beta\eta} \mathbf{I}$ $(\omega \equiv \lambda x.xx)$.

The following notion will enable us to consider the solvability of the system

$\mathbf{S} = (\Gamma,\{x\})$, with $\Gamma = \{x M_i = y_i \mid i \in \{1,\ldots,t\}\}$, (1.4)

where the M_i's are arbitrary terms not containing the variable x.

1.3.1. $\mathcal{F}$-indistinctness

The following definitions are needed to introduce the notion of $\mathcal{F}$-indistinctness:

- Let $T_1,T_2 \in$ HNF and γ be a path ($\gamma \in BT(T_i)$, i=1,2) :
  equivalence between terms[6]: $T_1 \sim T_2$ iff $\text{head}(T_1) = \text{head}(T_2)$ ∧ $\text{deg}(T_1) - \text{ord}(T_1) = \text{deg}(T_2) - \text{ord}(T_2)$.
  γ-equivalence between terms[7]: $T_1 \sim_\gamma T_2$ iff $(T_1)_\gamma \sim (T_2)_\gamma$.
- Let $\mathcal{F} \subset \Lambda$ be a finite set of terms, for every $F \in \mathcal{F}, \gamma \in BT(F)$; we say that[7] γ is *useful* for $\mathcal{F}$ iff $(\forall F \in \mathcal{F}, F_\gamma \neq \Omega) \wedge (\exists F_1,F_2 \in \mathcal{F}$ s.t. $F_1 \not\sim_\gamma F_2) \wedge (\forall \alpha < \gamma, \forall M,N \in \mathcal{F}: M \sim_\alpha N)$.

Definition 1. ($\mathcal{F}$-indistinctness, introduced in Refs. 17, 18, see also Ref. 10)

Let $\mathcal{F} \subset \Lambda$ and $M,N \in \mathcal{F}$; we define the relation $\simeq_{\mathcal{F}} \subseteq \mathcal{F} \times \mathcal{F}$, as follows:

$M \simeq_{\mathcal{F}} N \Leftrightarrow \exists \mathcal{P} \subseteq \mathcal{F}$ s.t. $(\{M,N\} \subseteq \mathcal{P} \wedge \neg(\exists\, \alpha$ useful for $\mathcal{P}))$. ❑

It comes out that the solvability of the system (1.4) can be reduced to the so-called *separability* problem, which was characterized in Ref. 7, where the relation of $\mathcal{F}$-indistinctness between terms, which was implicitly given, played a role similar to that of βη-convertibility in Böhm's theorem.

More precisely: Let $\mathcal{M} = \{\mathbf{M}_1,\ldots,\mathbf{M}_t\}$ be a set of closed terms; the system

$\mathbf{S} = (\Gamma,\{x\})$, with $\Gamma = \{x\mathbf{M}_i = y_i \mid i \in \{1,\ldots,t\}\}$, (1.4)*

where the y_i's are arbitrary variables, is βη-solvable iff, for i,k=1,...,t:

$\mathbf{M}_i \simeq_{\mathcal{M}} \mathbf{M}_k \Rightarrow y_i \equiv y_k$. (1.5)

The problem of solvability of (1.4) can then be easily reduced to that of (1.4)* substituting **Ω** for every free variable in the M_i's , thus obtaining a system with the shape of (1.4)*.

2. From Self-Application To Regular Systems

An important (and meanwhile restrictive) feature of systems having the shape (1.3,4) is the absence of self-application, the unknown being required to occur exactly once in every equation.

To fill this gap, we shall consider systems in which the unknowns are allowed to appear any number of times in left-hand side members of the equations.

It comes out that the notion of indistinctness is no more sufficient to characterize the solvability of systems with self-applicative features, as shown by the following example:

- $\mathbf{S} = (\Gamma, \{x\})$, with $\Gamma = \{ x(x\,\mathbf{I}) = y_1 , x\,\mathbf{I} = y_2 \}$,

which is not solvable while $x(x\,\mathbf{I}) \not\approx_{\{x(x\mathbf{I}), x\mathbf{I}\}} x\,\mathbf{I}$.

In order to characterize the solvability of systems in which self-application of unknowns appears, we introduce the notion of *left-regularity* for a system of combinatory equations.

Definition 2. (effective path)

A path γ is said *effective* for T ($\in$ HNF) iff $\gamma \equiv \langle a_1,\ldots,a_k \rangle$ $(k > 0)$ $\wedge$ $\deg(T) \geq a_1$. ❑

In self-applicative systems, in order to discriminate external occurrences of the unknowns from internal ones, we shall isolate the set of proper subterms of lhs's of the equations whose head is an unknown.

Definition 3. (critical subterms)

Given a system $\mathbf{S} = (\Gamma, X)$, we define the set $\mathcal{CS}(\mathbf{S})$ of *critical subterms* of $\mathbf{S}$:

$\mathcal{CS}(\mathbf{S}) = \{ M_\alpha \mid M \in \mathcal{L}_\Gamma \wedge \alpha \in BT(M) \wedge \text{head}(M_\alpha) \in X \}$. ❑

Free variables not belonging to the set of unknowns are not involved in any substitution, hence, without loss of generality, they will be considered as undefined objects.

Definition 4.

Given $\mathbf{S} = (\Gamma, X)$, let $Y = FV(\mathcal{L}_\Gamma) - X$. We define $\mathbf{S}_\Omega = (\Gamma_\Omega, X)$, where Γ_Ω is obtained from Γ substituting Ω for the elements of Y. ❑

We are now able to introduce the notion of left-regularity, which is the central issue in characterizing the solvability of self-applicative systems, requiring that external and internal occurrences of the unknown be discriminable; roughly speaking, a subterm of a lhs of any equation must not "collapse" with a lhs.

Definition 5. (left-regularity)

Let $\mathbf{S} = (\Gamma, X)$:

- We first define a new equivalence relation $\approx_{\mathbf{S}} \subset (\mathcal{L}_\Gamma \cup \mathcal{CS}(\mathbf{S})) \times (\mathcal{L}_\Gamma \cup \mathcal{CS}(\mathbf{S}))$:
  $U \approx_{\mathbf{S}} V \Leftrightarrow \text{head}(U) \equiv \text{head}(V) \wedge \exists \mathcal{P}_1 (\neq \emptyset) \subseteq \mathcal{L}_\Gamma, \mathcal{P}_2 \subseteq \mathcal{CS}(\mathbf{S})$ s.t.:
  $\{U, V\} \subseteq \mathcal{P}_1 \cup \mathcal{P}_2 \wedge \neg(\exists\, \alpha$ useful for $\mathcal{P}_1 \cup \mathcal{P}_2$ and effective for $\mathcal{P}_1)$.
- We say that $\mathbf{S}$ is *left-regular* iff the following conditions are both satisfied:
  (i) $\neg(\exists L \in \mathcal{L}_\Gamma, N \in \mathcal{CS}(\mathbf{S})$ s.t. $L \approx_{\mathbf{S}} N)$;
  (ii) $\forall U, V \in \mathcal{L}_\Gamma,\ U \approx_{\mathbf{S}} V \Rightarrow \deg(U) = \deg(V)$. ❑

Example 1.

- The system $\mathbf{S} = (\Gamma, \{x\})$ with $\Gamma = \{x(x\,\Omega) = y_1, x\,x = \mathbf{K}\}$ is not left-regular since $x\,\Omega \approx_{\mathbf{S}} x\,x$.
- The system $\mathbf{S} = (\{x(x\,\mathbf{K}) = x, x\,x = x\,\mathbf{B}\}, \{x\})$ is left-regular. ❑

We can "throw away unnecessary information" from the system $\mathbf{S}$ considering one of its approximations $\mathbf{S}'$. If $\mathbf{S}'$ is solvable then also $\mathbf{S}$ is solvable.

Definition 6.

- Let $U, V \in \Lambda$; we say that U *approximates* V (we write $U \sqsubseteq V$) if $BT(U) \subseteq BT(V)$;
- Let $M \in \Lambda$; we define $\text{Approx}(M) = \{N \in \Lambda \mid N \sqsubseteq M\}$;
- Let $\mathbf{S} = (\Gamma, X)$ be a system of combinatory equations; we define
  $\text{Approx}(\mathbf{S}) = \{\mathbf{S}' \mid \mathbf{S}' = (\Gamma', X) \wedge \Gamma' = \{L' = R \mid L = R \in \Gamma \wedge M' \in \text{Approx}(M)\}\}$. ❑

The following theorem characterizes the solvability of a class of systems exhibiting self-application of unknowns; the solvability of such systems has been studied in Refs. 17, 18, 19 and then characterized in Ref. 20 in the special case where rhs are pairwise distinct – the *X-separability* problem – and in Ref. 10 in the general case.

Theorem 1.

Let T be an sms theory and $\mathbf{S} = (\Gamma, X)$ a system such that:

(i) Any equation of Γ has the shape
$x N_1 \ldots N_m = y$, where $x \in X$ and $y \notin (X \cup FV(\mathcal{L}_\Gamma))$;

(ii) For every pair of equations $L_1 = R_1$, $L_2 = R_2$ in $\mathbf{S}_\Omega$:
$\text{head}(R_1) \equiv \text{head}(R_2) \Rightarrow L_1$ and L_2 are $\mathcal{L}_{\Gamma_\Omega}$-indistinct.

Then $\mathbf{S}$ is solvable in the theory T iff there exists $\mathbf{S}' = (\Gamma', X) \in \text{Approx}(\mathbf{S}_\Omega)$ such that:

- $\mathbf{S}'$ is left-regular;
- for every pair of equations $L_1 = R_1$, $L_2 = R_2$ in $\mathbf{S}'$:
  L_1 and L_2 are $\mathcal{L}_{\Gamma'}$-indistinct $\Rightarrow$ $\text{head}(R_1) \equiv \text{head}(R_2)$.

Proof. The most important issues in proving the theorem shall be discussed in the next subsection. For a more detailed proof see Ref. 10. ❑

2.1. From the Proof of Theorem 1 to Regular Systems: Some Remarks

2.1.1. Building up the solution for a self-applicative system

In order to characterize in a constructive way the solvability of a self-applicative system $\mathbf{S}$, we must define a method which allows us to keep control of the consequences of self-application, since multiple occurrences of the unknowns, appearing at different depths in the Böhm trees of lhs members of the equations, *must be substituted by the same terms*. Indeed, the left-regularity condition enables us to take into account the effects of the cited substitution over all the subterms of lhs members of the equations which are involved in it.

To be more pragmatic, the central issue in defining a method for solving equation systems lies in the shape of the the term constituting the solution to the problem itself. In our approach, the solution of a self-applicative system will be constructed in the following way: given

$\mathcal{S}' = (\Gamma', \{x_1,\ldots,x_n\})$, substitute for the unknown x_1 the term:

$$X_1 = \lambda a_1\ldots a_e.a_e z_1\ldots z_k a_1\ldots a_e, \qquad (2.1)$$

with e,k suitably large and $z_1,\ldots,z_k$ fresh variables, thus obtaining a new system in the unknowns $x_2,\ldots,x_n,z_1,\ldots,z_k$:

$$\mathcal{S}'_{(i)} = (\Gamma'[X_1/x_1], \{x_2,\ldots,x_n,z_1,\ldots,z_k\})$$

Under the condition of left-regularity, the iterated application of such substitutions leads to a system such that the distribution of the head variables of lhs members of the equations reflects that of rhs members. By the assignment of suitable terms to such variables we then obtain the result.

Note that terms having shape (2.1) have an algebraic interest[21] and the following important properties:

(i) they preserve left-regularity of systems;

(ii) they preserve the strong normalization property when substituted into normal forms.

While (i) is an essential feature for proving the correctness of our method, (ii) seems to indicate a very interesting direction for future work in the study the relationships among typed λ-calculus, combinatory algebras and solvability of combinatorial problems, as in Ref. 22.

2.1.2. The need for approximants

Note that, whenever a system $\mathcal{S} = (\Gamma, X)$ has a solution and there is an occurrence of Ω in the lhs member of an equation, the system $\mathcal{S}^V = (\Gamma^V, X)$ obtained replacing a term V for that Ω-occurrence is still solvable, even if it may happen that $\mathcal{S}^V$ is no more left-regular; for such a reason it is sufficient for the solvability of a system that the conditions of Theorem 1 hold for an approximant of the system itself. From here onwards, without loss of generality, we shall assume that such approximations have been performed in advance.

2.1.3. About condition (ii) in Theorem 1: an undecidable problem

As a consequence of the presence of self-application, it is important to point out that if we drop the hypothesis (ii) in Theorem 1, thus allowing an arbitrary distribution for right hand sides of the equations (as in (1.3) and (1.4)), the solvability problem for the class of systems so obtained turns out to be undecidable, as proved in Section 4.

2.1.4. Introducing regular systems

We first note that every system without self-application is trivially left-regular. Furthermore, comparing the condition of solvability for the class of self-applicative systems characterized in Theorem 1 with the condition (1.5) of solvability for a non-self-applicative system, we have that the essential new feature in the former is constituted by the condition of left-regularity. It follows that, whenever we assume a self-applicative system belonging to that class to be *a priori* left-regular, we can characterize its solvability exactly in the same way as we did for systems without self-application. Following this idea, we are led to introduce the notion of *regular system*, which enables us to extend in a considerable way the class of systems whose solvability is characterized in Theor.1.

2.2. *Regular Systems*

Definition 7.

Let $\mathcal{S} = (\Gamma, X)$ a system such that $\mathcal{S}_\Omega$ is left-regular, $\Gamma = \Gamma_1 \cup \Gamma_2$ and $X = \{x_1,\ldots,x_n\}$.
$\mathcal{S}$ is said to be *regular* if

(i) Any equation of Γ_1 has one of the following shapes
- $x N_1\ldots N_m = y N_1\ldots N_m$
- $x N_1\ldots N_m = y\, x_1\ldots x_n z_1\ldots z_r$

where $x \in X, y \notin X \cup FV(\mathcal{L}_\Gamma)$ and, for j=1,...,r, $z_j = (x N_1\ldots N_m)\alpha_j$ where α_j is an effective path for $x N_1\ldots N_m$ s.t. the variables appearing along it do not belong to $FV(x N_1\ldots N_m) - X$ (see Ref.2, §10.3, Böhm-out technique);

(ii) For every pair of equations $L_1 = R_1$, $L_2 = R_2$ in Γ_1: $head(R_1) \neq head(R_2)$;

(iii) Any equation of Γ_2 has the shape
- $x N_1\ldots N_m = N$

where $x \in X$ and N is a βη-normal form with $FV(N) \cap (X \cup FV(\mathcal{L}_\Gamma)) = \varnothing$;

(iv) For every pair of equations $L_1 = R_1 \in \Gamma_1$, $L_2 = R_2 \in \Gamma_2$:
L_1 and L_2 are $\mathcal{L}_{\Gamma_\Omega}$-indistinct $\Rightarrow$ $deg(R_1) \neq deg(R_2) - ord(R_2)$;

(v) For every pair of equations $L_1 = R_1$, $L_2 = R_2$ in Γ_Ω:
R_1 and R_2 are $\mathcal{R}_{\Gamma_\Omega}$-indistinct $\Rightarrow$ L_1 and L_2 are $\mathcal{L}_{\Gamma_\Omega}$-indistinct . ❑

Example 2.

The following system is regular

$$\mathcal{S} = (\{\ f(\lambda ab.b) = y_0,\ f(\lambda b.bzb) = y_1 f z,\ ff = \mathbf{K},\ f\mathbf{K} = \mathbf{S},\ f\mathbf{S} = y_2\}, \{f\})$$

On the other hand, the following systems are not regular, since in $\mathcal{S}_1$ the first equation does not satisfy condition (iii) of Definition 7, while $\mathcal{S}_2$ is not left-regular since it does not satisfy condition (ii) of Definition 5:

- $\mathcal{S}_1 = (\{\ ff = f\mathbf{B}, f\,\mathbf{K} = \mathbf{I}\ , \{f\}\ )$,
- $\mathcal{S}_2 = (\{\ fgf = \mathbf{B}, f\,g = \mathbf{I}\ , \{f\}\ )$. ❑

Theorem 2. (Main theorem)

Let T be an sms theory and $\mathcal{S} = (\Gamma, X)$ a regular system.
Then $\mathcal{S}$ is solvable in the theory T iff
for every pair of equations $L_1 = R_1$, $L_2 = R_2$ in $\mathcal{S}_\Omega$:
L_1 and L_2 are $\mathcal{L}_{\Gamma_\Omega}$-indistinct $\Rightarrow$ R_1 and R_2 are $\mathcal{R}_{\Gamma_\Omega}$-indistinct .

Sketch of the proof:

(⇒): From Theorem 1.

(⇐): As in Ref. 17, given the fresh variables $u_1,\ldots,u_t$, substitute for the variable x_i the term $u_i\, u_1\ldots u_t$, thus obtaining a new system in the unknowns $u_1,\ldots,u_t$. Then, use theor.1, Böhm-out technique and make the suitable substitutions. ❑

Note that for regular systems the conditions for solvability with fresh variables in rhs members of the equations or with combinators are the same. This is not the case for non-regular systems.

Moreover, by means of equations having the shape $x\ N_1 \dots N_m = y\ x_1 \dots x_n\ z_1 \dots z_r$, we are able to Böhm-out the variables $z_1, \dots, z_r$ as well as any term we substitute for them in $xN_1 \dots N_m$.

By means of the notion of regularity it is possible to give partial characterizations for the invertibility problem in extensional theories (see also Ref. 13).

Theorem 3. (left-invertibility)

Let T be an extensional sms theory and $M \equiv \lambda y\, x_1 \dots x_n . y\ M_1 \dots M_t$, where, for $i=1,\dots,t$, M_i is λ-free and $\mathrm{head}(M_i) \in X = \{x_1, \dots, x_n\}$.

Consider the system $\mathbf{S} = (\{M_i = y_i \mid i \in \{1,\dots,t\}\}, X)$ where the y_i's are pairwise distinct. Then:

M is T-left-invertible iff

there exists a left-regular approximation of $\mathbf{S}_\Omega$ whose lhs are distinct.

Proof. From Ref. 10, §2.7, Ref. 17, §9.1 and Theorem 2. ❑

Theorem 4. (right-invertibility)

Let T be an extensional sms theory and let $M \equiv \lambda x\, z_1 \dots z_r . x\, M_1 \dots M_m$ be s.t. the z_i's occur only as leaves in $BT(x\, M_1 \dots M_m)$ and, denoting $\mathcal{E} = (x\, M_1 \dots M_m = y, \{x\})$, the equation $\mathcal{E}_\Omega$ is left-regular. Then:

M is T-right-invertible iff

for $i=1,\dots,r$, $z_i \in FV(BT([\Omega/(FV(xM_1 \dots M_m) - \{x, z_1, \dots, z_r\})]\,(xM_1 \dots M_m)))$.

Proof. ($\Rightarrow$): trivial; ($\Leftarrow$): From theorem 2. ❑

3. Solving Problems By Means Of Regular Systems

We will now exhibit some examples about problems which can be solved by means of regular systems.

3.1. A Numeral System

Let $\mathbf{O} \equiv \lambda xy.y$, $\mathbf{W} \equiv \lambda xy.xyy$. As known, the sequence $\mathbf{O}$, $\mathbf{WO}$, $\mathbf{W(WO)}$,... (Shaap numerals[23]) is an adequate numeral system. Our aim is to give a proof for this, providing a term $\mathbf{It}_s$ (see Ref. 22) which gives a correspondence between the cited sequence and Church numerals, i.e. such that, being $\mathbf{n}_S$ the n-th element of the sequence, $\mathbf{It}_S\ \mathbf{n}_S = \mathbf{n}_C$, the n-th Church numeral.

Since $\mathbf{n}_S = \lambda x.x \dots x$ (n times), choosing $\mathbf{It}_S \equiv \lambda u.u\, H\, U$, the problem is solved if we find H which, whenever applied to itself a number of times and successively applied to U, gives the Church numeral corresponding to the number of times it has been applied to itself.

This amounts to solving the system of equations, in the unknowns h,u:

(a)
$$h\,x\,(h\,y) = h\,(s_C\,x)$$
$$h\,x\,u = x$$

where s_C is Church's successor combinator. In fact, if H' solves (a), then $H = H'\mathbf{O}$ is a solution for our problem. We consider the system:

(b)
$$h\,x\,(h\,y) = y_1 h\,x$$
$$h\,x\,u = y_2 h\,x$$

(c)
$$h\,x\,(h\,y) = y_1 h\,u\,x\,(h\,y)$$
$$h\,x\,u = y_2 h\,u\,x\,u$$

and we note that from (b) we obtain (a) substituting $\lambda\, uv.u(s_C v)$ for y_1 and $\lambda uv.v$ for y_2.

The solvability of (b) is a trivial consequence of Theorem 2, since by it we are able to solve e.g. (c). A possible solution is found taking $h = zz$ and considering the new system, in the unknowns z,u:

(b)*
$$z\,z\,x\,(z\,z\,y) = y_1 (z\,z)\,x$$
$$z\,z\,x\,u = y_2 (z\,z)\,x$$

and substituting $\lambda\, rst.t\,(\lambda\, abcdefg.y_1 (bb) f) rst$ for z and $\lambda\, abcd.y_2 c$ for u.

Note that, starting from the slightly different system:

$\overline{(a)}$
$$h\,x\,(h\,y) = h\,(s_C\,y)$$
$$h\,x\,u = x$$

we can also obtain, following a similar construction, a term H which, composed with itself a number of times, preserves the memory of the number of times it has been composed with itself.

3.2. Make Your Own Fixpoint Combinator

We recall (Ref.2, §6.1) that a fixed point combinator is a term M s.t.

$\forall F,\ M\,F = F\,(M\,F)$.

Two examples of fixed point combinators are Turing's one,

$\mathbf{Y}_T \equiv (\lambda xf.f(xxf))(\lambda xf.f(xxf))$, and Curry's one, $\mathbf{Y}_C \equiv \lambda x.(\lambda b.x(bb))(\lambda b.x(bb))$.

Obviously, they are both solutions of the equation, in the unknown y,

$$y\,x = x\,(y\,x)\ , \tag{3.1}$$

but, in order to recover constructivity, we may ask whether $\mathbf{Y}_C$ and $\mathbf{Y}_T$ can be obtained, via the same method of solution, from different equations equivalent to (3.1): in effect we have (see also Refs. 3 and 4 for different methods for constructing Turing's and Curry's fixpoint combinators):

(i) substituting z z for y in (3.1) we obtain the equation in the unknown z

$$z\,z\,x = x\,(z\,z\,x) \tag{3.2}$$

whose solution is found substituting $\lambda\, xf.f(xxf)$ for z, thus obtaining $\mathbf{Y}_T$.

(ii) substituting $\lambda x.zx(zx)$ for y in (3.1) we obtain the equation in the unknown z

$$z\,x\,(z\,x) = x\,(z\,x\,(z\,x)) \tag{3.3}$$

whose solution is found substituting $\lambda\, ab.a(bb)$ for z, thus obtaining $\mathbf{Y}_C$.

Indeed, it is easy to verify that the following equations in the unknown z are regular:

- $z\,z\,x = y\,z\,x$ from which we obtain (3.2) substituting $\lambda ab.b(aab)$ for y
- $z\,x\,(z\,x) = y\,x\,(z\,x)$ from which we obtain (3.3) substituting $\lambda ab.a(bb)$ for y.

In order to find Turing's and Curry's fixed point combinators, we can solve the equations susbstituting y for the unknown z and then make the mentioned substitution for y. More generally, given the fixed point equation (3.1), any way of rephrasing it as a regular equation leads us to construct a fixed point combinator.

As an example, substitute for y in (3.1) the term $z z M_1 \ldots M_w$, where $M_1,\ldots,M_w$ are arbitrary terms; solving the equation $z\, zM_1\ldots M_w x = y\, zM_1 \ldots M_w x$ and successively taking $y{:=}\lambda ab_1\ldots b_w c.c(a\, b_1\ldots b_w c)$ we succeed in generalizing the theorem in (Ref.2, §6.5.4).

3.3. Recursive Schemes and Functional Programming

Let us turn back to considering the solvability problem for the system (0.1). By Theorem 2, if we assume that in (0.1)

- **succ** is such that it is possible to Böhm-out z from **succ** z;
- The set $\{A_1,\ldots,A_n$, **K**, **zero**, **succ** $\Omega\}$ is distinct,

then the considered system is β-solvable and its solution is the program we were looking for.

As an example, we take Berarducci's numerals[24], where **zero** ≡ λab.b and **succ** ≡ λab.bab, and we exhibit the construction of the solution for the system in Example 2, as described in §2.1.1:

$$\mathbf{S} = (\{\ f(\lambda ab.b) = y_0,\ f(\lambda b.bzb) = y_1 f z,\ f f = \mathbf{K},\ f\mathbf{K} = \mathbf{S},\ f\mathbf{S} = y_2\},\ \{f\}) \qquad (3.4)$$

- Step 1: substitute for f the term $G\,G$ where $G \equiv \lambda ab.bu_1u_2u_3u_4u_5u_6ab$, thus obtaining a new system:

$$\begin{aligned}\mathbf{S}_{(1)} = (\{\ & u_2 u_3 u_4 G(\lambda ab.b) = y_0,\\ & u_1 z u_1 u_2 u_3 u_4 u_5 u_6 G(\lambda b.bzb) = y_1 (GG) z,\\ & u_1 u_1 u_2 u_3 u_4 u_5 u_6 G\, u_1 u_2 u_3 u_4 u_5 u_6 (GG) = \mathbf{K},\\ & u_1 u_3 u_4 u_5 u_6 G\,\mathbf{K} = \mathbf{S},\\ & u_1 u_3 (u_2 u_3) u_4 u_5 u_6 G\,\mathbf{S} = y_2\},\ \{u_1,u_2,u_3,u_4,u_5,u_6\})\end{aligned}$$

- Step 2: substitute for u_1 the term $U_1 \equiv \lambda abcd.dabcd$, thus obtaining a new system:

$$\begin{aligned}\mathbf{S}_{(2)} = (\{\ & u_2 u_3 u_4 G(\lambda ab.b) = y_0,\\ & u_3 z u_1 u_2 u_3 u_4 u_5 u_6 G(\lambda b.bzb) = y_1 (GG) z,\\ & u_4 U_1 u_2 u_3 u_4 u_5 u_6 G\, U_1 u_2 u_3 u_4 u_5 u_6 (GG) = \mathbf{K},\\ & u_6 u_3 u_4 u_5 u_6 G\,\mathbf{K} = \mathbf{S},\\ & u_5 u_3 (u_2 u_3) u_4 u_5 u_6 G\,\mathbf{S} = y_2\},\ \{u_2,u_3,u_4,u_5,u_6\})\end{aligned}$$

- The head variables of left-hand side members of the equations are now mutually different, so we can make the suitable substitutions in order to reconstruct right hand sides:
  - $U_2 \equiv \lambda a_1\ldots a_4.y_0$ for u_2
  - $U_3 \equiv \lambda a_1\ldots a_9.y_1(a_8a_8)a_1$ for u_3
  - $U_4 \equiv \lambda a_1\ldots a_{14}.\mathbf{K}$ for u_4
  - $U_5 \equiv \lambda a_1\ldots a_7.\mathbf{S}$ for u_5
  - $U_6 \equiv \lambda a_1\ldots a_7.y_2$ for u_6

Hence a possible solution for the system (3.4) is

$$F \equiv \lambda b.bU_1U_2U_3U_4U_5U_6(\lambda cd.dU_1U_2U_3U_4U_5U_6cd)b.$$

4. Undecidability Results

In this section we shall consider some classes of combinatory equations and systems for which we shall prove the solvability problem to be undecidable. These results confirm that meaningful generalizations of the class of regular systems turn out to be not decidable. Note also that in the proofs of both the following theorems, we shall make use of non left-regular systems.

Eliminating the condition of regularity, the solvability of an equation between normal forms is not decidable:

Theorem 5.

Given the β(βη)-normal form N, the solvability of the equation, in the unknown x,

$$M\,x = N, \qquad (4.1)$$

where M is a β(βη)-normal form and $x \notin M$, is not decidable in λβ(λβη)-calculus.

Proof. We shall reduce the solvability of (4.1) to the halting problem for recursive-functions.

Since N is a normal form, then for suitable u,v:

$N \equiv \lambda t_1\ldots t_u.\tau T_1\ldots T_v$, where $\tau \in \{t_1,\ldots,t_u\} \cup FV(N)$;

As in Ref. 25, for each Gödel number e we construct, uniformly in e (Ref. 2, §8.2–4), a term P_e s.t., with the usual notation of recursion theory:

$\varphi_e(e)$ converges $\Rightarrow P_e = \mathbf{I}\ \wedge\ \varphi_e(e)$ diverges $\Rightarrow P_e \notin SOL$

and we take the term $P_{e,t}$ obtained from P_e replacing each redex $(\lambda x.Y)X$ by $t\,(\lambda x.Y)X$ where t is a fresh variable not occurring in N.

Proof for λβ-calculus: take a leaf in BT(N), labelled with $\lambda z_1\ldots z_h.\zeta$ ($h \geq 0$), and let N^* be obtained from N replacing that label with $\lambda z_1\ldots z_h.t(P_{e,t}(t\,\zeta))$. Let then $M \equiv \lambda t.N^*$.

It is easy to verify that:

$M\,x = N$ is β-solvable $\Leftrightarrow \varphi_e(e)$ converges.

Proof for λβη-calc.: let a,b be variables not occurring in N and take

$M \equiv \lambda t t_1\ldots t_u ab.\tau T_1\ldots T_v(t\,a)(P_{e,t}\,b)$; then

$M\,x = N$ is βη-solvable $\Leftrightarrow \varphi_e(e)$ converges. ❑

The following theorem proves that dropping condition (v) from the definition of regular system leads to an undecidable problem (see also §2.1.3).

Theorem 6.

The solvability of a system $\mathbf{S} = (\Gamma, \mathcal{X})$ such that any equation of Γ has the shape $xN_1\ldots N_m = y$, where $x \in \mathcal{X}$ and $y \notin (\mathcal{X} \cup FV(\mathcal{L}_\Gamma))$ is not decidable in λβ-calculus..

Sketch of the proof. Let P_e and $P_{e,t}$ be as in the proof of Theorem 5 and let

$$\mathbf{S} = (\Gamma, \{t, z\}),\ \text{with}\ \Gamma = \{t P_{e,t} z = y,\ t(t P_{e,t} z) = y\},$$

which contradicts condition v of Def.2;

then $\mathbf{S}$ is β-solvable $\Leftrightarrow$ $\varphi_e(e)$ converges .

($\Leftarrow$): If $\varphi_e(e)$ converges, then the substitution $[I/t, y/z]$ is a solution for $\mathbf{S}$.

($\Rightarrow$): Let $[A/t, B/z]$ be a solution for $\mathbf{S}$. We will have: $A \equiv \lambda x_1 \ldots x_a . \xi Y_1 \ldots Y_b$ $(a,b \geq 0)$; it is easy to verify that we must have $a = 1$, $\xi \equiv x_1$, hence: $A \equiv \lambda x . x Y_1 \ldots Y_b$ $(b \geq 0)$; substituting into the second equation and taking into account the first one we must have:

$$A\,y = y\,Y_1 \ldots Y_b = y$$

which clearly holds iff $b = 0$. It follows that $A \equiv \mathbf{I}$, hence $P_e \in$ SOL and $\varphi_e(e)$ converges .❑

5. Concluding Remarks

Summarizing, we introduced the class of regular systems as a decidable class of systems of combinatory equations which seems to generalize in a significant way the ones already appearing in the literature. Regular systems have enough expressive power to be used as the core of an equational programming language, in which the compiler is the algorithm constructing the system's solution. Moreover, such class of systems seems not to be easily generalizable in meaningful ways without running into undecidable solvability problems.

Acknowledgements

We are grateful to Prof.Corrado Böhm for useful and stimulating suggestions and discussions about the topics of this paper.

This research has been supported by grants of the Ministero della Pubblica Istruszione (Italy), CNR (Italy) and "Projet Stimulation" (EEC).

References

1. M.J. O'Donnell, *Equational Logic as a Programming Language*, The MIT Press in the Foundations of Computing, 1985.
2. H.P. Barendregt, *The Lambda Calculus, Its Syntax and Semantics* (North Holland, 1984).
3. C. Böhm, "The CUCH as a formal and description language", *IFIP Working Conference on Formal Language Description Languages* (North-Holland, 1966) pp. 179-197.
4. J.W. Klop, "Combinatory reduction systems", Ph.D. thesis, 1980.
5. R. Hindley and J. Seldin, *Introduction to Combinators and λ-calculus* (Cambridge University Press, 1986).
6. C. Böhm, "Alcune proprietà delle forme β–η-normali nel λ–K-calcolo", IAC Pubbl. **696**, Roma, 1968.
7. M. Coppo, M. Dezani-Ciancaglini and S. Ronchi della Rocca, "(Semi-) separability of finite sets of terms in Scott's D∞ models of the λ-calculus", *ICALP '78, LNCS* **62**, 1978, pp. 142-164.
8. R. Statman, Private communication.
9. M. Dezani-Ciancaglini, Private communication.
10. C. Böhm, A. Piperno and E. Tronci, "Solving equations in Lambda Calculus", *Logic Colloquium 88*, Padova (North Holland, 1989) pp. 139-160.
11. C. Böhm and M. Dezani-Ciancaglini, "Combinatorial problems, combinator equations and normal forms", *ICALP '74, LNCS* **14**, 1974, pp. 185-199.
12. I. Margaria and M. Zacchi, "Right and Left Invertibility in λ–β-calculus", RAIRO **17**, 1983, pp. 71-88.
13. M. Dezani-Ciancaglini, "Pattern-matching problems inside λ–β–η-calculus", *Informatica*, 1976.

14. M. Dezani-Ciancaglini, "Characterization of normal forms possessing inverse in the λ–β–η-calculus", *Theor. Comp. Sci.* **2** (1976) pp. 323-337.
15. C. Böhm, M. Dezani-Ciancaglini, P. Peretti and S. Ronchi della Rocca, "A discrimination algorithm inside λ–β-Calculus", *Theor. Comp. Sci.* **6** (1979) 271-291.
16. C. Böhm and A. Piperno, "Surjectivity for finite sets of combinators by weak reduction", *CSL 87, Karlsruhe, LNCS* **329**, 1987, pp. 27-43.
17. C. Böhm and E. Tronci, "About systems of equations, X-separability and left-invertibility in the λ-calculus", *Inf. Comput.* to appear.
18. E. Tronci, "Risolubilità di sistemi di equazioni nel λ-calcolo", Tesi di Laurea in Ing. Elettronica, 1987.
19. C. Böhm and E. Tronci, "X-Separability and Left-Invertibility in Lambda-Calculus", *LICS 87*, Ithaca, N.Y., Computer Soc. of the IEEE, 1987, pp. 320-328.
20. C. Böhm and A. Piperno, "Characterizing X-separability and one-side invertiblity in λ–β–Ω-calculus", *LICS 88*, Edinburgh, Computer Soc. of the IEEE, 1988, pp. 91-101.
21. C. Bohm, "Functional programming and combinatory algebras", *MFCS '88*, LNCS 324, 1988, pp. 14-26.
22. C. Böhm, "Subduing self-application", *ICALP '89, LNCS* **372**, 1989, pp. 108-122.
23. A. Rezus, "Lambda-conversion and logic", Ph.D. thesis, 1981.
24. A. Berarducci, "Programmazione funzionale e rappresentabilità in alcuni sistemi di Logica Combinatoria", Tesi di Laurea in Matematica, 1983.
25. R. Statman, "On translating lambda terms into combinators: the basis problem", *LICS 86*, Boston, 1986, Computer Soc. of the IEEE, pp. 378-382.