Equational Programming in λ-calculus [0]

Enrico Tronci [1]

Department of Mathematics, Carnegie Mellon University, Pittsburgh, PA. 15213, USA

et0o@andrew.cmu.edu

Abstract

A system of equations in the λ- calculus is a pair (Γ, X) where Γ is a set of formulas of Λ (the equations) and X is a finite set of variables of Λ (the unknowns). A system $\mathbb{S}$ = (Γ, X) is said to be solvable in the theory $\mathbb{T}$ ($\mathbb{T}$-solvable) iff there exists a simultaneous substitution with closed λ-terms for the unknowns that makes the formulas of Γ theorems in the theory $\mathbb{T}$. We define a class of systems for which the β-solvability problem is decidable in Polynomial Time. This class yields an equational programming language in which constraints on the code generated by the compiler can be specified by the user and (properties of) data structures can be described in an abstract way.

Keywords : Systems of equations in the λ-calculus, λ-calculus, Equational Programming, Functional Programming, Automated Synthesis of Programs.

0. Introduction

A system $\mathbb{S}$ of equations in the λ-calculus can be viewed as a set of specifications (the equations) for a finite set of programs (the unknowns) whereas a solution for $\mathbb{S}$ yields executable codes for these programs.

0.0. Example. Consider the numerical system ($\underline{0}$, $\underline{s}$, $\underline{p}$, $\underline{Zero}$) ([Ber 83]), where: (the terms U^n_i are defined in 1.0)

$\underline{0} \equiv \lambda a\, b.\, b$, $\underline{s} \equiv \lambda a\, b.\, b\, a\, b$, $\underline{p} \equiv \lambda b.\, b\, U^3_1\, U^2_2$,

$\underline{Zero} \equiv \lambda b.\, b\, (U^4_1\, U^2_2)\, U^2_1$.

Let H_0 and H_1 be given combinators.

To find $F \in \Lambda^o$ s.t.

0.0.0. $F\, \underline{0} = H_0\, F\, \underline{0}$,

0.0.1. $F\, (\underline{s}\, z) = H_1\, F\, z$,

means to find a program that satisfies the specifications expressed by 0.0.0,1 (i.e. 0.0.0 and 0.0.1). By choosing H_0 and H_1 in 0.0.0,1 any partial recursive function can be specified. First note that if F satisfies 0.0.0,1 then F satisfies also: $F\, \underline{0}\, \vec{a} = H_0\, F\, \underline{0}\, \vec{a}$ and $F\, (\underline{s}\, z)\, \vec{a} = H_1\, F\, z\, \vec{a}$, where $\vec{a}$ is any vector of variables. If f is a projection function or a constantly zero function or the successor function or is obtained by composition from other functions or is obtained by primitive recursion it is straightforward to choose H_0 and H_1 so that any solution F satisfying 0.0.0,1 represents the function f. We show that the minimalization operator can be specified and hence a function defined by minimalization can be specified. Let F_{min} satisfying 0.0.0,1 when we choose $H_0 \equiv \lambda a\, z\, g.\, \underline{Zero}\, (g\, z)\, z\, (a\, (\underline{s}\, z)\, g)$ and $H_1 \equiv \lambda a\, z\, g.\, \underline{Zero}\, (g\, (\underline{s}\, z))\, (\underline{s}\, z)\, (a\, (\underline{s}\, (\underline{s}\, z))\, g)$. Then F_{min} represents the minimalization operator. Let f defined by minimalization from g, i.e. $f(x_1, \ldots x_n) = \min y\, [g(x_1, \ldots x_n, y) = 0]$. If we choose: $H_0 \equiv \lambda a\, x_1 \ldots x_n.\, F_{min}\, (G\, x_1 \ldots x_n)$ and $H_1 \equiv \lambda a\, x_1 \ldots x_n.\, F_{min}\, (G\, (\underline{s}\, x_1)\, x_2 \ldots x_n)$ (G is a term representing g) then any combinator F satisfying 0.0.0,1 will represent the function f. □

0.1. Example. Let ($\underline{0}$, $\underline{s}$, $\underline{p}$, $\underline{Zero}$) the numerical system in 0.0. Consider the equations

0.1.0. $D\, \vec{y}\, \underline{0} = y_0\, (D\, \vec{y})\, \underline{0}$,

0.1.1. $D\, \vec{y}\, (\underline{s}\, z) = y_1\, (D\, \vec{y})\, z$,

where $\vec{y} \equiv y_0, y_1$ are variables and D is an unknown combinator. If we can find a solution D for the equations 0.1.0,1 then we can find a solution for the equations 0.0.0,1 defining $F \equiv D\, H_0\, H_1$. Hence, replacing y_0 and y_1 with suitable combinators, the equations in 0.1.0,1 can specify any partial recursive function. □

A class of systems $\mathfrak{S}$ for which the solvability problem is effectively decidable defines an equational programming language $\mathfrak{L}$ and an algorithm to solve the systems in $\mathfrak{S}$ yields a compiler for $\mathfrak{L}$. Both specifications and results of the compilation process can be represented inside the λ-calculus.

---

0. This work is part of a Ph.D thesis that the author is developing under the supervision of Prof. Rick Statman.

1. This research has been partially supported by grants 203.01.50 and 203.01.50 bis from CNR, Italy.

This would not be possible in a term rewriting system without some abstraction mechanism. Moreover specifying sets of λ-terms with equations does not leave out any interesting set, to be precise: any recursively enumerable and β-closed set of closed λ-terms is the set of solutions to a combinator equation [Sta 89]. These features make the λ-calculus appealing as a *calculus* for automated synthesis of programs when specifications are expressed with equations. Unfortunately the existence of a solution for a system (as well as the existence of a program satisfying given specifications) is, in general, undecidable. Nevertheless many interesting equational languages have been defined in the literature (e.g. [O'D 85 (interpreter)], [BB 85 (compiler)], [BT 91 (compiler)], [PT 90 (compiler)]).

Though almost all the equational languages allow the specification of wide classes of recursive functions there are drastic differences respect to the kind of equations the user is allowed to write. The more schemata of equations are allowed in a language the easier is to write specifications. Moreover limiting the class of admissible equations might reduce the class of definable program properties.

0.2. Example. Given the numerical system in 0.0 find a λ-term F s.t.:

0.F represents an arbitrarily given partial recursive function;

1.F has form $\lambda t.\ t\,F_1\,F_2$.

We look for $D \in \Lambda^o$ s.t. $(\vec{y} \equiv y_0, y_1)$

0.2.0. $D\,\vec{y}\,\underline{0} = y_0\,(D\,\vec{y})\,\underline{0}$, (constraint 0)

0.2.1. $D\,\vec{y}\,(\underline{s}\,z) = y_1\,(D\,\vec{y})\,z$,

0.2.2. $D\,\vec{y}\,(\lambda a\,b.\ z) = z$, (constraint 1).

A possible β-solution is:

(the λ-terms $\mathbf{I}$, $\mathbf{U}^n_i$ are defined in 1.0)

$D \equiv G\,G = \lambda\vec{y}\,t.\ t\,(\lambda a_1\,a_2\,a_3.\ y_1\,(t\,(\mathbf{U}^3_1\,\mathbf{I})\,G\,G\,\vec{y})\,a_1)$
$(y_0\,(t\,\mathbf{I}\,\mathbf{I}\,G\,G\,\vec{y})\,t)$, where

$G \equiv \lambda u\,\vec{y}\,t.\ t\,(\lambda a_1\,a_2\,a_3.\ y_1\,(t\,(\mathbf{U}^3_1\,\mathbf{I})\,u\,u\,\vec{y})\,a_1)$
$(y_0\,(t\,\mathbf{I}\,\mathbf{I}\,G\,G\,\vec{y})\,t)$.

Hence (as in 0.0,1) $F \equiv D\,\vec{y}$ (note that F has normal form). Equations 0.2.0,1 are sufficient to specify any partial recursive function (as in 0.0,1), however they cannot express any constraint on the code of a program. This can be done using eq. 0.2.2. The system 0.2.0,1 can be transformed in an X-separability problem (2.1.1) [BT 87, 91], but the system 0.2.0-2 (i.e. 0.2.0, 0.2.1, 0.2.2) cannot because of the presence of equation 0.2.2. □

An unpleasant feature of the known compilers for equational programming is that the user (or someone else for him) has to specify the actual representation of the data structures (as we did in 0.0-2). It would be much better (and in same case essential) to leave this task to the compiler.

0.3. Example. Find a λ-term F and a numerical system ($\underline{0}$, $\underline{s}$, $\underline{p}$, $\underline{Zero}$) s.t.:

0. F represents an arbitrarily given partial recursive function;

1. F has form $\lambda t.\ t\,F_1\,F_2$;

2. A numeral applied to its constructors realizes an arbitrarily given partial recursive function.

We look for $D, D_{\underline{0}}, D_{\underline{s}}, D_{\underline{p}}, D_{\underline{Zero}} \in \Lambda^o$ s.t.:

$(\vec{y} \equiv y_0, y_1, y_2, y_3, y_4, y_5)$

(constraint 0)

0.3.0. $D\,\vec{y}\,(D_{\underline{0}}\,\vec{y}) =$
$y_0\,(D\,\vec{y})\,(D_{\underline{0}}\,\vec{y})\,(D_{\underline{s}}\,\vec{y})\,(D_{\underline{p}}\,\vec{y})\,(D_{\underline{Zero}}\,\vec{y})$,

0.3.1. $D\,\vec{y}\,(D_{\underline{s}}\,\vec{y}\,z) =$
$y_1\,(D\,\vec{y})\,(D_{\underline{0}}\,\vec{y})\,(D_{\underline{s}}\,\vec{y})\,(D_{\underline{p}}\,\vec{y})\,(D_{\underline{Zero}}\,\vec{y})\ z$,

0.3.2. $D\,\vec{y}\,(\lambda a\,b.\ z) = z$, (constraint 1)

(equations 0.3.3-6 specify the data structure)

0.3.3. $D_{\underline{p}}\,\vec{y}\,(D_{\underline{0}}\,\vec{y}) = (D_{\underline{0}}\,\vec{y})$,

0.3.4. $D_{\underline{p}}\,\vec{y}\,(D_{\underline{s}}\,\vec{y}\,z) = z$,

0.3.5. $D_{\underline{Zero}}\,\vec{y}\,(D_{\underline{0}}\,\vec{y}) = y_2$,

0.3.6. $D_{\underline{Zero}}\,\vec{y}\,(D_{\underline{s}}\,\vec{y}\,z) = y_3$.

(constraint 2)

0.3.7. $D_{\underline{0}}\,\vec{y}\,(D_{\underline{s}}\,\vec{y})\,(D_{\underline{0}}\,\vec{y}) =$
$y_4\,(D\,\vec{y})\,(D_{\underline{0}}\,\vec{y})\,(D_{\underline{s}}\,\vec{y})\,(D_{\underline{p}}\,\vec{y})\,(D_{\underline{Zero}}\,\vec{y})$,

0.3.8. $D_{\underline{s}}\,\vec{y}\,z\,(D_{\underline{s}}\,\vec{y})\,(D_{\underline{0}}\,\vec{y}) =$
$y_5\,(D\,\vec{y})\,(D_{\underline{0}}\,\vec{y})\,(D_{\underline{s}}\,\vec{y})\,(D_{\underline{p}}\,\vec{y})\,(D_{\underline{Zero}}\,\vec{y})\ z$,

($\underline{p} \equiv D_{\underline{p}}\,\vec{y}$ is a particular (0.3.3) left inverse for $\underline{s} \equiv D_{\underline{s}}\,\vec{y}$ (0.3.4), $\underline{Zero} \equiv D_{\underline{Zero}}\,\vec{y}$ recognizes the constructors (i.e. $\underline{0}$ and $\underline{s}$) (0.3.5,6)). The system 0.3.0-8 has a β-solution. Defining: $F = D\,\vec{y}$, $\underline{0} \equiv D_{\underline{0}}\,\vec{y}$, $\underline{s} \equiv D_{\underline{s}}\,\vec{y}$, $\underline{p} \equiv D_{\underline{p}}\,\vec{y}$, $\underline{Zero} \equiv D_{\underline{Zero}}\,\vec{y}$ we solve our problem. With this specification of the data structure we always get a *one shot predecessor* $\underline{p}$ (i.e. $\underline{s}$ is left-invertible in Λ). Consider the equation

0.3.9. $D\,\vec{y}\,\mathbf{U}^2_2 = \mathbf{U}^2_1$.

The system 0.3.0-9 is still β-solvable, however the system 0.2.0-2, 0.3.9 is no longer β-solvable because there is a conflict between eq. 0.2.0 and eq. 0.3.9 since we chose $\underline{0} = \mathbf{U}^2_2$. To avoid these ***unnatural*** conflicts the actual representation of the data structures should be chosen by the compiler (i.e. the constructors of the data structures should be regarded as unknowns in the system). The system 0.3.0-8 cannot be transformed in an X-separability problem (2.1.1) because of the presence of equations 0.3.2,4. Moreover the system 0.3.0-8 cannot be solved with the methods in [BB 85] because the representation of the data structure is an unknown (and it is determined together with the representation for the program). □

Because after all the constructors of a data structure are programs nothing prevents us from putting additional constraints on them.

0.4. Example. Find a representation ($\underline{0}$, $\underline{s}$, $\underline{p}$, $\underline{Zero}$) for the natural numbers s.t.:

0. The application of two natural numbers realizes an arbitrarily given partial recursive function;
1. The λ-term representing the successor has form $\lambda a\, b.\ b\, L_1\, L_2$.

We look for $D_{\underline{0}}, D_{\underline{s}}, D_{\underline{p}}, D_{\underline{Zero}} \in \Lambda^o$ s.t.:

($\vec{y} \equiv y_0, y_1, y_2, y_3, y_4, y_5$)

(specification of the data structure (as in 0.3.3-6))

0.4.0. $D_{\underline{p}}\, \vec{y}\, (D_{\underline{0}}\, \vec{y}) = D_{\underline{0}}\, \vec{y}$,

0.4.1. $D_{\underline{p}}\, \vec{y}\, (D_{\underline{s}}\, \vec{y}\, z) = z$,

0.4.2. $D_{\underline{Zero}}\, \vec{y}\ (D_{\underline{0}}\, \vec{y}) = y_4$,

0.4.3. $D_{\underline{Zero}}\, \vec{y}\ (D_{\underline{s}}\, \vec{y}\ z) = y_5$,

(constraint 0)

0.4.4. $D_{\underline{0}}\, \vec{y}\ (D_{\underline{0}}\, \vec{y}) =$
$y_0\, (D_{\underline{0}}\, \vec{y})\, (D_{\underline{s}}\, \vec{y})\, (D_{\underline{p}}\, \vec{y})\, (D_{\underline{Zero}}\, \vec{y})$,

0.4.5. $D_{\underline{0}}\, \vec{y}\ (D_{\underline{s}}\, \vec{y}\ z) =$
$y_1\, (D_{\underline{0}}\, \vec{y})\, (D_{\underline{s}}\, \vec{y})\, (D_{\underline{p}}\, \vec{y})\, (D_{\underline{Zero}}\, \vec{y})\, z$,

0.4.6. $D_{\underline{s}}\, \vec{y}\ z\, (D_{\underline{0}}\, \vec{y}) =$
$y_2\, (D_{\underline{0}}\, \vec{y})\, (D_{\underline{s}}\, \vec{y})\, (D_{\underline{p}}\, \vec{y})\, (D_{\underline{Zero}}\, \vec{y})\, z$,

0.4.7. $D_{\underline{s}}\, \vec{y}\ u\, (D_{\underline{s}}\, \vec{y}\ z) =$
$y_3\, (D_{\underline{0}}\, \vec{y})\, (D_{\underline{s}}\, \vec{y})\, (D_{\underline{p}}\, \vec{y})\, (D_{\underline{Zero}}\, \vec{y})\, u\, z$,

0.4.8. $D_{\underline{s}}\, \vec{y}\, \Omega\, (\lambda a\, b.\ z) = z$. (constraint 1).

Choosing $\underline{0} \equiv D_{\underline{0}}\, \vec{y}$, $\underline{s} \equiv D_{\underline{s}}\, \vec{y}$, $\underline{p} \equiv D_{\underline{p}}\, \vec{y}$, $\underline{Zero} \equiv D_{\underline{Zero}}\, \vec{y}$ yields the wanted representation for the natural numbers. A possible β-solution for this system is ($\vec{v} \equiv v_1, v_2, v_3, v_4$) (the terms $\mathbf{P}_q$ are defined in 1.0): $D_{\underline{0}} \equiv G_0{}^*\, G_0{}^*\, G_1{}^*\, D_{\underline{p}}\, D_{\underline{Zero}}$, $D_{\underline{s}} \equiv G_1{}^*\, G_0{}^*\, G_1{}^*\, D_{\underline{p}}\, D_{\underline{Zero}}$, $D_{\underline{p}} \equiv \lambda \vec{y}\, t.\ t\, \mathbf{P}_2\, \mathbf{U}^4_2\, \mathbf{U}^3_3\, t$, $D_{\underline{Zero}} \equiv \lambda \vec{y}\, t.\ t\, \mathbf{P}_2\, (\lambda a_1 \ldots a_4.\ y_5)\, (\lambda a_1\, a_2\, a_3.\ y_4)\, t$, where:

$G_0{}^* \equiv \lambda \vec{v}\, \vec{y}\, t.\ ((G_0\, \vec{y}\, t)$

$[y_0 := y_0\, (t\, (\mathbf{U}^2_1\, \mathbf{I})\, v_1\, \vec{v}\, \vec{y})\, (t\, (\mathbf{U}^2_1\, \mathbf{I})\, v_2\, \vec{v}\, \vec{y})\, (v_3\, \vec{y})\, (v_4\, \vec{y})$,

$y_1 := y_1\, (t\, (\mathbf{U}^3_1\, \mathbf{I})\, v_1\, \vec{v}\, \vec{y})\, (t\, (\mathbf{U}^3_1\, \mathbf{I})\, v_2\, \vec{v}\, \vec{y})\, (v_3\, \vec{y})\, (v_4\, \vec{y})])$,

$G_1{}^* \equiv \lambda \vec{v}\, \vec{y}\, t_1\, t_2.\ ((G_1\, \vec{y}\, t_1\, t_2)$

$[y_2 := y_2\, (t_2\, (\mathbf{U}^2_1\, \mathbf{I})\, v_1\, \vec{v}\, \vec{y})\, (t_2\, (\mathbf{U}^2_1\, \mathbf{I})\, v_2\, \vec{v}\, \vec{y})\, (v_3\, \vec{y})\, (v_4\, \vec{y})$,

$y_3 := y_3\, (t_2\, (\mathbf{U}^3_1\, \mathbf{I})\, v_1\, \vec{v}\, \vec{y})\, (t_2\, (\mathbf{U}^3_1\, \mathbf{I})\, v_2\, \vec{v}\, \vec{y})\, (v_3\, \vec{y})\, (v_4\, \vec{y})])$,

$G_0 \equiv \lambda \vec{y}\, t.\ t\, (t\, \mathbf{P}_2\, (\lambda a_1 \ldots a_6.\ y_1\, a_2)\, (\lambda a_1 \ldots a_4.\ y_0)\, t)$,

$G_1 \equiv$

$\lambda \vec{y}\, t_1\, t_2.\ t_2\, (t_2\, \mathbf{P}_2\, (\lambda a_1 \ldots a_7.\ y_3\, a_7\, a_6)\, (\lambda a_1 \ldots a_4.\ y_2)\, t_2)\, t_1$.

The system 0.4.0-8 cannot be transformed in an X-separability problem (2.1.1) because of the presence of equations 0.4.1,8 and, as in 0.3, cannot be solved with the methods in [BB 85]. □

Of course analogous considerations apply to any data structure definable with a (heterogeneous) term algebra (8.2). Systems like those in Ex. 0.2-4 cannot be solved with the methods in [BB 85], [BT 87, 91], [BPT 88] or [PT 90] (2.2). As a matter of fact the β (βη)-solvability problem for this kind of systems is, in general, undecidable (3.1 and 3.3). The simultaneous presence of self-application (e.g. equation 0.4.4) and bounding on the degree (defined in 1.1.6) of (subterms of) the solutions (e.g. equation 0.4.8) constitutes the main difficulty to face for this kind of systems. Consider equation 0.2.2: the degree of $D\, \vec{y}$ is bounded by the order (defined in 1.1.6) of ($\lambda a\ b.\ z$). The smaller this order the more cleverness we need to find a solution. For this reason the solvability of systems similar to those in Ex.0.2-4 is, in general, undecidable. In this paper we define a class of systems (5.3, 8.0) (strictly larger then the classes introduced in [BT 87, 91]) containing systems like those in Ex. 0.2-4 and for which the β-solvability problem is decidable (7.3, 8.0) in Polynomial Time (9.0,1). This class defines an equational programming language in which constraints on the code generated by the compiler can be specified (e.g. as 0.2.2) and (properties of) data structures can be described in an abstract way (e.g. as in 0.3,4).

1. The λ -calculus

We assume the reader familiar with [Bar 84] of which, unless otherwise stated, we use notations and conventions.

Var is the set of variables of Λ, the symbol ≡ denotes syntactic equality; $\vec{M} \equiv M_1, \ldots M_n$; $|\vec{M}| = n$; $\{\vec{M}\} = \{M_1, \ldots M_n\}$.

1.0. Example. $\mathbf{U}^n_i \equiv \lambda x_1 \ldots x_n.\ x_i$ $(1 \le i \le n)$, $\mathbf{I} \equiv \mathbf{U}^1_1$, $\mathbf{K} \equiv \mathbf{U}^2_1$, $\omega \equiv \lambda x.\ x\, x$, $\Omega \equiv \omega\, \omega$, $\mathbf{P}_q \equiv \lambda x_1 \ldots x_q\, x_{q+1}.\ x_{q+1}\, x_1 \ldots x_q$, $<M_1, \ldots M_n> \equiv \mathbf{P}_n\, M_1 \ldots M_n$ are λ-terms. □

A term M is said to be λ-free if $M \equiv y\, \vec{M}$ and $\mathfrak{I} \subsetneq \Lambda$ is said to be λ-free if its elements are λ-free. Λ[ ] (Λ^0[ ]) is the set of contexts on Λ (with no free variables) ([Bar 84, 2.1.18]). If $Z \subseteq Var$ then $M[Z := N]$ is the λ-term obtained from M by substituting N for all the (free) occurrences of $z \in Z$ in M. We write $M[z := N]$ for $M[\{z\} := N]$ and $(M = N)[Z := Q]$ for $M[Z := Q] = N[Z := Q]$. $Form(\Lambda) = \{M = N \in \Lambda \mid M, N \in \Lambda\}$. The elements of Form($\Lambda$) are called formulas of Λ. A theory $\mathbb{T}$ is a set of formulas of Λ. If $M = N \in \mathbb{T}$ we write $M =_{\mathbb{T}} N$ and we say that $M = N$ is a theorem in $\mathbb{T}$. We write $M =_{(\eta)} N$ for $M = N \in \boldsymbol{\lambda}\ (\boldsymbol{\lambda\eta})$ [Bar 84, 2.1.4,28] (it will be clear from the mathematical context if $M = N$ is a formula or a theorem of $\boldsymbol{\lambda}$). A theory $\mathbb{T}$ is called a λ-theory if $\mathbb{T}$ is consistent and $\mathbb{T} = \boldsymbol{\lambda} + \mathbb{T}$ [Bar 84, 2.1.30, 4.1.1]. SOL is the set of solvable λ- terms [Bar 84, 2.2.10-12]. We write $\mathbb{T}$ is sms for $\mathbb{T}$ is semi sensible ([Bar 84, 4.1.7]) ($\boldsymbol{\lambda}$ and $\boldsymbol{\lambda\eta}$ are sms theories). We say that M has nf if M has β-normal form ([Bar 84, 3.1.8]). Conventions: $A \subset_f B$ stands for $A \subset B$ and A is finite, $\max \varnothing = 0$.

1.1. Definition. Let $M, N \in \Lambda$, $\mathfrak{F} \subset_f \Lambda$ and $\alpha \in Seq$ (see [Bar 84, pg. xiii, 10.1.7-13, 10.2.3, 10.2.18-21]).

0. $\mathfrak{F}_\alpha = \{M_\alpha \mid M \in \mathfrak{F}\}$.
1. We write $\alpha \in_\eta BT(M)$ iff
   $\forall \beta < \alpha\, [\beta \in BT(M) \Rightarrow M_\beta \in SOL]$.
2. We write $\alpha \in_\eta BT(\mathfrak{F})$ iff $\forall Q \in \mathfrak{F}\ [\alpha \in_\eta BT(Q)]$.
3. We write $M \mid \alpha \downarrow$ iff $[\alpha \in_\eta BT(M)$ and $M_\alpha \in SOL]$, $M \mid \alpha \uparrow$ otherwise.
4. We write $\mathfrak{F} \mid \alpha \downarrow$ iff $\forall Q \in \mathfrak{F}\ Q \mid \alpha \downarrow$.
5. Let $Q \in \Lambda$. We define: $M^{(<>, Q)} \equiv Q\, M$,
   $M^{(<j> * \alpha, Q)} \equiv$
   if $M = \lambda x_1 \ldots x_n.\, y\, M_0 \ldots M_{m-1}$
   then
   if $j < m$
   then $\lambda x_1 \ldots x_n.\, y\, M_0 \ldots M_j^{(\alpha, Q)} \ldots M_{m-1}$
   else $\lambda x_1 \ldots x_n t_m \ldots t_j.$
   $y\, M_0 \ldots M_{m-1} t_m \ldots t_j^{(\alpha, Q)}$
   else $Q\, M$.
6. We define the functions deg (degree), ord (order), head (head) as follows:
   if $M = \lambda x_1 \ldots x_n.\, y\, M_1 \ldots M_m$ then $deg(M) = m$, $ord(M) = n$, $head(M) \equiv y$;
   if $M \notin SOL$ then $deg(M) = ord(M) = 0$ and $head(M) \uparrow$.
7. We write $M \sqsubseteq N$ for $BT(M) \subseteq BT(N)$.
8. We write $C[\ ] \sqsubseteq D[\ ]$ for $C[z] \sqsubseteq D[z]$ (z fresh).
9. $M \sim_\alpha M$ iff $[[M \mid \alpha \uparrow$ and $N \mid \alpha \uparrow]$ or
   $[deg(M_\alpha) - ord(M_\alpha) = deg(N_\alpha) - ord(N_\alpha)$ and $head(M_\alpha) \equiv head(N_\alpha)]]$.
10. [Bar 84, 10.4.6] The node α is said to be useful for $\mathfrak{F}$ (α is usf for $\mathfrak{F}$) iff $[\mathfrak{F} \mid \alpha \downarrow$ and $[Card(\mathfrak{F}) = 1$ or $\exists M, N \in \mathfrak{F}\ \neg [M \sim_\alpha N]]]$.
11. [Bar 84, 10.3.10] We say that $\mathfrak{F}$ agrees up to α (α is agt for $\mathfrak{F}$) iff $[\forall M, N \in \mathfrak{F}\ \forall \beta < \alpha\ M \sim_\beta N]$
12. [BT 87, 91] The node α is said to be $\mathfrak{F}$-adherent (α is adh for $\mathfrak{F}$) iff $[\exists M \in \mathfrak{F}\ \alpha \in BT(M)]$.
13. [BT 91] The binary relation ind($\mathfrak{F}$) is defined as follows:
    $\forall P, Q \in \Lambda\, [ind(\mathfrak{F}, P, Q)$ iff
    $[P, Q \in \mathfrak{F}$ and $[\forall \alpha \in_\eta BT(\mathfrak{F})$
    $[[\alpha$ is usf and agt for $\mathfrak{F}$ and $Card(\mathfrak{F}) > 1] \Rightarrow$
    $[P \sim_\alpha Q$ and $ind(\{M \in \mathfrak{F} \mid P \sim_\alpha M\}, P, Q)]]]]]$.
14. Let $\vec{x} \equiv x_1, \ldots x_n \in Var$ and $\vec{y} \in (Var - \{\vec{x}\})$.
    $(\vec{x} : \vec{y})$ is the sequence $(x_1\, \vec{y}), \ldots (x_n\, \vec{y})$. □

1.2. Remark. [BT 87] Let $\mathfrak{F} \subset_f \Lambda$ and $\alpha \in Seq$. If α is usf and agt for $\mathfrak{F}$ then α is usf, agt and adh for $\mathfrak{F}$. □

1.3. Definition. 0. A system of equations (system) is a pair (Γ, X) where $\Gamma \subseteq Form(\Lambda)$ and $X \subset_f Var$.
Unless otherwise specified we will assume that Γ is finite.

1. Let $\mathbb{S} = (\Gamma, X)$ be a system. A formula $M = N \in \Gamma$ is said to be an equation of $\mathbb{S}$. By abuse of language we write also $M = N \in \mathbb{S}$. A variable $x \in X$ is said to be an unknown of $\mathbb{S}$.
2. Let $\mathbb{T}$ be a theory and $\mathbb{S} = (\Gamma, \{x_1, \ldots x_n\})$ be a system.

2.0. $\mathbb{S}$ is said to be $\mathbb{T}$-solvable iff
  $\exists D[\ ] \equiv (\lambda x_1 \ldots x_n.[\ ])\, D_1 \ldots D_n \in \Lambda^0[\ ]$
  s.t. $\forall M = N \in \mathbb{S}\ D[M] =_{\mathbb{T}} D[N]$
  (note that $D_1, \ldots D_n \in \Lambda^0$).
  $D[\ ] \equiv (\lambda x_1 \ldots x_n.[\ ])\, D_1 \ldots D_n$ is said to be a $\mathbb{T}$-solution for $\mathbb{S}$.

2.1. If $\mathbb{S} = (\Gamma, \{x\})$ and $D[\ ] \equiv (\lambda x.[\ ])\, D$ is a $\mathbb{T}$-solution for $\mathbb{S}$ by abuse of language we say also that D is a $\mathbb{T}$-solution for $\mathbb{S}$.

3. Let $\mathbb{S} = (\Gamma, X)$ be a system. We define:
   $left(\mathbb{S}) = left(\Gamma) = \{M \mid \exists M = N \in \mathbb{S}\}$.
4. Let $\mathbb{S} = (\Gamma, X)$ be a system. We define:
   $right(\mathbb{S}) = right(\Gamma) = \{N \mid \exists M = N \in \mathbb{S}\}$.
5. We define: $Card(\mathbb{S}) = Card(\Gamma)$.

6. Let $\mathbf{S} = (\Gamma, X)$ be a system with equations having form $x\,\vec{M} = \lambda\vec{a}.\ y\,\vec{Q}$, where $x \in X$ and $y \in (FV(\lambda\vec{a}.\ y\,\vec{Q}) - X)$. We define: $\mathbf{S}_\Omega = (\{M\,[(FV(M) - (X \cup \{head(N)\})) := \Omega] = head(N) \mid M = N \in \mathbf{S}\}, X)$. □

2. SL-systems

Many interesting systems (e.g. those in Ex. 0.1-4) can be transformed in systems with equations having form $x\,\vec{M} = z$ (SL-systems). Solving SL-systems will be the core of our systems solving algorithm.

2.0. Definition. 0. A system $\mathbf{S} = (\Gamma, X)$ is said to be an SL-system (separation like) if its equations have form $x\,\vec{M} = z$, where $x \in X$ and $z \notin X$.

1. An SL-system $\mathbf{S} = (\Gamma, \{x\})$ is said to be an HSL-system (head separation like) if its equations have form $x\,\vec{M} = z$, where $x \notin FV(\vec{M})$. □

A separability problem (2.1.0) ([CDR 78], [Bar 84, 10.4.4]) is a particular HSL-system and an X-separability problem (2.1.1) ([BT 87, 91]) is a particular SL-system.

2.1. Example. 0. Let $\mathfrak{F} = \{M_1, \ldots M_m\} \subset_f \Lambda^o$ and $\mathbb{T}$ be a λ-theory. $\mathfrak{F}$ is said to be $\mathbb{T}$-separable iff $\mathbf{S} = (\{x\,\vec{y}\,M_1 = y_1, \ldots x\,\vec{y}\,M_m = y_m\}, \{x\})$ is $\mathbb{T}$-solvable ($\vec{y} \equiv y_1 \ldots y_m$ fresh).

1. Let $\mathfrak{F} = \{M_1, \ldots M_m\} \subset_f \Lambda$ and $\mathbb{T}$ be a λ-theory. $\mathfrak{F}$ is said to be $\{\vec{x}\}$-separable iff (see 1.1.14) $\mathbf{S} = (\{(\lambda\vec{x}.\ M_i)\,(\vec{x} : \vec{y}) = y_i \mid i = 1, \ldots m\}, \{\vec{x}\})$ is $\mathbb{T}$-solvable ($\vec{y} \equiv y_1 \ldots y_m$ fresh). □

However an HSL-system is more general than a separability problem and an SL-system is more general than an X-separability problem (2.2).

2.2. Example. Let $\mathbf{S} = (\{x\,(\lambda a.\ a\,U_1^2\,(a\,z\,\Omega\,\Omega)) = z,\ x\,(\lambda a.\ a\,U_2^2\,(a\,\Omega\,z\,\Omega)) = z,\ x\,(\lambda a.\ z) = z\}, \{x\})$.
A possible β-solution for $\mathbf{S}$ is
$D \equiv \lambda t.\ t\,(\lambda a\,b.\ t\,\mathbf{P}_3\,\Omega < \lambda u\,v.\ u\,U_1^3,\ \lambda u\,v.\ u\,U_2^3 >)$,
but there is no β-solution for $\mathbf{S}$ having form
2.2.0.: $\lambda t_1 \ldots t_e \ldots t_p.\ t_e\,H_1 \ldots H_q\,t_{i_1} \ldots t_{i_r}$,
where $H_1, \ldots H_q \in \Lambda^o$ and $r \le p$. (Hence a β-solution for $\mathbf{S}$ must have a *local memory*, e.g. the rightmost occurrence of t in D.) In fact suppose $\mathbf{S}$ has a β-solution L having form 2.2.0. The only possible cases yield an absurd.

Case 0. $L = \lambda t.\ t\,H$. Then:
$L\,(\lambda a.\ a\,U_1^2\,(a\,z\,\Omega\,\Omega)) = H\,U_1^2\,(H\,z\,\Omega\,\Omega)$,
$L\,(\lambda a.\ a\,U_2^2\,(a\,\Omega\,z\,\Omega)) = H\,U_2^2\,(H\,\Omega\,z\,\Omega)$.
Because $H \in \Lambda^o \cap$ SOL we have two subcases:
0. $H = \lambda t_1\,t_2.\ t_1\,\vec{Q}$. Then $(H\,\Omega\,z\,\Omega) \notin$ SOL.
1. $H = \lambda t_1\,t_2.\ t_2\,\vec{Q}$. Then $(H\,z\,\Omega\,\Omega) \notin$ SOL.
Case 1. $L = \lambda t.\ t\,t$. Then $L\,(\lambda a.\ a\,U_1^2\,(a\,z\,\Omega\,\Omega)) \neq z$.

On the other hand if a (X -) separability problem has a β-solution then it has a β-solution having form 2.2.0 (([BPT 88]) [CDR 78], [Bar 84, 10.4.12]). □

As a matter of fact HSL-systems are so powerful that their β ($\beta\eta$)-solvability problem is undecidable (3.1). However the β-solvability for an interesting class of SL-systems can be characterized (7. 3).

3. Undecidability results

To avoid hopeless search for a characterization we give some undecidability result.
The following idea comes from [Sta 87].

3.0. Definition. [Sta 87] Let $k \in \mathbb{N}$. We define: $P_k \equiv \lambda x.\ G_k\,x \ulcorner k \urcorner\,\mathbf{K}\,\mathbf{I}\,\mathbf{I}$, where: $G_k \equiv \lambda t.\ F_k^{\#}$, $F_k^{\#}$ is obtained from F_k replacing every redex $(\lambda a.\ P)\,Q$ in F_k by $t\,(\lambda a.\ P)\,Q$ (hence $F_k^{\#}$ has nf), F_k λ-defines $\{k\}$ (the k -th partial recursive function) as in [Bar 84, 8.4]. Note that P_k has nf. We have: $P_k\,\mathbf{I} =_\beta$ **if** $\{k\}(k)\downarrow$ **then I else** unsolvable. □

The β ($\beta\eta$)-solvability problem for HSL-systems is, in general, undecidable.

3.1. Proposition. Let $\mathbf{S} = (\{x\,(\lambda a.\ a\,z) = z,\ x\,(\lambda a.\ a\,(a\,u)) = u,\ x\,(\lambda a.\ a\,(a\,(P_k\,a\,y))) = y\}, \{x\})$.
Then $\mathbf{S}$ is β ($\beta\eta$)-solvable iff $\{k\}(k)\downarrow$.
Proof. ($\Leftarrow$). Then $D \equiv \lambda t.\ t\,\mathbf{I}$ is a β-solution for $\mathbf{S}$.
($\Rightarrow$). Let D[ ] a $\beta\eta$-solution for $\mathbf{S}$. Let σ_1 be the standard reduction $D[x\,(\lambda a.\ a\,z)] \twoheadrightarrow_\beta$
$F \equiv \lambda t_1 \ldots t_r.\ (\lambda a.\ a\,z)\,H_{1,1} \ldots H_{1,h} \twoheadrightarrow_\beta$
$Q \equiv \lambda t_1 \ldots t_q.\ z\,Q_{1,1} \ldots Q_{1,q} \twoheadrightarrow_{\beta\eta} z$, where:
F is the first term in the reduction σ_1 in which $(\lambda a.\ a\,z)$ is on the head and the leftmost occurrence of z in F will come on the head in σ_1 and Q is the first term in σ_1 where z is on the head. Let $* \equiv [z, u, y := \Omega]$. We have:
$\forall i \in \{1, \ldots q\}\ [Q_{1,i} \twoheadrightarrow_{\beta\eta} t_i$ and $Q_{1,i}^* \twoheadrightarrow_{\beta\eta} t_i]$ and

$\lambda t_1 \ldots t_r. (\lambda a.\, a\, z)\, H_{1,1}{}^* \ldots H_{1,h}{}^* \twoheadrightarrow_\beta$
$\lambda t_1 \ldots t_r.\, H_{1,1}{}^*\, z\, H_{1,2}{}^* \ldots H_{1,h}{}^* \twoheadrightarrow_\beta$
$\lambda t_1 \ldots t_q.\, z\, G_{1,1} \ldots G_{1,q} \twoheadrightarrow_{\beta\eta} z.$
Consider now the standard reduction
$\sigma_2 : D[x\, (\lambda a.\, a\, (a\, u))] \twoheadrightarrow_{\beta\eta} u$. We have:
$D[x\, (\lambda a.\, a\, (a\, u))] \twoheadrightarrow_\beta$
$\lambda t_1 \ldots t_r. (\lambda a.\, a\, (a\, u))\, H_{2,1} \ldots H_{2,h} \twoheadrightarrow_\beta$
$\lambda t_1 \ldots t_r.\, H_{2,1}\, (H_{2,1}\, u)\, H_{2,2} \ldots H_{2,h} \twoheadrightarrow_\beta$
$\lambda t_1 \ldots t_q.\, H_{2,1}\, u\, Q_{2,1} \ldots Q_{2,q} \twoheadrightarrow_{\beta\eta} u,$
where $\forall i \in \{1, \ldots h\}\ \exists L_i \in \Lambda^o$
$[H_{1,i} = L_i\, (\lambda a.\, a\, z)$ and $H_{2,i} = L_i\, (\lambda a.\, a\, (a\, u))]$ and
$\forall i \in \{1, \ldots q\}\ \exists V_i \in \Lambda^o$
$[Q_{1,i} = V_i\, (\lambda a.\, a\, z)$ and $Q_{2,i} = V_i\, (\lambda a.\, a\, (a\, u))]$.
Hence $\forall i \in \{1, \ldots q\}\ [Q_{1,i}{}^* \sqsubseteq Q_{2,i}]$.
This implies $Q_{2,i} =_\eta t_i$. Hence $H_{2,1}\, u =_\eta u$.
Suppose $H_{2,1}{}^*\, u \notin$ SOL, then $H_{1,1}{}^*\, z \notin$ SOL. This is absurd. Hence $H_{2,1}{}^*\, u \in$ SOL. This implies $H_{2,1}{}^* =_\eta \mathbf{I}$ and because $H_{2,1}{}^* \sqsubseteq H_{2,1}$ we have also $H_{2,1} =_\eta \mathbf{I}$.
Consider the standard reduction
$\sigma_3 : D[x\, (\lambda a.\, a\, (a\, (P_k\, a\, y)))] \twoheadrightarrow_{\beta\eta} y$.
We have: $D[x\, (\lambda a.\, a\, (a\, (P_k\, a\, y)))] \twoheadrightarrow_\beta$
$\lambda t_1 \ldots t_r. (\lambda a.\, a\, (a\, (P_k\, a\, y)))\, H_{3,1} \ldots H_{3,h} \twoheadrightarrow_\beta$
$\lambda t_1 \ldots t_r.\, H_{3,1}\, (H_{3,1}\, (P_k\, H_{3,1}\, y))\, H_{3,2} \ldots H_{3,h} \twoheadrightarrow_\beta$
$\lambda t_1 \ldots t_q.\, (H_{3,1}\, (P_k\, H_{3,1}\, y))\, Q_{3,1} \ldots Q_{3,q}$.
As before we have: $\forall i \in \{1, \ldots q\}\ [Q_{1,i}{}^* \sqsubseteq Q_{3,i}$ and $Q_{3,i} =_\eta t_i]$. Moreover $\mathbf{I} =_\eta H_{2,1}{}^* \sqsubseteq H_{3,1}$, hence $H_{3,1} =_\eta \mathbf{I}$. This implies $P_k\, \mathbf{I}\, y \in$ SOL and $P_k\, \mathbf{I} \in$ SOL.
Hence $\{k\}(k)\downarrow$. □

Thus, though both one side (left, right) invertibility problems are decidable in β ([BD 74], [MZ 83]), the problem of finding a common left inverse for a finite set of combinators is, in general, undecidable in β as well as in $\beta\eta$.

The existence of a common β ($\beta\eta$)-right inverse for a finite set of combinators is undecidable as well [Sta 87] (3.2). We report the proof with a system $\mathbf{S}$ (inspired from [PT 90, 4.6]) slightly different from that in [Sta 87].

3.2. Proposition. [Sta 87] The SL-system
$\mathbf{S} = (\{\, x\, y\, \mathbf{U}^2_1\ (P_k\, (x\, y\, \mathbf{U}^2_1))\, (x\, y\, \mathbf{U}^2_2) = y,$
$x\, y\, \mathbf{U}^2_1\, (x\, y\, \mathbf{U}^2_1\ (P_k\, (x\, y\, \mathbf{U}^2_1))\, (x\, y\, \mathbf{U}^2_2)) = y\},\ \{x\})$
is β ($\beta\eta$)-solvable iff $\{k\}(k)\downarrow$.

Proof. ($\Leftarrow$). Then $D \equiv \lambda t_1\, t_2.\, t_2\, \mathbf{I}\, t_1$ is a β-solution for $\mathbf{S}$.
($\Rightarrow$). Let D a $\beta\eta$-solution for $\mathbf{S}$. Then $D\, y\, \mathbf{U}^2_1\, y =_\eta y$.
If $D\, y\, \mathbf{U}^2_1\, \Omega =_\eta y$ then $D\, y\, \mathbf{U}^2_1\ (P_k\, (D\, y\, \mathbf{U}^2_1))\, (D\, y\, \mathbf{U}^2_2)$ $=_\eta y\, (D\, y\, \mathbf{U}^2_2) =_\eta y$, which is impossible.
Hence $D\, \Omega\, \mathbf{U}^2_1\, y =_\eta y$. Then $D\, \Omega\, \mathbf{U}^2_1 =_\eta \mathbf{I}$. We have:
$\mathbf{I}\, (\mathbf{I}\ (P_k\, \mathbf{I})\, (D\, y\, \mathbf{U}^2_2)) =_\eta P_k\, \mathbf{I}\, (D\, y\, \mathbf{U}^2_2) =_\eta y$.
Hence $P_k\, \mathbf{I} \in$ SOL. This implies $\{k\}(k)\downarrow$. □

The β-solvability problem for systems like those in Ex. 0.2-4 is, in general, undecidable (3.3.1 and 3.1).

3.3. Proposition. The following systems are β-solvable iff $\{k\}(k)\downarrow$:
0. $\mathbf{S} = (\{\, x\, y\, (\lambda a\, b.\, P_k\, (x\, y\, (\mathbf{U}^3_1\, \mathbf{I})\, f)\, a)\, (x\, y\, (\lambda u\, v.\, z)\, f) = y\, z\},\ \{x\})$.
1. $\mathcal{P} = (\{\, x\, y\, (\lambda a\, b.\, P_k\, (x\, y\, (\mathbf{U}^3_1\, \mathbf{I})\, f)\, a)\, (x\, y\, (\lambda u\, v.\, z)\, f) = y\, (x\, y)\, z\},\ \{x\})$.

Proof. 0. ($\Leftarrow$). Then $D \equiv \lambda t_1\, t_2\, t_3.\, t_2\, (t_1\, t_3)\, \Omega$ is a β-solution for $\mathbf{S}$.
($\Rightarrow$). Let $D \equiv \lambda t_1\, t_2\, t_3.\, t_2\, (D_1\, t_1\, t_2\, t_3) \ldots (D_q\, t_1\, t_2\, t_3)$ a β-solution for $\mathbf{S}$. We have: $(D\, y\, (\lambda u\, v.\, z)\, f) =$
$z\, (D_3\, y\, (\lambda u\, v.\, z)\, f) \ldots (D_q\, y\, (\lambda u\, v.\, z)\, f)$, this implies $q = 2$.
Hence: $D\, y\, (\lambda a\, b.\, P_k\, (D\, y\, (\mathbf{U}^3_1\, \mathbf{I})\, f)\, a)\, (D\, y\, (\lambda u\, v.\, z)\, f) =$
$P_k\, (\mathbf{U}^3_1\, \mathbf{I}\, (D_1 \ldots)\, (D_2 \ldots))\, (D_1 \ldots) = P_k\, \mathbf{I}\, (D_1 \ldots) = y\, z$.
This implies $P_k\, \mathbf{I} \in$ SOL and $\{k\}(k)\downarrow$.
1. ($\Leftarrow$). Let $G \equiv \lambda u\, t_1\, t_2\, t_3.\, t_2\, (t_1\, (t_2\, \mathbf{I}\, \Omega\, u\, u\, t_1)\, t_3)\, \Omega$ and $D \equiv G\, G$. Then D is a β-solution for $\mathcal{P}$.
($\Rightarrow$). Let D a β-solution for $\mathcal{P}$, then
$D' \equiv \lambda y.\, D\, (\lambda a.\, y)$ is a β-solution for $\mathbf{S}$ (in 0). □

4. A necessary condition of β-solvability for SL-systems

An SL-system is said to be canonical iff each RHS variable occurs on the LHS, the RHS variables are pairwise distinct and there is no *garbage* in the LHS terms (4.0.0.0-2).

4.0. Definition. Let $\mathbf{S} = (\Gamma, X)$ be an SL-system.
0. We say that $\mathbf{S}$ is canonical iff the following conditions are satisfied (M has form $x\, \vec{M}$ (2.0)):
0.0. $\forall M = z \in \mathbf{S}\ \exists \alpha \in BT(M)\ [\mathrm{head}(M_\alpha) \equiv z]$.
0.1. The variables in right($\mathbf{S}$) are pairwise distinct.
0.2. $\forall M = z \in \mathbf{S}\ [FV(BT(M)) \subseteq (X \cup \{z\})]$.
1. A canonical version of $\mathbf{S}$ is a system $\mathbf{S}^+ = (\Gamma^+, X)$ s.t.

$\mathbb{S}^+$ is canonical and Γ^+ is obtained from Γ replacing $M = z \in \Gamma$ with $M'^* = z^*$ where:
$M' \equiv M[v := \Omega \mid v \in (FV(M) - (X \cup \{z\}))]$ and
$* \equiv [z := u]$ with u fresh variable. □

4.1. Remark. Let $\mathbb{S} = (\Gamma, X)$ be an SL-system and $\mathbb{T}$ be a sms theory. Then $\mathbb{S}$ is $\mathbb{T}$-solvable iff there exists a canonical version $\mathbb{S}^+$ of $\mathbb{S}$ s.t. $\mathbb{S}^+$ is $\mathbb{T}$-solvable. Moreover, up to redenomination of the names of the free variables, there is at most one canonical version of $\mathbb{S}$. Hence it is not restrictive to consider only canonical SL-systems. □

4.2. Example. Let $\mathbb{S} = (\{x\, x\, (x\, (\lambda a.\, z)) = z,$
$x\, x\, (x\, (x\, z)) = z,\ x\, v\, (\lambda a\, b.\, z) = z\}, \{x\})$.
Then $\mathbb{S}$ is β-solvable iff $\mathbb{S}^+$ is β-solvable, where:
$\mathbb{S}^+ = (\{x\, x\, (x\, (\lambda a.\, u)) = u,\ x\, x\, (x\, (x\, y)) = y,$
$x\, \Omega\, (\lambda a\, b.\, z) = z\}, \{x\})$ ($\mathbb{S}^+$ is a canonical version of $\mathbb{S}$). □

Even for an HSL-system distinction ([CDR 78], [Bar 84, 10.4.7]) (4.5) is not a sufficient condition for β-solvabability.

4.3. Example. Let $\mathbb{S} = (\{x\, (\lambda a\, b.\, z) = z,$
$x\, (\lambda a.\, u) = u\}, \{x\})$ be a canonical HSL-system.
Then left($\mathbb{S}$) is distinct, but $\mathbb{S}$ is not β-solvable because the order of $(\lambda a.\, u)$ is too small. □

If $\mathbb{S}$ is a β-solvable SL-system and $M = z \in \mathbb{S}$ and $\mathrm{head}(M_\alpha) \equiv z$ then $\mathrm{ord}(M_\alpha)$ cannot be *too small.*

4.4. Lemma. Let $\mathbb{S} = (\Gamma, X)$ be a canonical SL-system.
If $\mathbb{S}$ is β-solvable then: $\exists\ \alpha$ usf and agt for left($\mathbb{S}$) s.t.
$\forall M \in \mathrm{left}(\mathbb{S})\ [\mathrm{head}(M_\alpha) \in \mathrm{right}(\mathbb{S}) \Rightarrow$
$[\deg(M_\alpha) = 0$ and $\mathrm{ord}(M_\alpha) = \max\{\mathrm{ord}(L_\alpha) \mid L \in \mathrm{left}(\mathbb{S})\}]]$.
Proof. If Card(left($\mathbb{S}$)) = 1 trivial. Let Card(left($\mathbb{S}$)) > 1.
As in the proof of [Bar 84, 14.4.13] we can prove that there exists α usf and agt for left($\mathbb{S}$). Let D[ ] a β-solution for $\mathbb{S}$. Hence $\forall\ M = z \in \mathbb{S}$ $D[M] = z$. Consider the standard reduction (* is a suitable substitution) $\sigma : D[M] \twoheadrightarrow M_\alpha{}^* \vec{H}_M \twoheadrightarrow z$, where $\alpha \in$ Seq is the first node usf and agt for left($\mathbb{S}$) that comes on the head during the standard reduction σ and $M = z \in \mathbb{S}$ (note that α does not depend on the choice of $M = z \in \mathbb{S}$). Because α is the first useful node that comes on the head we have $\forall M, N \in \mathrm{left}(\mathbb{S})\ |\vec{H}_M| = |\vec{H}_N|$. Hence $\forall M \in \mathrm{left}(\mathbb{S})\ |\vec{H}_M| \geq \max\{\mathrm{ord}(Q_\alpha) \mid Q \in \mathrm{left}(\mathbb{S})\}$.

Suppose that $\mathrm{head}(M_\alpha) \in \mathrm{right}(\mathbb{S})$. Then $\mathrm{ord}(M_\alpha) = |\vec{H}_M| \geq \max\{\mathrm{ord}(L_\alpha) \mid L \in \mathrm{left}(\mathbb{S})\}$ and $\deg(M_\alpha) = 0$. □

The β-solvability of an SL-system depends also on the order of the LHS terms at a useful node (4.5,6).

4.5. Definition. Let $Z \subset_f \mathrm{Var}$ and $\mathfrak{F} \subset_f \Lambda$. We say that $\mathfrak{F}$ is Z-distinct iff the following conditions are satisfied:

0. If $\mathrm{Card}(\mathfrak{F}) = 1$ then
   $[Z \neq \varnothing \Rightarrow \exists\ \alpha$ usf and agt for $\mathfrak{F}\ \forall M \in \mathfrak{F}$
   $[\mathrm{head}(M_\alpha) \in Z$ and $\deg(M_\alpha) = 0]]$;
1. If $\mathrm{Card}(\mathfrak{F}) > 1$ then $\exists\ \alpha$ usf and agt for $\mathfrak{F}$ s.t.:

1.0. $\forall M \in \mathfrak{F}\ [\mathrm{head}(M_\alpha) \in Z \Rightarrow [\deg(M_\alpha) = 0$ and $\mathrm{ord}(M_\alpha) = \max\{\mathrm{ord}(L_\alpha) \mid L \in \mathfrak{F}\}]]$;

1.1. $\forall P \in \mathfrak{F}/\sim_\alpha$ [P is Z-distinct]. □

Note that the $\varnothing$-distinction is the distinction introduced in [CDR 78] (also in [Bar 84, 10.4.7]).

4.6. Lemma. Let $\mathbb{S} = (\Gamma, X)$ be a canonical SL-system.
If $\mathbb{S}$ is β-solvable then left($\mathbb{S}$) is right($\mathbb{S}$)-distinct.
(Note that right($\mathbb{S}$) $\subset_f$ Var.)
Proof. By induction on Card($\mathbb{S}$) and using 4.4. □

If an SL-system is β-solvable then any proper initial part of an LHS term can be *distinguished* from an LHS term (4.7.1, 4.8). Definition 4.7 is inspired from [BT 91, 4.1.4].

4.7. Definition. Let $\mathbb{S} = (\Gamma, X)$ be an SL-system.
We define:

0. $\mathrm{prefix}(\mathbb{S}) = \{\langle x, M_1, \ldots M_m, \Omega, z\rangle \mid x\, M_1 \ldots M_m = z \in \mathbb{S}\} \cup \{\langle x, M_1, \ldots M_m, z, \Omega\rangle \mid x\, M_1 \ldots M_{m+k} = z \in \mathbb{S}$ and $k > 0\}$.
1. $\mathbb{S}$ is said to be PFR ($\mathbb{S}$ satisfies the prefix rule) iff prefix($\mathbb{S}$) is $\varnothing$-distinct (4.5). □

4.8. Lemma. Let $\mathbb{S} = (\Gamma, X)$ be a canonical SL-system and $\mathbb{T}$ be a sms theory. If $\mathbb{S}$ is $\mathbb{T}$-solvable then $\mathbb{S}$ is PFR.
Proof. Let D[ ] a $\mathbb{T}$-solution for $\mathbb{S}$ and $\mathrm{prefix}^*(\mathbb{S}) = \{x\, M_1 \ldots M_m \mid x\, M_1 \ldots M_{m+k} = z \in \mathbb{S}$ and $k > 0\}$. Note that $\forall M \in (\mathrm{left}(\mathbb{S}) \cup \mathrm{prefix}^*(\mathbb{S}))\ D[M] \in$ SOL. If $\mathbb{S}$ is not PFR then prefix($\mathbb{S}$) is not $\varnothing$-distinct (4.7.1), hence there are $M \in \mathrm{left}(\mathbb{S})$ and $N \in \mathrm{prefix}^*(\mathbb{S})$ s.t. ind(left($\mathbb{S}$) $\cup$ prefix*($\mathbb{S}$), M, N) (1.1.13). Then, by [BT 91, 3.4.0], ind(D[left($\mathbb{S}$) $\cup$ prefix*($\mathbb{S}$)], D[M], D[N]). Hence

head(D[M]) ≡ head(D[N]). This is absurd because N cannot be an initial subterm of M and $\mathcal{S}$ is canonical. □

From 4.1,6,8 we get a necessary condition of β-solvability for SL-systems (4.9).

4.9. Theorem. Let $\mathcal{S} = (\Gamma, X)$ be an SL-system. If $\mathcal{S}$ is β-solvable then there exists a canonical version $\mathcal{S}^*$ of $\mathcal{S}$ s.t. $\mathcal{S}^*$ is PFR and left($\mathcal{S}^*$) is right($\mathcal{S}^*$)-distinct. □

4.10. Example. 0. Let $\mathcal{S}$ as in 4.3. Then $\mathcal{S}$ is not β-solvable because left($\mathcal{S}$) is not right($\mathcal{S}$)-distinct.

1. $\mathcal{S} = (\{x\ (\lambda a.\ z) = z,\ x\ \Omega\ y = y\},\ \{x\})$ is not PFR, hence it is not β-solvable. □

Of course (by 3.1) the conditions in 4.9 are not sufficient.

4.11. Counter-example. $\mathcal{S} = (\{x\ (\lambda a.\ a\ z) = z,\ x\ (\lambda a.\ a\ (a\ u)) = u,\ x\ (\lambda a.\ a\ (a\ (a\ \Omega\ y))) = y\},\ \{x\})$ is PFR and left($\mathcal{S}$) is right($\mathcal{S}$)-distinct, but it is not β-solvable. □

5. Regularity

If $\mathcal{S}$ is a canonical SL-system and $M = z \in \mathcal{S}$ and head(M_α) ≡ z then the smaller is the order of M_α the more cleverness we need to find a solution for $\mathcal{S}$ (if any). For this reason the β-solvability problem for SL-systems is, in general, undecidable. To get a decidable class of SL-systems in 5.0-3 we ask that the order of M_α is *large enough*. The functions in Q(X) (5.0) describe the *skeleton* of the λ-terms that will be substituted for the unknowns in an SL-system (7.0).

5.0. Notation. Let $X \subset_f$ Var. We define:
$Q(X) = \{e : X \times \{0, 1, 2, 3\} \to \mathbb{N} \mid \forall x \in X\ [1 \le e(x, 0) \le e(x, 1)$ and $e(x, 2) \ge 1]$ and $\forall x, x' \in X\ [e(x, 2) - e(x', 2) = e(x, 1) - e(x, 0) + e(x', 0) - e(x', 1) \Rightarrow x \equiv x']\}$. □

If $M = z \in \mathcal{S}$ and head(M_α) ≡ z then using the function rad (5.1.1) we can check if ord(M_α) is big enough (5.2.0.2) to allow the use of the Böhm-out technique in our systems solving algorithm.

5.1. Definition. Let $X \subset_f$ Var, $e \in Q(X)$, $\mathfrak{I} \subset_f \Lambda$, $M \in \mathfrak{I}$ and $\alpha \in$ Seq.

0. rad(X, e, $\mathfrak{I}$, M, α) =
   case
   $M \mid \alpha \uparrow$ **then** 0;
   head(M_α) ∈ X
   then e(head(M_α), 1) − deg(M_α) + ord(M_α);
   ∃ β < α head(M_β) ≡ head(M_α)
   then max {deg(N_β) | β ≤ α and head(N_β) ≡ head(M_α) and N ∈ $\mathfrak{I}$} + 1 − deg(M_α) + ord(M_α);
   ¬ ∃ β < α head(M_β) ≡ head(M_α) **then** ord(M_α);
   end.

1. rad(X, e, $\mathfrak{I}$, α) = max{rad(X, e, $\mathfrak{I}$, M, α) | M ∈ $\mathfrak{I}$}. □

Taking into account what our systems solving algorithm can do (Z, X, e)-distinction (5.2.1) yields β-solvability (whereas Z-distinction (4.5) does not).

5.2. Definition. Let $X, Z \subset_f$ Var, $\mathfrak{I} \subset_f \Lambda$, $e \in Q(X)$.

0. A node α ∈ Seq is said to be (Z, X, e)-safe in $\mathfrak{I}$ iff it satisfies the following conditions:
   0.0. [∃ M, N ∈ $\mathfrak{I}$ [head(M_α) ≡ x ∈ X and head(N_α) ∉ FV(N_α)] ⇒ [e(x, 2) > e(x, 1) − e(x, 0) + max {deg(Q_α) | head(Q_α) ∉ FV(Q_α) and Q ∈ $\mathfrak{I}$}]].
   0.1. ∀ β ≤ α ∀ M ∈ $\mathfrak{I}$ [head(M_β) ∈ X ⇒ deg(M_β) < e(head(M_β), 0)].
   0.2. ∀ M ∈ $\mathfrak{I}$ [head(M_α) ∈ Z ⇒ [deg(M_α) = 0 and ord(M_α) = rad(X, e, $\mathfrak{I}$, α)]].
1. We say that $\mathfrak{I}$ is (Z, X, e)-distinct iff the following conditions are satisfied:
   1.0. If Card($\mathfrak{I}$) = 1 then ∃ α usf and agt for $\mathfrak{I}$ s.t. [α is (Z, X, e)-safe in $\mathfrak{I}$ and ∀ M ∈ $\mathfrak{I}$ head(M_α) ∈ Z]
   1.1. If Card($\mathfrak{I}$) > 1 then ∃ α usf and agt for $\mathfrak{I}$ s.t. [α is (Z, X, e)-safe in $\mathfrak{I}$ and ∀ P ∈ $\mathfrak{I} / \sim_\alpha$ P is (Z, X, e)-distinct]. □

The more severe are the constraints we set on the code to be generated by the compiler the more cleverness is required from the compiler to satisfy these constraints. If we ask for too much cleverness such a compiler does not exist. This is the meaning of the undecidability of the β-solvability problem for SL-systems (3.1). To get a class of SL-systems for which the β-solvability problem is decidable we prevent the user from writing too severe constraints on the code to be generated by the compiler. This lead us to the definition of regular SL-systems.

An SL-system is said to be regular iff the following conditions are satisfied:

if Q is a proper subterm of an LHS term and the head of Q is an unknown then the degree of Q is not *too large* (5.3.0.0,1, 5.3.0.5);

if Q is a proper subterm of an LHS term and the head of Q is a RHS term then the order of Q is *large enough* (5.3.0.2,3,4, 5.3.0.5).

5.3. Definition. Let $\mathbb{S} = (\Gamma, X)$ be a system and $e \in Q(X)$.

0\. A canonical SL-system $\mathbb{S}$ is said to be e-regular iff whenever $\mathbb{S}$ is PFR (4.7.1) and left($\mathbb{S}$) is right($\mathbb{S}$)-distinct (4.5) the following conditions are satisfied:

0.0. $\forall x \in (X \cap \text{head}(\text{left}(\mathbb{S})))$
$[e(x, 1) = \min\{m \mid x\, M_1 \ldots M_m = z \in \mathbb{S}\}]$.

0.1. $\forall\, x\, M_1 \ldots M_m = z \in \mathbb{S}\ [M_{e(x,0)} \in \text{SOL}]$.

0.2. $\forall\, x\, M_1 \ldots M_m = z \in \mathbb{S}\ [\text{head}(M_{e(x,0)}) \equiv z \Rightarrow$
$e(x, 2) = e(x, 1) - m + \text{ord}(M_{e(x,0)})]$.

0.3. $\forall\, x\, M_1 \ldots M_m = z \in \mathbb{S}\ [\text{head}(M_{e(x,0)}) \notin FV(M_{e(x,0)})$
$\Rightarrow e(x, 2) \geq \text{ord}(M_{e(x,0)})]$.

0.4. $\forall\, x\, M_1 \ldots M_m = z \in \mathbb{S}\ [\text{head}(M_{e(x,0)}) \equiv x' \in X \Rightarrow$
$e(x, 2) \geq e(x', 1) - \deg(M_{e(x,0)}) + \text{ord}(M_{e(x,0)})]$.

0.5. Let (see 1.1.5)
$\mathfrak{F}_x = \{\langle\Omega, M_1, \ldots M_m\rangle \mid x\, M_1 \ldots M_m = z \in \mathbb{S}\}$,
$\mathfrak{G}_x = \{\langle z, M_1', \ldots M_m'\rangle \mid \exists\, k > 0$
$[x\, M_1 \ldots M_{m+k} = z \in \mathbb{S}$ and
$* = \{(\alpha, \lambda t.\, t\, \Omega_1 \ldots \Omega_k) \mid \alpha \in BT(x\, M_1 \ldots M_{m+k})$
and $\text{head}((x\, M_1 \ldots M_{m+k})_\alpha) \equiv z\}$ and
$(x\, M_1 \ldots M_{m+k})^* = x\, M_1' \ldots M_{m+k}']\}$. Then:

0.5.0. $\forall x \in (X \cap \text{head}(\text{left}(\mathbb{S})))$
$[\langle e(x, 0)\rangle$ is (right($\mathbb{S}$), X, e) -safe in $\mathfrak{F}_x$].

0.5.1. $\forall x \in (X \cap \text{head}(\text{left}(\mathbb{S})))$
$[(\mathfrak{F}_x \cup \mathfrak{G}_x)$ is (right($\mathbb{S}$), X, e)-distinct].

1\. A canonical SL-system $\mathbb{S} = (\Gamma, X)$ is said to be regular iff $\exists\, e \in Q(X)$ s.t. $\mathbb{S}$ is e-regular.

2\. An SL-system $\mathbb{S}$ is said to be regular iff whenever there exists a canonical version $\mathbb{S}^\#$ of $\mathbb{S}$ s.t. $\mathbb{S}^\#$ is PFR and left($\mathbb{S}^\#$) is right($\mathbb{S}^\#$)-distinct there exists also a canonical version $\mathbb{S}^+$ of $\mathbb{S}$ s.t. $\mathbb{S}^+$ is regular, PFR and left($\mathbb{S}^+$) is right($\mathbb{S}^+$)-distinct.

3\. Let $\mathbb{S} = (\Gamma, X)$ be a system with equations having form $x\, \vec{M} = \lambda\vec{a}.\, y\, \vec{Q}$, where $x \in X$ and $y \in (FV(\lambda\vec{a}.\, y\, \vec{Q}) - X)$. $\mathbb{S}$ is said to be regular iff $\mathbb{S}_\Omega$ is regular (see 1.3.6). □

5.4. Example. 0. The systems in 0.1-4, 2.2 and 4.2 are regular.

1\. $\mathbb{S} = (\{x\, (\lambda a\, b.\, z) = z,\ x\, (\lambda a\, b.\, a\, y)\, \Omega = y\}, \{x\})$ is regular.

2\. $\mathbb{S} = (\{x\, (\lambda a.\, a\, z) = z,\ x\, (\lambda a.\, a\, (a\, u)) = u\}, \{x\})$ is not regular (5.3.0.5.1 fails).

3\. A separability problem (2.1.0) ([CDR 78], [Bar 84, 10.4.4]) is a regular SL-system.

4\. All the systems studied in [BT 87, 91] are regular systems. However there are regular systems (e.g. those in 0.2-4, 5.4.1, 8.2) that are not in the classes studied in [BT 87, 91], [BPT 88] or [PT 90]. On the other hand not all the systems studied in [BPT 88] or [PT 90] are regular systems because in those classes there are systems having proper subterms (of an LHS term) with head an unknown and with degree *too large*. □

6. Regular HSL-Systems

In 6.0 is a sufficient condition for the existence of a common β-left-inverse for $\mathfrak{F} \subset_f \Lambda^o$ which yields a characterization of the β-solvability problem for regular HSL-systems (6.1).

6.0. Lemma. Let $\mathbb{S} = (\Gamma, \{x\})$ be a canonical HSL-system with equations having form $x\, M = z$.
If left($\mathbb{S}$) is (right($\mathbb{S}$), ∅, ∅)-distinct then $\mathbb{S}$ is β-solvable.

Proof. (*sketch*) The proof can be carried out by induction on Card($\mathbb{S}$) endowing the Böhm-out technique with a *local memory* (see 2.2) and using only safe nodes (which is possible because left($\mathbb{S}$) is (right($\mathbb{S}$), ∅, ∅)-distinct). □

The β-solvability problem for regular HSL-systems is decidable.

6.1. Theorem. Let $\mathbb{S} = (\Gamma, \{x\})$ be a canonical and regular HSL-system. Then $\mathbb{S}$ is β-solvable iff $\mathbb{S}$ is PFR and left($\mathbb{S}$) is right($\mathbb{S}$)-distinct.

Proof. (*sketch*) (⇒). By 4.9. (⇐). By induction on Card($\{\deg(M) \mid M \in \text{left}(\mathbb{S})\}$) and using 6.0. □

6.2. Example. 0. Let $\mathbb{S} = (\{x\, (\lambda a\, b.\, z) = z,$
$x\, (\lambda a\, b.\, a\, y)\, \Omega = y\}, \{x\})$.
$\mathbb{S}$ is regular, PFR and left($\mathbb{S}$) is right($\mathbb{S}$)-distinct.
Hence by 6.1 $\mathbb{S}$ is β-solvable. A possible β-solution for $\mathbb{S}$ is: $D \equiv \lambda t.\, t\, (\lambda a.\, t\, \mathbf{I}\, \Omega)\, \Omega$.

1\. Let $\mathbb{S}' = (\{x\, (\lambda a\, b.\, z) = z,\ x\, (\lambda a\, b\, c.\, y)\, \Omega = y\}, \{x\})$.
$\mathbb{S}'$ is regular, PFR and left($\mathbb{S}'$) is right($\mathbb{S}'$)-distinct.
Hence by 6.1 $\mathbb{S}'$ is β-solvable. A possible β-solution for $\mathbb{S}'$ is: $D \equiv \lambda t.\, t\, \Omega\, \Omega$.

2\. The system $\mathbb{S}'' = (\{x\, (\lambda a.\, a\, z) = z,\ x\, (\lambda a.\, a\, (a\, u)) = u,$
$x\, (\lambda a.\, a\, (a\, (a\, \mathbf{I}\, y))) = y\}, \{x\})$ is not regular (see

5.4.2, 4.11 and 3.1), but it is β-solvable.

A possible β-solution for $\mathbb{S}''$ is: $G \equiv \lambda t.\, t\, \mathbf{I}$. □

7. Regular SL-systems

We characterize the β-solvability for regular SL-systems. To construct a β-solution for such systems we introduce a particular context.

7.0. Notation. Let $\{\vec{x}\} = \{x_1, \ldots x_n\} \subset_f$ Var, and $e \in Q(\{\vec{x}\})$.

0. $D_{x,e} \equiv \lambda t_1 \ldots t_{e(x,0)} \ldots t_{e(x,1)}.\, t_{e(x,0)}$
   $(t_{e(x,0)}\, \Omega_1 \ldots \Omega_{e(x,3)}\, t_1 \ldots t_{e(x,0)} \ldots t_{e(x,1)})$
   $\Omega_2 \ldots \Omega_{e(x,2)}$.
1. $D_{\vec{x},e}[\ ] \equiv (\lambda x_1 \ldots x_n.\, [\ ])\, D_{x_1,e} \ldots D_{x_n,e}$. □

We break self-application finding a common solution for an infinite class of systems.

7.1. Lemma. Let $\mathbb{S} = (\Gamma, \{\vec{x}, u\})$ be a canonical and regular SL-system with equations having form $u\,\vec{M} = z$ where $u \notin FV(\vec{M})$. If $\mathbb{S}$ is PFR and left($\mathbb{S}$) is right($\mathbb{S}$)-distinct then $\exists e \in Q(\{\vec{x}\})\, \exists F \in \Lambda^o\ \forall G[\ ] \sqsupseteq D_{\vec{x},e}[\ ]$
$\forall u\, M_1 \ldots M_m = z \in \mathbb{S}\quad F\, G[M_1] \ldots G[M_m] = z$.

Proof. (*sketch*) Let $e' \in Q(\{\vec{x}, u\})$ s.t. $\mathbb{S}$ is e'-regular and $e \in Q(\{\vec{x}\})$ be the restriction of e' to $\{\vec{x}\}$. In this situation it is possible to show that $\mathbb{S}_1 = (\{u\, D_{\vec{x},e}[M_1] \ldots D_{\vec{x},e}[M_m] = z \mid u\, M_1 \ldots M_m = z \in \mathbb{S}\}, \{u\})$ is regular and PFR and left($\mathbb{S}_1$) is right($\mathbb{S}_1$)-distinct. Hence, by 6.1, $\mathbb{S}_1$ is β-solvable. Let F a β-solution for $\mathbb{S}_1$. Let $G[\ ] \sqsupseteq D_{\vec{x},e}[\ ]$. Then:
$\forall u\, M_1 \ldots M_m = z \in \mathbb{S}\quad F\, G[M_1] \ldots G[M_m] \sqsupseteq$
$F\, D_{\vec{x},e}[M_1] \ldots D_{\vec{x},e}[M_m] = z$. □

The β-solvability problem for regular SL-systems is decidable (7.2,3).

7.2. Theorem. Let $\mathbb{S} = (\Gamma, X)$ be a canonical and regular SL-system.

Then $\mathbb{S}$ is β-solvable iff $\mathbb{S}$ is PFR and left($\mathbb{S}$) is right($\mathbb{S}$)-distinct.

Proof. (*sketch*) (⇒). By 4.9.

(⇐). Let $e \in Q(X)$ s.t.: 0. $\mathbb{S}$ is e-regular;

1. $d = \max\{\max\{\deg(M_{e(x,0)}) \mid x\, M_1 \ldots M_{e(x,1)} \in$ left($\mathbb{S}$)$\}, \max\{e(x,2) \mid x \in X\}, \max\{e(x,1) \mid x \in X\}\}$;
2. $\forall x \in X\ \ e(x,3) = d + 1 + \max\{\mathrm{ord}(M_{e(x,0)}) - \deg(M_{e(x,0)}) \mid x\, M_1 \ldots M_m \in$ left($\mathbb{S}$)$\}$.

By 5.3 such an e exists. Let $x \in X = \{\vec{x}\} = \{x_1, \ldots x_n\}$.
$\forall x \in X$ let $\mathbb{S}(x) = (\{u\, M_1 \ldots M_m = z \mid x\, M_1 \ldots M_m = z \in \mathbb{S}\}, \{u, \vec{x}\})$ (u fresh). Then $\forall x \in X$ $\mathbb{S}(x)$ is regular and PFR and left($\mathbb{S}(x)$) is right($\mathbb{S}(x)$)-distinct. Hence by 7.1
$\forall x \in X\ \exists F_x \in \Lambda^o\ \forall C[\ ] \sqsupseteq D_{\vec{x},e}[\ ]$
$\forall u\, M_1 \ldots M_m = z \in \mathbb{S}(x)\quad F_x\, C[M_1] \ldots C[M_m] = z$.
$\forall x \in X$ define:
$A(x) = \{(i, d + 1 + \mathrm{ord}(M_{e(x,0)}) - \deg(M_{e(x,0)})) \mid$
$\quad x\, M_1 \ldots M_m \in$ left($\mathbb{S}$) and
$\quad M_{e(x,0)} = \lambda\vec{b}.\, b_i\, \vec{Q}$ and $b_i \in \{\vec{b}\}\}$,
$B(x) = \{(e(x',0) + \mathrm{ord}(M_{e(x,0)}) - \deg(M_{e(x,0)}),$
$\quad d + 1 + \mathrm{ord}(M_{e(x,0)}) - \deg(M_{e(x,0)}) + e(x',1) -$
$\quad e(x',2)) \mid x\, M_1 \ldots M_m \in$ left($\mathbb{S}$) and
$\quad M_{e(x,0)} = \lambda\vec{b}.\, x'\, \vec{Q}$ and $x' \in X\}$.
Note that $\forall i \in \mathbb{N}\ [(i,i) \notin A(x)$ and $(i,i) \notin B(x)]$ and (by 5.3.0.5.0 and 5.2.0.0) $\forall x \in X\ [A(x) \cap B(x) = \emptyset]$.
Taking into account 5.0 for all $x \in X$ we can define:
$1(x) \equiv [a_{x,i,i} := \mathbf{P}_d \mid \exists k \in \mathbb{N}\ (i,k) \in A(x)]$,
$2(x) \equiv [a_{x,i,k} :=$
$\quad \lambda t_1 \ldots t_{2(d-k)+1+e(x,1)+e(x,2)+e(x,3)}.\, F_x$
$\quad t_{d+1-k+e(x,3)} \ldots t_{d-k+e(x,1)+e(x,3)} \mid (i,k) \in A(x)]$,
$3(x) \equiv [a_{x,i,i} := \mathbf{P}_d \mid \exists k \in \mathbb{N}\ (i,k) \in B(x)]$,
$4(x) \equiv [a_{x,i,k} :=$
$\quad \lambda t_1 \ldots t_{2(d-k)+1+e(x,1)+e(x,2)+e(x,3)}.\, F_x$
$\quad t_{d+1-k+e(x,3)} \ldots t_{d-k+e(x,1)+e(x,3)} \mid (i,k) \in B(x)]$,
$D(x,e) \equiv \lambda t_1 \ldots t_{e(x,0)} \ldots t_{e(x,1)}.\, t_{e(x,0)}$
$(t_{e(x,0)}\, a_{x,1,1} \ldots a_{x,1,e(x,3)}\, t_1 \ldots t_{e(x,0)} \ldots t_{e(x,1)})$
$\ldots\ldots\ldots\ldots\ldots\ldots$
$(t_{e(x,0)}\, a_{x,e(x,2),1} \ldots a_{x,e(x,2),e(x,3)}\, t_1 \ldots t_{e(x,0)} \ldots t_{e(x,1)})$.
$\forall x \in X$ define: $G_x \equiv D(x,e)^{1(x)\, 2(x)\, 3(x)\, 4(x)}$,
$G[\ ] \equiv (\lambda x_1 \ldots x_n.\, [\ ])\, G_{x_1} \ldots G_{x_n}$.
Then $G[\ ]$ is a β-solution for $\mathbb{S}$. □

7.3. Corollary. Let $\mathbb{S} = (\Gamma, X)$ be a regular SL-system. Then $\mathbb{S}$ is β-solvable iff there exists a canonical version $\mathbb{S}^*$ of $\mathbb{S}$ s.t. $\mathbb{S}^*$ is PFR and left($\mathbb{S}^*$) is right($\mathbb{S}^*$)-distinct.

Proof. By 4.1, 4.9 and 7.2. □

7.4. Example. 0. Let $\mathbb{S} = (\{x\, y\, \Omega\, (\lambda a.\, x\, \Omega\, \Omega) = y,\ x\, \Omega\, \Omega\, (\lambda a\, b.\, z) = z\}, \{x\})$. $\mathbb{S}$ is regular and left($\mathbb{S}$) is right($\mathbb{S}$)-distinct. Hence by 7.2 $\mathbb{S}$ is β-solvable. A possible β-solution for $\mathbb{S}$ is: $D \equiv \lambda y\, t_1\, t_2.\, t_2\, t_1\, (U_1^3\, y)$.

1. The system in 4.2 is β-solvable. □

8. Applications

Using SL-systems it is possible to solve systems like those in Ex. 0.1-4.

8.0. Theorem. Let $\mathbf{S} = (\Gamma_1 \cup \Gamma_2, \{\vec{x}\})$ be a system and $\vec{y} \in$ (Var - $\{\vec{x}\}$) s.t.: (see 1.1.14)

0. For each $M \in$ left(S) there exists $Q \in \Lambda$ s.t. $M = (\lambda\vec{x}.\, Q)\,(\vec{x} : \vec{y})$ and $\vec{y} \notin FV(Q)$.
1. Each equation in Γ_1 has form $x\,\vec{M} = y\,(\vec{x} : \vec{y})\,\vec{z}$, where $x \in \{\vec{x}\}$, $y \in \{\vec{y}\}$, $\vec{z} \notin \{\vec{x}, \vec{y}\}$ and the variables in head(right(Γ_1)) are pairwise distinct.
2. Each equation in Γ_2 has form $x\,\vec{M} = z$, where $x \in \{\vec{x}\}$ and $z \notin \{\vec{x}, \vec{y}\}$.
3. $\mathbf{S}^{\#} = (\{x\,\vec{M} = y \mid x\,\vec{M} = y\,(\vec{x} : \vec{y})\,\vec{z} \in \Gamma_1\} \cup$
   $\{u_{y\,z}\,\vec{M} = z \mid x\,\vec{M} = y\,(\vec{x} : \vec{y})\,\vec{z} \in \Gamma_1, z \in \{\vec{z}\},$
   $u_{y\,z} \in \{\vec{u}\}\} \cup \Gamma_2, \{\vec{x}, \vec{u}\})$ is regular.

Then:

0. $\mathbf{S}$ is β-solvable iff there exists a canonical version $\mathbf{S}^+$ of $\mathbf{S}^{\#}$ s.t.: $\mathbf{S}^+$ is PFR and left($\mathbf{S}^+$) is right($\mathbf{S}^+$)-distinct.
1. If $\mathbf{S}$ is β-solvable then it has a β-solution having nf.

Proof. (*sketch*) Transforming $\mathbf{S}$ into an SL-system (note that $\mathbf{S}^{\#}$ is a step in this direction). □

8.1. Example. By 8.0 the systems in 0.1-4 are β-solvable.□

8.2. Example. It is well known that many interesting data structures can be represented using (heterogeneous) term algebras (e.g. [BB 85]).

Let $\mathfrak{A}_j = <\{A_{j,\,1}, \ldots A_{j,\,n(j)}\},$
$\{g_{j,\,i} : A_{j,\,k(j,\,1)} \times \ldots \times A_{j,\,k(j,\,a(j,\,i))} \rightarrow A_{j,\,b(j,\,i)} \mid$
$i = 1, \ldots m(j)\}>$ be term algebras ($j = 1, 2$).

A partial recursive function f from $\mathfrak{A}_1$ to $\mathfrak{A}_2$ can be represented in the λ-calculus solving the system

$\mathbf{S} = (\Gamma_1 \cup \Gamma_2 \cup \Gamma_3, \{f, \vec{g}, \vec{p}, \vec{d}\})$, where:

(left-invertibility of the constructors)

$\Gamma_1 = \{p_{j,\,i,\,h}\,\vec{y}\,(g_{j,\,i}\,\vec{y}\,z_1 \ldots z_{k(j,\,a(j,\,i))}) = z_h \mid$
$i = 1, \ldots m(j)$ and $h = 1, \ldots k(j, a(j, i))$ and $j = 1, 2\}$,

(recognizability of the constructors)

$\Gamma_2 = \{d_j\,\vec{y}\,(g_{j,\,i}\,\vec{y}\,z_1 \ldots z_{k(j,\,a(j,\,i))}) = y_{0,\,j,\,i} \mid$
$j = 1, 2$ and $i = 1, \ldots m(j)\}$,

(specification of the function f)

$\Gamma_3 = \{f\,\vec{y}\,(g_{1,\,i}\,\vec{y}\,z_1 \ldots z_{k(1,\,a(1,\,i))}) =$
$y_{1,\,i}\,(f, \vec{g}, \vec{p}, \vec{d} : \vec{y})\,z_1 \ldots z_{k(1,\,a(1,\,i))} \mid i = 1, \ldots m(1)\}$,

$\vec{g} \equiv g_{1,\,1}, \ldots g_{2,\,m(2)}$,
$\vec{p} \equiv p_{1,\,1,\,1}, \ldots p_{2,\,m(2),\,k(2,\,a(2,\,m(2)))}$, $\vec{d} \equiv d_1, d_2$,
$\vec{y} \equiv y_{0,\,1,\,1}, \ldots y_{0,\,2,\,m(2)}\,y_{1,\,1}, \ldots y_{1,\,m(1)}$.

The equations in $\Gamma_1 \cup \Gamma_2$ describe the data structures and the equations in Γ_3 describe the function f (a set of functions if we consider types). By 8.0 $\mathbf{S}$ is β-solvable. A solution $H[\ ] \equiv (\lambda f\,\vec{g}\,\vec{p}\,\vec{d}.\ [\ \ ])\ F\,\vec{G}\,\vec{P}\,\vec{D}$ for $\mathbf{S}$ yields the wanted representation for program and data structures ($F\,\vec{y}$ for the program for the function f, etc). Any partial recursive function can be specified (and hence represented) replacing $y_{1,\,1}, \ldots y_{1,\,m(1)}$ with suitable combinators (see 0.0,1). Of course it is possible to add equations to $\mathbf{S}$ for example to choose a representation with some particular property (e.g. as we did in 0.2-4) or to synthesize more than one program at the same time, etc. Consider $\mathbf{S}' = (\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \{f\,\vec{y}\,(\lambda a\ b.\ z) = z\}, \{f, \vec{g}, \vec{p}, \vec{d}\})$. By 8.0 the system $\mathbf{S}'$ is β-solvable. This means that any partial recursive function from $\mathfrak{A}_1$ to $\mathfrak{A}_2$ can be represented with a λ-term having degree 2. The system $\mathbf{S}$ ($\mathbf{S}'$) cannot be transformed in an X-separability problem (2.1.1) because of the presence of the equations in Γ_1 ($\Gamma_1 \cup \{f\,\vec{y}\,(\lambda a\ b.\ z) = z\}$) and, as in 0.3, cannot be solved with the methods in [BB 85]. □

9. Complexity Analysis

Testing the regularity of an SL-system $\mathbf{S}$, testing the β-solvability of $\mathbf{S}$ and constructing a β-solution for $\mathbf{S}$ are Polynomial Time task (9.0.0-2). On the other hand we can easily define a class of non regular SL-systems for which the β-solvability problem is NP-complete (9.2).

We only consider λ-terms having a finite Böhm-tree. Moreover we assume of having an oracle that given a λ-term M computes BT(M) (or equivalently we assume that λ-terms are given in $\beta\Omega$-normal form).

9.0. Proposition. Let $\mathbf{S} = (\Gamma, X)$ be an SL-system.

0. To test if there exists a canonical version $\mathbf{S}^*$ of $\mathbf{S}$ s.t. $\mathbf{S}^*$ is PFR and left($\mathbf{S}^*$) is right($\mathbf{S}^*$)-distinct (4.7.1 and 4.5) is a Polynomial Time task.
1. To test if $\mathbf{S}$ is regular (5.3.1) is a Polynomial Time task.
2. The solution in 7.2 can be constructed in Polynomial Time.

Proof. (*sketch*) 0,1. All the algorithms involved are Polynomial Time.
2. The construction in 7.2 imitates the checking of the conditions $\mathbf{S}$ is regular, PFR and left($\mathbf{S}$) is right($\mathbf{S}$)-distinct.□

To solve systems like those in 8.0 is a Polynomial Time task.

9.1. Proposition. Let $\mathbf{S} = (\Gamma_1 \cup \Gamma_2, \{\vec{x}\})$ satisfying the hypotheses of 8.0. To test if $\mathbf{S}$ is β-solvable and to construct a β-solution for $\mathbf{S}$ (if any) are Polynomial Time task.

Proof. (*sketch*) $\mathbf{S}$ can be transformed into an SL-system in Polynomial Time. Then the thesis follows from 9.0. □

Even simple looking extensions of the class of the regular SL-systems yield an NP-complete β-solvability problem.

9.2. Proposition. There is a class of SL-systems for which the β-solvability problem is NP-complete.

Proof. We codify the satisfability problem for propositional formulas with SL-systems. Let PropForm (PropVar) the set of Propositional Formulas (Variables).

Let L : PropForm → Λ defined as follows: $L(x) = x$,
$L(\neg A) = L(A)\, \mathbf{U}^2_2\, \mathbf{U}^2_1$, $L(A \vee B) = L(A)\, L(B)\, \mathbf{U}^2_2$,
$L(A \wedge B) = L(A)\, \mathbf{U}^2_1\, L(B)$
(we are representing true (T) with $\mathbf{U}^2_1$ and false (F) with $\mathbf{U}^2_2$).
Let A ∈ PropForm s.t. $FV(A) = \{x_1, \ldots x_n\}$. We define:
$\mathrm{Transl}(A) = (\{L(A)\ z\ \Omega = z\} \cup \{x_i\ z\ z = z \mid i = 1, \ldots n\}, \{x_1, \ldots x_n\})$.
Transl(A) is an SL-system and is β-solvable iff A is satisfiable. In fact:
(⇒). Let $D[\ ] = (\lambda x_1 \ldots x_n.[\ ])\, D_1 \ldots D_n$ a β-solution for Transl(A). Then $\forall i \in \{1, \ldots n\}$ $[D_i = \mathbf{U}^2_1$ or $D_i = \mathbf{U}^2_2]$ and $D[L(A)] = \mathbf{U}^2_1$. Choosing $* \equiv [x_i :=$ **if** $D_i = \mathbf{U}^2_1$ **then** T **else** F $\mid i = 1, \ldots n]$ we have $A^* = T$.
(⇐). Let $*$ s.t. $A^* = T$. $\forall i \in \{1, \ldots n\}$ define:
$D_i \equiv$ **if** $x_i{}^* = T$ **then** $\mathbf{U}^2_1$ **else** $\mathbf{U}^2_2$. Then a β-solution for Transl(A) is $D[\ ] \equiv (\lambda x_1 \ldots x_n.[\ ])\, D_1 \ldots D_n$.
Define: SLP = {Transl(A) | A ∈ PropForm}. Then the β-solvability problem for the systems in SLP is NP-complete. Note that the systems in SLP are not regular. □

10. Conclusions

Though the β-solvability problem for SL-systems (2.0) is undecidable (3.1) there is an interesting class of SL-systems (5.3) definable in Polynomial Time (9.0.1) for which the β-solvability problem is decidable (7.3) in Polynomial Time (9.0.0,2). This class yields (8.0, 9.1) an equational programming language in which constraints (e.g. like those in 0.2-4, 8.2) on the code generated by the compiler can be specified by the user, (properties of) data structures can be described in an abstract way (e.g. as in 0.2-4, 8.2), the λ-terms representing the programs have normal form and the inverse functions of the constructors (of a data structure) are always *one shot* (e.g. as in 0.3, 8.2). To widen the language introduced seems to be next step.

Acknowledgements

I am grateful to Rick Statman for the many helpful discussions on the topics of this paper and for his valuable suggestions. Without some of the questions I received from Rick Statman Section 9 would not be here.
I am also grateful to Mariangiola Dezani for her questions and comments about a preliminary version of this paper.
I thank the referees for their useful remarks and suggestions.

References

[Bar 84] Barendregt, H.P., *The lambda calculus*, North Holland, 1984

[BB 85] Böhm, C., Berarducci, A. *Automatic Synthesis of Typed Λ-Programs on Term Algebras*, Theor. Comp. Science., 39, 1985, 135-154

[BD 74] Böhm, C. and Dezani-Ciancaglini, M., *Combinatorial problems, combinator equations and normal forms*, LNCS 14, 1974, 185-199

[Ber 83] Berarducci, A., *Program. funz. e rappresen. in alcuni sistemi di logica combinatoria*, Tesi di Laurea in Matematica, 1983 (supervisor C. Böhm)

[BPT 88] C. Böhm, A. Piperno, E. Tronci, *Solving equations in λ-calculus*, proceedings of "Logic Colloquium 88", Padova, North-Holland 1989.

[BT 87] Böhm, C. and Tronci, E., *X-Separability and Left-Invertibility in Lambda-Calculus*, LICS 87, Computer Soc.of the IEEE, pp.320-328

[BT 91] Böhm, C., Tronci, E., *About systems of equations, X-separability and left-invertibility in the λ-calculus*, Infor. and Comp. 90, 1-32 (1991)

[CDR 78] Coppo, M., Dezani-Ciancaglini, M. and Ronchi della Rocca, S., *(Semi-) separability of finite sets of terms in Scott's D_∞ models of the λ-calculus*, LNCS 62, 1978, 142-164

[MZ 83] Margaria, I., Zacchi, M., *Right and Left Invertibility in λ-β-calculus*, RAIRO Th.Inf. 17, 1983, 71-88

[O'D 85] O'Donnell, Michael J., *Equational Logic as a Programming Language*, MIT Press Series in the Foundation of Computation, 1985

[PT 90] Piperno, A., Tronci, E. *Regular systems of equations in the λ-calculus*, Intern. Jour. of Foundations of Computer Science, Vol.1, N.3 (1990) 325-339

[Sta 89] Statman, R., *On sets of solutions to combinator equations*, Theoretical Computer Science 66 (1989) 99 - 104

[Sta 87] Statman, R., Private communication