ELSEVIER Theoretical Computer Science 160 (1996) 185–216

Theoretical Computer Science

Fundamental Study

Equational programming in λ-calculus via SL-systems. Part 2[☆]

Enrico Tronci[*,1]

Dipartimento di Matematica Pura ed Applicata, Università dell'Aquila, Via Vetoio, Coppito, 67100 L'Aquila, Italy

Received January 1992; revised July 1995
Communicated by C. Böhm

Abstract

A system of equations in the λ-calculus is a set of formulas of Λ (the equations) together with a finite set of variables of Λ (the unknowns). A system $\mathscr{S}$ is said to be β-solvable ($\beta\eta$-solvable) iff there exists a simultaneous substitution with closed λ-terms for the unknowns that makes the equations of $\mathscr{S}$ theorems in the theory β ($\beta\eta$). A system $\mathscr{S}$ can be viewed as a set of specifications (the equations) for a finite set of programs (the unknowns) whereas a solution for $\mathscr{S}$ yields executable codes for such programs.

A class $\mathfrak{S}$ of systems for which the solvability problem is effectively decidable defines an equational programming language and a system solving algorithm for $\mathfrak{S}$ defines a compiler for such language.

This leads us to consider separation-like systems (SL-systems), i.e. systems with equations having form $x\vec{M} = z$, where x is an unknown and z is a free variable which is not an unknown.

It is known that the β ($\beta\eta$)-solvability problem for SL-systems is undecidable.

Here we show that there is a class of SL-systems (NP-regular SL-systems) for which the β-solvability problem is NP-complete. Moreover, we show that any SL-system $\mathscr{S}$ can be transformed into an NP-regular SL-system $\mathscr{S}'$. This transformation consists of adding abstractions to the LHS occurrences of the RHS variables of $\mathscr{S}$. In this sense NP-regular SL-systems isolate the *source* of undecidability for SL-systems, namely: a shortage of abstractions on the LHS occurrences of the RHS variables.

NP-regular SL-systems yield an equational programming language in which unrestrained self-application is handled, constraints on executable code to be generated by the compiler can be specified by the user and (properties of) data structures can be described in an abstract way. However, existence of executable code satisfying a specification in such language is an

[☆] This paper is a revised version of [7, Part 2].

[*] Email: tronci@smaq20.univaq.it.

[1] This research has been partially supported by grant 203.01.50 from CNR, Italy.

SSDI 0304-3975(95)00106-9

NP-complete problem. This is the price we have to pay for allowing unrestrained self-application in our language.

Keywords: Systems of equations in the λ-calculus; λ-calculus; Equational programming; Functional programming; Automated synthesis of programs

Contents

0. Introduction

For a general introduction and motivations we refer the reader to [8, 0].

It is known [8, 4.1] that the β-solvability problem for SL-systems is, in general, undecidable. However [8, 5.2] for regular SL-systems the β-solvability problem is decidable in Polynomial Time. Moreover, [8, 5.2.4] any SL-system containing only a *limited amount* of self-application (quasi-regular SL-system) can be transformed into a regular SL-system. This transformation (called relaxation (see [8, 5.1])) consists of adding abstractions to the LHS occurrences of the RHS variables of an SL-system.

Here we strengthen such result showing (2.0) that there is a class of SL-systems (NP-regular SL-systems) for which the β-solvability problem is NP-complete and such that for any SL-system $\mathscr{S}$ there is an NP-regular SL-system $\mathscr{S}'$ s.t. $\mathscr{S}'$ is a relaxation of $\mathscr{S}$. This shows:

- that the only source of undecidability for SL-systems is a shortage of abstractions on the LHS occurrences of the RHS variables;
- self-application does not yield undecidability but the complexity of the β-solvability problem depends on how much self-application is present. Namely: for moderate self-application the β-solvability problem is Polynomial (regular SL-systems [8, 5.2.4]) whereas for unrestrained self-application (NP-regular SL-systems (2.0.4)) the β-solvability problem is NP-complete.

Regular SL-systems yield an equational programming language [8, 6.4]. A similar result holds for NP-regular SL-systems (3.0). However, compiling (i.e. finding executable code) will take, in general, exponential time. This is the price we have to pay for allowing an unrestrained presence of self-application in our equational programming language.

We show (2.5) that X-separability [4, 5, 8, 3.1] is an NP-complete problem. Moreover, via NP-regular SL-systems we give (2.2, 2.3) a new proof of decidability for X-separability. Our proof is shorter then (a full version of) the proof in [2] or [6].

We assume the reader familiar with [1] and [8] of which, unless otherwise stated, we use notation and conventions. In particular, we adopt [8, 2.3].

1. Summary

Section 2 gives our main result: There is an *interesting* class of SL-systems, NP-regular SL-systems, for which β-solvability is decidable and s.t. for any SL-system $\mathscr{S}$ there is a relaxation $\mathscr{S}'$ of $\mathscr{S}$ s.t. $\mathscr{S}'$ is a NP-regular SL-system. Moreover, we show that:

- the problem of deciding if a given SL-system is NP-regular is in NP;
- the problem of deciding if a given NP-regular SL-system has a β-solution is NP-complete;
- if an NP-regular SL-system has a β-solution then we can find it in exponential time.

Section 3 builds a compiler for an equational programming language which extends the one defined in [8, 6.4] by allowing unrestrained self-application. The price that we have to pay for such enhancement is NP-completeness of the β-solvability problem.

2. NP-regular SL-systems

As in [8] we will transform program specifications into SL-systems (see [8, 2.2, 3.0]). Since we are looking for a language allowing unrestrained self-application we start by studying the β-solvability problem for SL-systems allowing unrestrained self-application.

Though β-solvability for SL-systems is undecidable even when no self-application is present ([8, 4.1]) we show (2.0) that there is an *interesting* class of SL-systems (NP-regular SL-systems) allowing unrestrained self-application and for which the β-solvability problem is decidable. However, unlike regular SL-systems ([8, 5.2, 5.3]), NP-regular SL-systems have an NP-complete β-solvability problem.

Relaxation was defined in [8, 51]. It consists of adding abstractions to the LHS occurrences of the RHS variables of an SL-system.

Theorem 2.0 is our main result. It is the core of our system solving algorithm.

Theorems 2.0.0, 2.0.2 give a necessary condition (NP_OK_NEC) of β-solvability for SL-systems and show that the problem of deciding if for a given SL-system $\mathscr{S}$ NP_OK_NEC($\mathscr{S}$) = true holds is in NP.

Theorems 2.0.1, 2.0.3 give a sufficient condition (NP_OK_SUFF) of β-solvability for SL-systems and show that the problem of deciding if for a given SL-system $\mathscr{S}$ NP_OK_SUFF($\mathscr{S}$) = true holds is in NP. Moreover, (the proof of) 2.0.3 gives an exponential time algorithm to construct a β-solution to an SL-system $\mathscr{S}$ s.t. NP_OK_SUFF($\mathscr{S}$) = true.

Since, by [8, 4.1], β-solvability is undecidable NP_OK_NEC and NP_OK_SUFF cannot be equal. However for each SL-system $\mathscr{S}$ s.t. NP_regular_SL($\mathscr{S}$) = [**if** NP_OK_NEC($\mathscr{S}$) **then** NP_OK_SUFF($\mathscr{S}$) **else** true] is true the β-solvability problem is decidable. Again by [8, 4.1] the function NP_regular_SL cannot be identically true. In some sense the *goodness* of our result is measured by *how often* NP_regular_SL is true.

Theorem 2.0.4 says that NP_regular_SL takes value true often enough to isolate the very reason of undecidability for SL-systems, namely: a shortage of abstractions on the LHS occurrences of the RHS variables. Moreover, we are able to compute in Exponential Time an upper bound for such shortage. That is, given any SL-system $\mathscr{S}$ we can always find (in Exponential Time) a relaxation $\mathscr{S}'$ of $\mathscr{S}$ s.t. NP_regular_SL($\mathscr{S}'$) = true. In other words, though β-solvability for SL-systems is undecidable given any SL-system $\mathscr{S}$ we can always find a relaxation $\mathscr{S}'$ of $\mathscr{S}$ s.t. β-solvability for $\mathscr{S}'$ is decidable.

2.0. Theorem. *There are functions* NP_OK_NEC, NP_OK_SUFF *from SL-systems to Boole s.t.:*

0. *The problem of deciding if* NP_OK_NEC($\mathscr{S}$) = true *holds is in* NP.
1. *The problem of deciding if* NP_OK_SUFF($\mathscr{S}$) = true *holds is in* NP.
2. *For all SL-systems $\mathscr{S}$: if $\mathscr{S}$ is β-solvable then* NP_OK_NEC($\mathscr{S}$) = true.
3. *For all SL-systems $\mathscr{S}$: if* NP_OK_SUFF($\mathscr{S}$) = true *then $\mathscr{S}$ is β-solvable and we can construct a β-solution for $\mathscr{S}$ in time exponential in* Size($\mathscr{S}$).
4. *Let* NP_regular_SL *be the function from SL-systems to Boole defined as follows:* NP_regular_SL($\mathscr{S}$) = **if** NP_OK_NEC($\mathscr{S}$) **then** NP_OK_SUFF($\mathscr{S}$) **else** true. *Then:*

(a) NP_regular_SL *can be computed in exponential time.*

(b) *For any SL-system $\mathscr{S}$ there exists an SL-system $\mathscr{S}'$ s.t.: $\mathscr{S}'$ is a relaxation of $\mathscr{S}$,* NP_regular_SL($\mathscr{S}'$) = true *and $\mathscr{S}'$ can be computed from $\mathscr{S}$ in Exponential Time.*

Proof. See Appendix A. □

2.1. Definition. (1) From now on NP_regular_SL, NP_OK_SUFF, NP_OK_NEC are the functions defined in (the proof of) 2.0. However, the reader not interested in the technical details can read what follows without looking at the definitions (in the appendix) for such functions. In this case Examples 2.3 and 2.4 should be read as corollaries of Theorem 2.0.

(2) An SL-system $\mathscr{S}$ is said to be NP-regular iff NP_regular_SL($\mathscr{S}$) = true.

2.2. Remark. The β-solvability problem for NP-regular SL-system is in NP. In fact, by 2.0, we have: if NP_regular_SL($\mathscr{S}$) then [$\mathscr{S}$ is β-solvable iff NP_OK_NEC($\mathscr{S}$)].

2.3. Example. (1) An X-separability problem ([4] or [8, 3.1]) for λ-free sets is an NP-regular SL-system. Thus 2.0 and 2.2 yield a new proof of the decidability of the

X-separability problem. All together our proof is shorter then (a full version of) the proofs in [2, 3, 6].

(2) An NP-regular SL-system needs not be an X-separability problem. The SL-system $\mathscr{G} = (\{x(\lambda b.\, x(bb))(\lambda a_1 a_2 a_3.\, y) = y, x(\lambda b.\, x(\lambda a_1, \ldots, a_{38}.\, z))\ \ \Omega = z\},\ \{x\})$ is NP-regular. However, $\mathscr{G}$ is not regular (see [8, 5.3]), is not an X-separability problem and is not in the classes of systems defined in [3, 6 or 2]. Thus, the class of NP-regular SL-systems is strictly larger then the classes of systems defined in [2, 3] or [6].

(3) Let $\mathscr{G}$ be as above, $\mathscr{G}$ is NP-regular and NP_OK_NEC($\mathscr{G}$) = true. Thus, by 2.0, $\mathscr{G}$ is β-solvable. A β-solution in $G \equiv D\mathbf{I}$, where: $\mathbf{D}_1 \equiv \cdots \mathbf{D}_{16} \equiv \lambda_1 x_2.\, x_2 x_1 x_2$, $D \equiv \lambda t_1 t_2 t_3.\, t_2(\lambda a_1 \cdots a_{36}.\, t_3)\mathbf{D}_1 \cdots \mathbf{D}_{16} t_1 t_2 t_3$.

2.4. Example. Consider the system $\mathscr{S} = (|x(\lambda a\, b\, c.\, y)(\lambda ab.\, a) = y,\quad x(\lambda a\, b\, c.\, y)(\lambda a\, b.\, z) = z\},\ \{x\})$. The system $\mathscr{S}$ is regular [8, 5.3], but it is not NP-regular. This is because when the presence of self-application is restrained (as in [8, 5.3]), it is possible to design more clever compilers (i.e. system solving algorithms). Of course, we can consider the class of SL-systems s.t. [NP_regular_SL($\mathscr{S}$) or regular_SL($\mathscr{S}$)]. In this case, given an SL-system $\mathscr{S}$, we will use the algorithms in (the proof of) [8, 5.2] if regular_SL($\mathscr{S}$) = true and we will use the algorithms in (the proof of) 2.0 if NP_regular_SL($\mathscr{S}$) = true.

Theorem 2.0 shows that the β-solvability problem for NP-regular SL-systems is in NP. This result can be considerably sharpened showing that the X-separability problem (see [8, 3.1] or [4]) is an NP-complete problem. This, in particular, β-solvability for NP-regular SL-systems is an NP-complete problem.

2.5. Theorem. *The X-separability problem for λ-free sets* [8, 3.1] *is NP-complete.*

Proof. See Appendix B. □

2.6. Corollary. *The β-solvability problem for* NP-*regular SL-systems is* NP-*complete.*

Proof. From 2.5 and 2.0 simply noting that the X-separability problem for a λ-free set is the β-solvability problem for an NP-regular SL-system. □

2.7. Corollary. *Given an* NP-*regular SL-system $\mathscr{S}$ the problem of deciding if* NP_OK_NEC($\mathscr{S}$) = true *is* NP-*complete.*

Proof. From 2.2 and 2.6.

3. Applications

Regular SL-systems [8, 5.3] only allow a restrained presence of self-application. Thus the equational programming language based on regular SL-systems defined in

[8, 6.2] only allows a restrained presence of self-application. Using NP-regular SL-systems we can remove this restriction.

In this section we define an equational programming language which extends the one defined in [8, 6.2] by allowing unrestrained self-application. Any system satisfying the hypotheses of [8, 6.2] or those of 3.0 belongs to our language (see 2.4). A compiler for systems satisfying the hypotheses in [8, 6.2] has already been defined in [8, Appendix C]. Thus we only have to define a compiler for systems satisfying the hypotheses in 3.0. This is done in the proof of 3.0. Note that $\mathscr{S}^{\#}$ in 3.0 is an SL-system.

3.0. Theorem. *Let* $\mathscr{S} = (\Gamma_1 \cup \Gamma_2, \{x_1 \ldots x_n\})$ *be a system and* $\vec{y} \in (\mathrm{Var} - \{x_1 \ldots x_n\})$ *s.t.:*

H0. $\vec{y}$ *is assignable in* $\mathscr{S}$ *(see* [8, 6.0]).

H1. *Each equation in* Γ_1 *has form* $x\vec{M} = y(x_1\vec{y})\ldots(x_n\vec{y})\vec{z}$, *where:* $x \in \{x_1 \ldots x_n\}$, $y \in \{\vec{y}\}$, $\vec{z} \notin \{x_1 \ldots x_n, \vec{y}\}$ *and the variables in* $\{y \mid x\vec{M} = y(x_1\vec{y})\ldots(x_n\vec{y})\vec{z} \in \Gamma_1\}$ *are pairwise distinct.*

H2. *Each equation in* Γ_2 *has form* $x\vec{M} = z$, *where* $x \in \{x_1 \ldots x_n\}$ *and* $z \notin \{x_1 \ldots x_n, \vec{y}\}$.

H3. $\mathscr{S}^{\#} = (\{x\vec{M} = y \mid x\vec{M} = y(x_1\vec{y})\ldots(x_n\vec{y})\vec{z} \in \Gamma_1\} \cup \{u_{yz}\vec{M} = z \mid x\vec{M} = y(x_1\vec{y})\ldots(x_n\vec{y})\vec{z} \in \Gamma_1\}$ *and* $z \in \{\vec{z}\}$ *and* $u_{yz} \in \{\vec{u}\}\} \cup \Gamma_2$, $\{x_1 \ldots x_n, \vec{u}\})$ *is NP-regular, where* $\vec{u}$ *is a sequence of fresh variables s.t.:*

$$\{\vec{u}\} = \{u_{yz} \mid x\vec{M} = y(x_1\vec{y})\ldots(x_n\vec{y})\vec{z} \in \Gamma_1 \text{ and } z \in \{\vec{z}\}\}.$$

Then:

0. $\mathscr{S}$ *is* β*-solvable iff* NP_OK_NEC($\mathscr{S}^{\#}$). *Thus* β*-solvability is an* NP*-complete problem.*

1. *If* $\mathscr{S}$ *is* β*-solvable then a* β*-solution for* $\mathscr{S}$ *can be constructed in exponential time.*

2. *If* $\mathscr{S}$ *is* β*-solvable and* $\mathrm{Card}(\Gamma_1 \cup \Gamma_2) > 1$ *then* $\mathscr{S}$ *has a* β*-solution having normal form.*

Proof. See Appendix C. □

4. Conclusions

Though the β-solvability problem for SL-systems is undecidable [8, 4.1] we showed (2.0) that any SL-system has a *relaxation* [8, 5.1] for which the β-solvability problem is decidable. This shows that the only *source* of undecidability for SL-systems is a shortage of abstractions on the LHS occurrences of the RHS variables. This property yields a natural class of SL-systems (NP-regular SL-systems (2.1)) for which the β-solvability problem is decidable. However, the unrestrained presence of self-application makes the β-solvability problem for NP-regular SL-systems NP-complete (2.6).

NP-regular SL-systems and regular SL-systems (defined in [8, 2.3]) yield (3.0 and [8, 6.2]) an equational programming language in which:

- unrestrained presence of self-application is allowed,
- constrains on executable code to be generated by the compiler can be specified,
- (properties of) data structures can be described in an abstract way,

- λ-terms representing programs have normal form,
- inverse functions of constructors (of a data structure) run in constant time (e.g. as in [8, 0.6, 6.4]).

The price for allowing an unrestrained presence of self-application in our language is that deciding consistency of specifications is an NP-complete problem and finding executable codes (if any) satisfying a set of program specifications in the language takes, in general, exponential time.

Acknowledgements

I am grateful to Rick Statman for the many helpful discussions on the topics of this paper and for his valuable suggestions and stimulating questions.

I am also grateful to Corrado Böhm for his comments and suggestions about a preliminary and (even more) *indigestible* version of this paper.

Appendix A. Proof of 2.0

This proof is quite long so we divide it into many parts. In A.0 we give some useful basic definitions. In A.1 we define NP_OK_NEC and prove 2.0.0. In A.2 we define NP_OK_SUFF and prove 2.0.1. In A.3 we prove 2.0.2. In A.4 we prove 2.0.3. In A.5 we prove 2.0.4.

A.0. More on the λ-calculus

We introduce some notations that we will use later. Moreover, we assume the reader familiar with [8, A.0].

A.0.0. Notation. Let $k, q \in \mathbb{N}$. We define: $\mathbf{D}_q \equiv \lambda x_1 \ldots x_q x_{q+1} . x_{q+1} x_1 \ldots x_q x_{q+1}$, $\vec{\mathbf{D}}(k) = \mathbf{D}_1, \ldots \mathbf{D}_1$, where $|(\vec{\mathbf{D}}(k)| = k$.

A.0.1. Definition. Let $M \in \Lambda$, $\mathfrak{F} \subset_{\mathrm{f}} \Lambda$ and $X \subset_{\mathrm{f}} \mathrm{Var}$.

- A term G is said to be an X-version of M iff G is obtained from M as follows:

if $\mathrm{FV}(\mathrm{BT}(M)) \subseteq \mathrm{X}$
then $G := M$
else Begin
  Choose arbitrarily $\sigma \in BT(M)$ s.t.:
  $[\mathrm{head}(M_\sigma) \equiv b \in (\mathrm{FV}(\mathrm{BT}(M)) - X)]$ and $\forall \alpha < \sigma [\mathrm{head}(M_\sigma) \notin (\mathrm{FV}(M) - X)]$;
  Let g be a fresh variable;
  Let Q be the term obtained from M replacing the occurrence of b in σ with g;
  $G := Q[\mathrm{FV}(M) - X) := \Omega]$
end.

- A set $\mathfrak{G}$ is said to be an X-version of $\mathfrak{F}$ iff $\mathfrak{G}$ can be obtained from $\mathfrak{F}$ replacing each $M \in \mathfrak{F}$ with an (arbitrarily chosen) X-version of M.
- version $(X, \mathfrak{F}) = \{\mathfrak{G} \mid \mathfrak{G}$ is an X-version of $\mathfrak{F}\}$.

A.0.2. Example. (a) Let $M = \lambda a.\, xbbaa$.
Then $G_1 = \lambda a.\, x\Omega gaa$ and $G_2 = \lambda a.\, xg\Omega aa$ are $\{x\}$-versions of M.
(b) Let $\mathfrak{F} = \{\lambda a.\, xbbaa, \lambda a.\, a(x\, u)a\}$.
Then $\{\lambda a.\, xg\Omega aa, \lambda a.\, a(x\, v)a\} \in \text{version}(\{x\}, \mathfrak{F})$.

A.0.3. Notation. Let $\mathscr{S} = (\Gamma, X)$ be a system of $x \in X$.
- $\deg(\mathscr{S}) = \max\{\deg(M_\alpha) \mid M \in \text{left}(\mathscr{S})$ and $\alpha \in \text{BT}(M)\}$.
- $\deg(\mathscr{S}, x) = \max\{\deg(M_\alpha) \mid M \in \text{left}(\mathscr{S})$ and $\alpha \in \text{BT}(M)$ and $\text{head}(M_\alpha) \equiv x\}$.
- $\text{intdeg}(\mathscr{S}, x) = \max\{\deg(M_\alpha) \mid M \in \text{left}(\mathscr{S})$ and $\alpha \in \text{BT}(M)$ and $\alpha \neq \langle\ \rangle$ and $\text{head}(M_\alpha) \equiv x\}$.
- $\text{sat}(\mathscr{S}, x) = \max\{\text{intdeg}(\mathscr{S}, x) + 1, \deg(\mathscr{S}, x)\}$.
- $\text{degrgh}(\mathscr{S}) = \max\{\deg(M_\alpha) \mid M \in \text{left}(\mathscr{S})$ and $\alpha \in \text{BT}(M)$ and $\text{head}(M_\alpha) \notin \text{head}(\text{right}(\mathscr{S}))\}$.
- $\text{ord}(\mathscr{S}) = \max\{\text{ord}(M_\alpha) \mid M \in \text{left}(\mathscr{S})$ and $\alpha \in \text{BT}(M)\}$.
- $\text{ordrgh}(\mathscr{S}) = \max\{\text{ord}(M_\alpha) \mid M \in \text{left}(\mathscr{S})$ and $\alpha \in \text{BT}(M)$ and $\text{head}(M_\alpha) \notin \text{head}(\text{right}(\mathscr{S}))\}$.
- $\text{ord}(\mathscr{S}, x) = \max\{\text{ord}(M_\alpha) \mid M \in \text{left}(\mathscr{S})$ and $\alpha \in \text{BT}(M)$ and $\text{head}(M_\alpha) \equiv x\}$.

As in [8, 6.2] finding β-solutions to SL-systems is the core of our compiler. GSL-systems will be used (in A.4) to construct β-solutions to SL-systems.

A.0.4. Definition. (i) A system $\mathscr{S} = (\Gamma, X)$ is said to be a GSL-system (generalized SL-system) iff its equations have form $x\vec{M} = z\vec{M}$, where $x \in X$ and $z \notin X$.
(ii) A system $\mathscr{S} = (\Gamma, \{x\})$ is said to be a GHSL-system (generalized HSL-system) iff its equations have form $x\vec{M} = z\vec{M}$, wehre $z \notin \{x\}$ and $x \notin \text{FV}(\vec{M})$.

The following notions will be useful to work with SL-systems and GSL-systems.

A.0.5. Notation. (1) Let $\mathscr{S} = (\Gamma, X)$ be an SL-system. We define (see [8, A.0.0]):
(2) $\text{gsl}(\mathscr{S}) = (\{x\vec{M} = z\vec{M} \mid x\vec{M} = z \in \mathscr{S}\}, X)$.
(3) $\text{gsl}^*(\mathscr{S}) = (\{x\vec{M}' = z\vec{M}' \mid x\vec{M} = z \in \mathscr{S}$ and $x\vec{M}' = (x\vec{M})[\sigma := \lambda t.\, t\Omega_1 \ldots \Omega_{\deg(x\vec{M})} \mid \sigma \in \text{BT}(x\vec{M})$ and $\text{head}((x\vec{M})_\sigma) \equiv z]\}, X)$.
(4) Let $\mathscr{S} = (\Gamma, X)$ be a GSL-system. We define:
(5) $\text{sl}(\mathscr{S}) = (\{x\vec{M} = z \mid x\vec{M} = z\vec{M} \in \mathscr{S}\}, X)$.
(6) $\text{sl}^*(\mathscr{S}) = (\{(x\vec{M})[z := \lambda a_1 \ldots a_{\deg(x\vec{M})}.\, z] = z \mid x\vec{M} = z\vec{M} \in \mathscr{S}\}, X)$.

If an SL-system is β-solvable then the degree of the nodes having as head an RHS variable should be 0.

A.0.6. Definition. An SL-system $\mathscr{S} = (\Gamma, X)$ is said to be β-suitable iff $\forall M = z \in \mathscr{S}\ \forall \alpha \in \mathrm{BT}(M)[\mathrm{head}(M_\alpha) \equiv z \Rightarrow \deg(M_\alpha) = 0]$ (see [8, A.0.0]).

Note that if $\mathscr{S}$ is LR-distinct (see [8, A.1.5.3]) then $\mathscr{S}$ is β-suitable.

A.0.7. Remark. (a) Let $\mathscr{S} = (\Gamma, X)$ be an SL-system. If $D[\ ]$ is a β-solution for $\mathrm{gsl}^*(\mathscr{S})$ and $\mathrm{gsl}^*(\mathscr{S})$ is β-suitable then $D[\ ]$ is also a β-solution for $\mathscr{S}$.
(b) Let $\mathscr{S} = (\Gamma, X)$ be a GSL-system and let T be a λ-theory (e.g. β or $\beta\eta$). If $D[\ ]$ is a $\mathbb{T}$-solution for $\mathscr{S}$ then $D[\ ]$ is a $\mathbb{T}$-solution for $\mathrm{sl}^*(\mathscr{S})$.

A.1. Definition of NP_OK_NEC *and Proof of 2.0.0*

We define (A.1.0) the predicate NP_OK_NEC. NP_OK_NEC gives a necessary condition of β-solvability for SL-systems. The predicates canonical and LR-distinct were deifned in [8, A.1.1, A.1.5.3]. Predicates GPFR and approx are defined in A.1.3.3, A.1.4.

A.1.0. Definition. Let $\mathscr{S} = (\Gamma, X)$ be an SL-system. NP_OK_NEC$(\mathscr{S})$ = true iff $\exists \mathscr{S}^1 \in \mathrm{canonical}(\mathscr{S})\ \exists \mathscr{S}^2 \in \mathrm{approx}(\mathscr{S}^1)[\mathscr{S}^2$ is GPFR and LR-distinct].

NP_OK_NEC can be seen as a test of *consistency* for a set of SL-specifications $\mathscr{S}$. If $\mathscr{S}$ does not pass such test then there are no executable codes satisfying the specifications in $\mathscr{S}$. NP_OK_NEC takes into account the effect of self-application (whereas OK_NEC in [8, A.1.0] does not). The price for the present refinement is that to compute NP_OK_NEC is an NP-complete problem (2.7), whereas to compute OK_NEC is a Polynomial Time problem [8, 5.2.0].

In A.1.1–A.1.3 we define GPFR. If an SL-system is β-solvable then any subterm of an LHS term should be *distinguishable* (A.1.2.3) from any LHS term. The following example will clarify the matter.

A.1.1. Example. Let $\mathscr{S} = (\{xy(x\Omega\Omega) = y, x\Omega z = z\}, \{x\})$. The system $\mathscr{S}$ is canonical (see [8, A.1.1]), PFR (see [8, A.1.7]) and LR-distinct (see [8, A.1.5.3]), but it is not β-solvable. This is because the subterm $x\Omega\Omega$ of $xy(x\Omega\Omega)$ is *indistinguishable* from $x\Omega z$ in left$(\mathscr{S})$ (see [8, A.0.2]). This suggests the following strengthening of the notion of PFR.

An SL-system $\mathscr{S}$ is said to be GPFR ($\mathscr{S}$ satisfies the generalized prefix rule) iff any subterm (proper or improper) of an LHS term is *distinguishable* from any LHS term (a formal definition is in A.1.2.3).

Definitions A.1.2.2, A.1.2.3 are reformulations of, respectively [8, A.1.7] and [3, 2.5.1].

A.1.2. Definition. Let $\mathscr{S} = (\Gamma, X)$ be an SL-system.
0. $\mathrm{unif}(\mathscr{S}) = \{xM_1 \dots M_m\Omega_{m+1} \dots \Omega_{\mathrm{sat}(\mathscr{S},x)} \mid xM_1 \dots M_m = z \in \mathscr{S}\}$.

1. $\mathrm{crit}(\mathscr{S}) = \{xQ_1 \ldots Q_q u_{M_\alpha, q+1} \ldots u_{M_\alpha, \mathrm{sat}(\mathscr{S}, x)} \mid \exists M \in \mathrm{left}(\mathscr{S}) \exists \alpha \in \mathrm{BT}(M)$ $[\alpha \neq \langle\ \rangle$ and $M_\alpha = \lambda \vec{a}.\ xQ_1 \ldots Q_q$ and $x \in X$ and $\{u_{M_\alpha, i} \mid i \in \{1, \ldots, \mathrm{sat}(\mathscr{S}, x)\}\}$ is a set of pairwise distinct fresh variables$]\}$.

2. The system $\mathscr{S}$ is said to be PFR ($\mathscr{S}$ satisfies the prefix rule) iff $\mathrm{unif}(\mathscr{S})$ is distinct [8, A.1.6].

3. The system $\mathscr{S}$ is said to be GPFR ($\mathscr{S}$ satisfies the generalized prefix rule) iff $\exists \mathfrak{G} \in \mathrm{version}(X, \mathrm{crit}(\mathscr{S}))$ $[(\mathrm{unif}(\mathscr{S}) \cup \mathfrak{G})$ is distinct$]$.

A.1.3. Remark. Let $\mathscr{S}$ be an SL-system.

0. If $\mathscr{S}$ is GPFR then $\mathscr{S}$ is PFR.
1. If $\mathscr{S}$ is PFR then $\mathrm{left}(\mathscr{S})$ is distinct (see [8, A.1.6]).
2. The problem of deciding if $\mathscr{S}$ is GPFR is in NP.

In A.1.4 we define approx. Even a canonical SL-system (see [8, A.1.1]) might contain parts that are not useful for the computation. We can get rid of such parts considering only suitable approximations of the LHS terms. This leads us to the following definition.

A.1.4. Definition. Let $\mathscr{S} = (\Gamma, X)$ be an SL-system. $\mathrm{approx}(\mathscr{S}) = \{\Pi, X) \mid \Pi$ is obtained from Γ replacing each $M = z \in \Gamma$ with $M' = z$ s.t.: $M' \sqsubseteq M$ and $\exists! \alpha \in \mathrm{BT}(M')$ $[\mathrm{head}(M'_\alpha) \equiv z]\}$. Note that if $\mathscr{S}$ is canonical then any $\mathscr{S}' \in \mathrm{approx}(\mathscr{S})$ is canonical.

A.1.5. Remark. Let $\mathbb{T}$ be a sms theory (e.g. β or $\beta\eta$) and $\mathscr{S} = (\Gamma, X)$ be an SL-system.

- $\mathscr{S}$ is $\mathbb{T}$-solvable iff $\exists \mathscr{S}^+ \in \mathrm{approx}(\mathscr{S})$ $[\mathscr{S}^+$ is $\mathbb{T}$-solvable$]$.
- If $\mathscr{S}$ is β-solvable then $\exists \mathscr{S}^+ \in \mathrm{approx}(\mathscr{S})$ $[\mathscr{S}^+$ is β-suitable$]$.

A.1.6. Remark. Let $\mathscr{S} = (\Gamma, X)$ be an SL-system. By continuity $\mathscr{S}$ is β ($\beta\eta$)-solvable iff there exists $\mathscr{S}' \in \mathrm{approx}(\mathscr{S})$ s.t.: $[\mathscr{S}'$ is β ($\beta\eta$)-solvable and $\forall M \in \mathrm{left}(\mathscr{S}')$ $[\mathrm{BT}(M)$ is finite$]]$.

Thus, when studying β ($\beta\eta$)-solvability, it is not restrictive to consider only SL-systems with LHS terms having finite Böhm-trees. This is what we are doing (since we adopt [8, 2.3]). If $\mathbb{T}$ is a sms theory (different from β or $\beta\eta$) the argument above in general does not work. Thus in this case it is a restriction to assume finite Böhm-trees for LHS terms.

The definition of NP_OK_NEC (in A.1.0) is now complete. We prove 2.0.0.

A.1.7. Proof of 2.0.0. From A.1.0 and A.1.3.2. □

A.2. Definition of NP_OK_SUFF *and Proof of 2.0.1*

We define (A.2.0) the predicate NP_OK_SUFF. NP_OK_SUFF gives a sufficient condition of β-solvability for SL-systems. All functions used in A.2.0 will be defined in A.2.1–A.2.8.

A.2.0. Definition. Let $\mathscr{S} = (\Gamma, X)$ be an SL-system. NP_OK_SUFF($\mathscr{S}$) = true iff $\exists \mathscr{S}^1 \in$ canonical(gsl*(single(Relax_asg_like($\mathscr{S}$)))) $\exists \mathscr{S}^2 \in$ approx($\mathscr{S}^1$) $\exists \mathscr{S}^3 \in$ dev($\mathscr{S}^2$) $\exists \mathscr{S}^4 \in$ canonical($\mathscr{S}^3$) s.t. [PFRLR*($\mathscr{S}^4$) is (right($\mathscr{S}^4$), $\emptyset$, 0)-distinct].

In A.2.1 we extend [8, A.1.1], A.1.4, A.0.6, A.1.2 to GSL-systems.

A.2.1. Notation. Let $\mathscr{S} = (\Gamma, X)$ be a GSL-system.

0. canonical($\mathscr{S}$) = $\{$gsl($\mathscr{G}$) | $\mathscr{G} \in$ canonical(sl($\mathscr{S}$)).
1. approx($\mathscr{S}$) = $\{$gsl($\mathscr{G}$) | $\mathscr{G} \in$ approx(sl($\mathscr{S}$))$\}$.
2. $\mathscr{S}$ is said to be β-suitable (PFR, GPFR, LR-distinct) iff sl($\mathscr{S}$) is β-suitable (PFR, GPFR, LR-distinct).

A.2.2. Remark. Let $\mathscr{S} = (\Gamma, X)$ be an SL-system or a GSL-system. By the definition of LR-distinction (see [8, A.1.5.3]) we have: If $\exists \mathscr{S}^1 \in$ canonical($\mathscr{S}$) $\exists \mathscr{S}^2 \in$ approx($\mathscr{S}^1$) [$\mathscr{S}^2$ is GPFR and LR-distinct] then $\exists \mathscr{S}^1 \in$ canonical($\mathscr{S}$) $\exists \mathscr{S}^2 \in$ approx($\mathscr{S}^1$) [$\mathscr{S}^2$ is GPFR and β-suitable].

In A.2.3 we define single. Any system of equations $\mathscr{S} = (\Gamma, X)$ can be transformed into a system of equations $\mathscr{S}' = (\Gamma', \{x\})$ with only one unknown (e.g. as in [5, 5.6.0]).

A.2.3. Definition. (Böhm and Tronci [5, 5.6.0]). Let $\mathscr{S} = (\Gamma, \{x_1, \dots x_n\})$ be a system. We define:
single($\mathscr{S}$) = $(\{M[x_i := x\mathbf{U}_i^n \mid i \in \{1, \dots, n\}] = N[x_i := x\mathbf{U}_i^n \mid i \in \{1, \dots, n\}] \mid M = N \in \mathscr{S}\}$, $\{x\})$, where x is a fresh variable.

A.2.4. Remark (Böhm and Tronci [5, 5.6.0]). Let $\mathbb{T}$ be a λ-theory (e.g. β or $\beta\eta$) and $\mathscr{S} = (\Gamma, X)$ be a system. Then $\mathscr{S}$ is $\mathbb{T}$-solvable iff single($\mathscr{S}$) is $\mathbb{T}$-solvable. Thus it is not restrictive to consider only systems with only one unknown.

In A.2.5 we define PFPLR*. This definition is analogous to that in [8, A.2.2].

A.2.5. Definition. Let $\mathscr{S} = (\Gamma, X)$ be a GSL-system. We define: PFRLR*($\mathscr{S}$) = $\{\langle x, M_1, \dots, M_m, \Omega \rangle \mid xM_1 \dots M_m = zM_1 \dots M_m \in \mathscr{S}\} \cup [\langle x, M_1, \dots, M_m, z \rangle \mid xM_1 \dots M_{m+k} = zM_1 \dots M_{m+k} \in \mathscr{S}$ and $k > 0\}$.

In A.2.6 we define Relax_asg_like.

A.2.6. Definition. Let $\mathscr{S} = (\Gamma, \{\vec{x}\})$ be a system and $\vec{y} \equiv y_0, \dots, y_{k-1} \in (\mathrm{Var} - \{\vec{x}\})$.

- The sequence $\vec{y}$ is said to be asg-like (assignable-like) in $\mathscr{S}$ iff $\exists j \in \mathbb{N}\ \forall M = N \in \mathscr{S}$ $\forall \alpha \in \mathrm{BT}(M)$ [head$(M_\alpha) \in \{\vec{x}\} \Rightarrow \forall i \in \{0, \dots, k-1\}\ M_{\alpha * \langle j+i \rangle} = y_i$].
- Let $f: \{\vec{y}\} \to \mathbb{N}$.
  $R(\vec{y}, f)[\ ] \equiv (\lambda y_0 \dots y_{k-1}.[\ ])(\lambda v_1 \dots v_{f(y_0)}.y_0) \dots (\lambda v_1 \dots v_{f(y_{k-1})}.y_{k-1})$.
- Let $f: \{\vec{y}\} \to \mathbb{N}$. $R(\vec{y}, f)[\mathscr{S}] = (R(\vec{y}, f)[M] = N \mid M = N \in \mathscr{S}\}, \{\vec{x}\})$.

- Let $\mathscr{S} = (\Gamma, \{\vec{x}\})$ be a system and $\vec{y} \equiv y_0, \ldots, y_{k-1} \in (\mathrm{Var} - \{\vec{x}\})$ be the sequence of its asg-like variables. Let $f: \{\vec{y}\} \to \mathbb{N}$ be defined as follows: for all $y \in \{\vec{y}\}$ $f(y) = \max\{\deg(M) \mid M \in \mathrm{left}(\mathscr{S})\}$. We define: $\mathrm{Relax\_asg\_like}(\mathscr{S}) = R(\vec{y}, f)[\mathscr{S}]$. Note that if $\{\vec{y}\}$ is empty then $\mathrm{Relax\_asg\_like}(\mathscr{S}) = \mathscr{S}$. The effect of $\mathrm{Relax\_asg\_like}(\mathscr{S})$ is of adding abstractions to (relaxing) the asg-like variables of $\mathscr{S}$.

In A.2.7–A.2.8 we define dev. A node is minimal (A.2.7) if it has no unknown below it.

A.2.7. Definition. Let $X \subset_f \mathrm{Var}$ and $M \in \Lambda$. A node $\alpha \in \mathrm{Seq}$ is said to be X-minimal for M iff $\{\alpha \neq \langle\ \rangle$ and $\mathrm{head}(M_\alpha) \in X$ and $\neg\exists\beta \in \mathrm{BT}(M)$ $[\beta > \alpha$ and $\mathrm{head}(M_\beta) \in X]]$.

In order to find a β-solution to the SL-system $\mathscr{S}$ we look for a β-solution to the GSL-system $\mathrm{gsl}^*(\mathscr{S})$.

We construct a β-solution to a GSL-system $\mathscr{S}$ transforming $\mathscr{S}$ into a GHSL-system (thus eliminating self-application). This last transformation is defined in A.2.8.

A.2.8. Definition. Let $\mathscr{S} = (\Gamma, \{x\})$ be a GSL-system and $p \in \mathbb{N}$.

0. A system $\mathscr{G}$ is said to be a p-development of $\mathscr{S}$ iff $\mathscr{G}$ is obtained from $\mathscr{S}$ as follows:

Begin
  For all $x\vec{M} = z\vec{M} \in \mathscr{S}$ **do**
  Begin
    $M' := x\vec{M}$;
    While $\exists\alpha \in \mathrm{BT}(M')$ $[\alpha$ is $\{x\}$-minimal for $M']$ **do**
    Begin
      Let α an arbitrarily chosen $\{x\}$-minimal node for M' (e.g. the leftmost);
      Let $M'_\alpha = \lambda\vec{a}.\, xQ_1 \ldots Q_q$;
      Let $t_{q+1} \ldots t_{\mathrm{sat}(\mathscr{S},x)}$ be fresh variables;
      Choose arbitrarily $\sigma \in \mathrm{BT}(xQ_1 \ldots Q_q t_{q+1} \ldots t_{\mathrm{sat}(\mathscr{S},x)})$ s.t.
      $[(xQ_1 \ldots Q_q t_{q+1} \ldots t_{\mathrm{sat}(\mathscr{S},x)})_\sigma = \lambda v_1 \ldots v_r.\, bH_1 \ldots H_h$ and
      $b \in \mathrm{FV}(xQ_1 \ldots Q_q t_{q+1} \ldots t_{\mathrm{sat}(\mathscr{S},x)})$ and $b \notin \{x\}]$;
      $\mathrm{top1} := \mathrm{degrgh}(\mathscr{S}) + \mathrm{ordrgh}(\mathscr{S})] + 1 + p$; (see 2.4)
      $\mathrm{top2} := 2[\mathrm{degrgh}(\mathscr{S}) + \mathrm{ordrgh}(\mathscr{S})] + 1$;
      $k := \mathrm{Card}(\{\beta \in \mathrm{Seq} \mid \alpha < \beta < \alpha^* \sigma$ and $\deg(M'_\beta) > \mathrm{degrgh}(\mathscr{S})\}) + 1$;
      $\mathrm{top} := \mathrm{top1} + k\,\mathrm{top2}$;
      if $b \equiv z$
      then Begin
        $j :=$ **if** $r \geqslant \mathrm{top} + \mathrm{sat}(\mathscr{S}, x)$ **then** $\mathrm{top} + \mathrm{sat}(\mathscr{S}, x)$ **else** r;
        Let M' be the term obtained from M' replacing M'_α with
        $\lambda\vec{a}\, t_{q+1} \ldots t_{\mathrm{sat}(\mathscr{S},x)}\, b_{j+1} \ldots b_r.\, z$;

Let z' be a fresh variable;
Let $xQ_1^1 \dots Q_q^1 t_{q+1}^1 \dots t_{\mathrm{sat}(\mathscr{S},x)}^1$ be the term obtained from
$xQ_1 \dots Q_q t_{q+1} \dots t_{\mathrm{sat}(\mathscr{S},x)}$ replacing the occurrence of z in σ with z';
(see [8, A.0.0]) $(xQ_1^2 \dots Q_q^2 t_{q+1}^2 \dots t_{\mathrm{sat}(\mathscr{S},x)}^2) :=$
$(xQ_1^1 \dots Q_q^1 t_{q+1}^1 \dots t_{\mathrm{sat}(\mathscr{S},x)}^1)\,[\sigma := \lambda t.\, t\Omega_1 \dots \Omega_{\mathrm{sat}(\mathscr{S},x)+r-j}]$;
$\Delta_1 := \Delta_1 \cup \{xQ_1^2 \dots Q_q^2 t_{q+1}^2 \dots t_{\mathrm{sat}(\mathscr{S},x)}^2 = z'Q_1^2 \dots Q_q^2 t_{q+1}^2 \dots t_{\mathrm{sat}(\mathscr{S},x)}^2\}$;
end;
else Begin
Let $xQ_1^3 \dots Q_q^3 t_{q+1}^3 \dots t_{\mathrm{sat}(\mathscr{S},x)}^3$ be the term obtained from
$xQ_1 \dots Q_q t_{q+1} \dots t_{\mathrm{sat}(\mathscr{S},x)}$ replacing
$(xQ_1 \dots Q_q t_{q+1} \dots t_{\mathrm{sat}(\mathscr{S},x)})_\sigma$ with
$\lambda v_1 \dots v_r v_{r+1} \dots v_{\mathrm{top}}.\, bH_1 \dots H_h v_{r+1} \dots v_{\mathrm{top}}$;
Let M' be the term obtained from M' replacing M'_α with (see 2.0)
$\lambda \vec{a} t_{q+1} \dots t_{\mathrm{sat}(\mathscr{S},x)}.\, b\Omega_1 \dots \Omega_h \Omega_{r+1} \dots \Omega_{\mathrm{top1}}\, \vec{\mathbf{D}}(k^*\,\mathrm{top2})$
$Q_1^3 \dots Q_q^3 t_{q+1}^3 \dots t_{\mathrm{sat}(\mathscr{S},x)}^3$;
Let g be a fresh variable;
Let $xQ_1^4 \dots Q_q^4 t_{q+1}^4 \dots t_{\mathrm{sat}(\mathscr{S},x)}^4$ be the term obtained from
$xQ_1^3 \dots Q_q^3 t_{q+1}^3 \dots t_{\mathrm{sat}(\mathscr{S},x)}^3$ replacing the occurrence of b in σ
with g;
$xQ_1^5 \dots Q_q^5 t_{q+1}^5 \dots t_{\mathrm{sat}(\mathscr{S},x)}^5 :=$
$(xQ_1^4 \dots Q_q^4 t_{q+1}^4 \dots t_{\mathrm{sat}(\mathscr{S},x)}^4)[g := \lambda c_1 \dots c_h v_{r+1} \dots v_{\mathrm{top}}.\, g]$;
$\Delta_1 := \Delta_1 \cup \{xQ_1^5 \dots Q_q^5 t_{q+1}^5 \dots t_{\mathrm{sat}(\mathscr{S},x)}^5 = gQ_1^5 \dots Q_q^5 t_{q+1}^5 \dots t_{\mathrm{sat}(\mathscr{S},x)}^5\}$
end
end;
Let $M' = x\vec{M}'$;
$\Gamma_1 := \Gamma_1 \cup \{x\vec{M}' = z\vec{M}'\}$
end;
$\mathscr{G} := (\Gamma_1 \cup \Delta_1, \{\vec{x}\})$
end.

1. $\mathrm{dev}(p, \mathscr{S}) = \{\mathscr{G} \mid \mathscr{G}$ is a p-development of $\mathscr{S}\}$.
2. $\mathrm{dev}(\mathscr{S}) = \cup\{\mathrm{dev}(p, \mathscr{S}) \mid p \in \{0, \dots, \mathrm{ord}(\mathscr{S}) - \mathrm{ordrgh}(\mathscr{S})\}\}$.

The definition of NP_OK_SUFF in A.2.0 is now complete. We prove 2.0.1.

A.2.9. Proof of 2.0.1. From A.2.0, A.2.1, A.2.3, A.2.5, A.2.6, A.2.8. □

A.3. Proof of 2.0.2

We show (A.3.0) that NP_OK_NEC gives a necessary condition of β-solvability for SL-systems. This proves 2.0.2. Theorem 2.0.2 strengthens [8, 5.2.2] and [3, A.3.4].

A.3.0. Proof of 2.0.2. From [8, A.1.2, A.3.4], A.3.4 and A.1.0. □

Before proving A.3.4 we look at a few examples.

A.3.1. Example. (i) The system $\mathscr{S} = (\{xy(x\Omega\Omega) = y, x\Omega z = z\}, \{x\})$ is not GPFR (but it is PFR). Moreover, there is no $\mathscr{S}^1 \in \text{approx}(\mathscr{S})$ s.t. $\mathscr{S}^1$ is GPFR. Hence, by 2.0.2, $\mathscr{S}$ is not β-solvable.

(ii) The system $\mathbb{Q} = (\{x(\lambda ab.\, z) = z, x(\lambda a.\, u) = u\}, \{x\})$ is GPFR, but it is not LR-distinct (see [8, A.1.5.3]). Hence, by 2.0.2, $\mathbb{Q}$ is not β-solvable.

Of course, by [8, 4.1], the conditions in 2.0.2 are not sufficient.

A.3.2. Counterexample. The system $\mathscr{S} = (\{x(\lambda a.\, az) = z, x(\lambda a.\, a(au)) = u, x(\lambda a.\, a(a(a\Omega y))) = y\}, \{x\})$ is GPFR and LR-distinct, but it is not β-solvable.

From Theorem 2.0.2 we also get a necessary condition of β-solvability for GSL-systems.

A.3.3. Proposition. *Let $\mathscr{S} = (\Gamma, X)$ be a GSL-system. If $\mathscr{S}$ is β-solvable then* NP_OK_NEC(sl*($\mathscr{S}$)) = true.

Proof. From 2.0.2 and A.0.7. □

Lemma A.3.4 generalizes [8, A.3.5] and completes the proof of 2.0.2.

A.3.4. Lemma. *Let $\mathscr{S} = (\Gamma, X)$ be a canonical SL-system and $\mathbb{T}$ be a sms theory (e.g. β or $\beta\eta$). If $\mathscr{S}$ is $\mathbb{T}$-solvable then $\exists \mathscr{G} \in \text{approx}(\mathscr{S})$ [$\mathscr{G}$ is* GPFR].

Proof. Let $D[\ ]$ be a $\mathbb{T}$-solution for $\mathscr{S}$. Let $\mathscr{G} \in \text{approx}(\mathscr{S})$ s.t. $D[\ ]$ is a $\mathbb{T}$-solution for $\mathscr{G}$ and $\forall M = z \in \mathscr{G}\ \forall M' \in \Lambda[[M' \sqsubseteq M$ and $\text{BT}(M') \neq \text{BT}(M)] \Rightarrow D[M'] \notin \text{SOL}]$. Note that $\forall M \in (\text{unif}(\mathscr{G}) \cup \text{crit}(\mathscr{G}))\, D[M] \in \text{SOL}$. Note that $\forall Q \in \text{crit}(\mathscr{G})\, D[Q]$ can only have one of the following forms:
form 0: $D[Q] = \lambda \vec{a}.\, u\vec{L}$ and $u \in \{\vec{a}\}$;
form 1: $D[Q] = \lambda \vec{a}.\, u\vec{L}$ and $u \in (\text{FV}(Q) - (X \cup \text{right}(\mathscr{G})))$;
form 2: $D[Q] = \lambda \vec{a}.\, z\vec{L}$ and $z \in \text{right}(\mathscr{G})$.
Thus there is an X-version $\mathscr{F}$ of crit($\mathscr{G}$) s.t.:
P0. $\forall Q \in (\text{unif}(\mathscr{G}) \cup \mathscr{F})[D[Q] \in \text{SOL}]$;
P1. If $Q = xQ_1 \ldots Q_q \in \text{crit}(\mathscr{G})$ and $D[Q]$ has form 0 then the X-version Q^+ of Q in $\mathscr{F}$ has form $Q^+ = xQ_1^+ \ldots Q_q^+$ and $Q_q^+ \in \text{SOL}$.

Suppose that $(\text{unif}(\mathscr{G}) \cup \mathscr{F})$ is not distinct. Then there are $M, N \in (\text{unif}(\mathscr{G}) \cup \mathscr{F})$ s.t. $\text{ind}(\text{unif}(\mathscr{G}) \cup \mathscr{F}, M, N)$ (see [8, A.0.0]).
By [5, 3.4.0] this implies $\text{ind}(D[\text{unif}(\mathscr{G}) \cup \mathscr{F}], D[M], D[N])$.

Case 0: $D[M]$ and $D[N]$ have both form 0.
Then, by [5, 3.4.0], $\forall Q \in \text{ind}(\text{unif}(\mathscr{G}) \cup \mathscr{F}, M) = \{L \mid \text{ind}(\text{unif}(\mathscr{G}) \cup \mathscr{F}, M, L)\}$ [$D[Q]$ has form 0]. Hence, by P1 and the construction of $\mathscr{F}$, $\neg\text{ind}(\text{unif}(\mathscr{G}) \cup \mathscr{F}, M, N)$. Contradiction.

Case 1: At least one of $D[M], D[N]$ does not have form 0.
Then $\neg\text{ind}(D\,\text{unif}(\mathscr{G}) \cup \mathscr{F}], D[M], D[N])$. By [5, 3.4.0] this implies $\neg\text{ind}(\text{unif}(\mathscr{G}) \cup \mathscr{F}, M, N)$. Contradiction.

Thus $(\text{unif}(\mathscr{G}) \cup \mathscr{F})$ is distinct. □

A.4. Proof of 2.0.3

We show (A.4.0) that NP_OK_SUFF is a sufficient condition of β-solvability for SL-systems and we give an Exponential Time algorithm to find a β-solution to an SL-system satisfying NP_OK_SUFF. This algorithm is the core of our compiler (3.0).

A.4.0. Proof of 2.0.3. From A.2.0, A.4.6 and A.4.4 considering that the algorithms in A.4.6, A.4.4, A.4.2, A.4.1, A.2.8 run in exponential time. □

In A.4.1–A.4.3 we solve GHSL-systems. In A.4.4–A.4.6 we solve SL-systems.

The following proposition (analogous to [8, A.4.2.5]) gives a sufficient condition of β-solvability for GHSL-systems (see [8, A.2.1.4] for (Z, X, e)-distinction).

A.4.1. Proposition. *Let* $\mathscr{S} = (\Gamma, \{x\})$ *be a GHSL-system. If* $\exists \mathscr{G} \in \text{canonical}(\mathscr{S})$ [PFRLR*$(\mathscr{G})$ is $(\text{right}(\mathscr{G}), \emptyset, 0)$-*distinct*] *then* $\mathscr{S}$ *is* β-*solvable.*

Proof. Let $\mathscr{G} \in \text{canonical}(\mathscr{S})$ s.t. PFRLR*$(\mathscr{G})$ is $(\text{right}(\mathscr{G}), \emptyset, 0)$-distinct. By induction on $\text{Card}(\{\deg(M) \mid M \in \text{left}(\mathscr{G})\}) = d(\mathscr{G})$ we construct a β-solution for $\mathscr{G}$ and hence for $\mathscr{S}$.

Case 0: $d(\mathscr{G}) = 1$. Let $m = \min\{\deg(M) \mid M \in \text{left}(\mathscr{G})\}$,
$\Pi = \{x\langle M_1, \ldots, M_m\rangle = z \mid xM_1 \ldots M_m = zM_1 \ldots M_m \in \mathscr{G}\}$ and $\mathscr{G}' = (\Pi, \{x\})$.
By [8, A.4.2.4] $\mathscr{G}'$ is β-solvable. Let F be a β-solution for $\mathscr{G}'$.
Then $D \equiv \lambda t_1 \ldots t_m.\, F\langle t_1 \ldots t_m\rangle t_1 \ldots t_m$ is a β-solution for $\mathscr{G}$.

Case 1: $d(\mathscr{G}) > 1$. Let $m = \min\{\deg(M) \mid M \in \text{left}(\mathscr{G})\}$,

$$\mathscr{G}_{1,m} = (\Gamma_{1,m}, \{x\}) = (\{x\Omega M_1 \ldots M_m = zM_1 \ldots M_m \mid xM_1 \ldots M_m = zM_1 \ldots M_m \in \mathscr{G}\}, \{x\}),$$

$$\mathscr{G}_{2,m} = (\Gamma_{2,m}, \{x\}) = (\{xzM_1 \ldots M_m = zM_1 \ldots M_m \mid \exists k > 0\ [xM_1 \ldots M_{m+k} = zM_1 \ldots M_{m+k} \in \mathscr{G}]\}, \{x\}),$$

$$\mathscr{G}_{3,m} = (\Gamma_{3,m}, \{x\}) = (\Gamma_{1,m} \cup \Gamma_{2,m}, \{x\}).$$

Let $\mathscr{G}^{\#}_{3,m} = (\Gamma^{\#}, \{x\})$, where

$$\Gamma^{\#} = \{x\langle M_0, \ldots, M_m\rangle = z \mid xM_0 M_1 \ldots M_m = zM_1 \ldots M_m \in \mathscr{G}_{3,m}\}.$$

Since PFRLR*$(\mathscr{G})$ is $(\text{right}(\mathscr{G}), \emptyset, 0)$-distinct there exists $\mathscr{G}'' \in \text{canonical}(\mathscr{G}^{\#}_{3,m})$ s.t. $\text{left}(\mathscr{G}'')$ is $(\text{right}(\mathscr{G}''), \emptyset, 0)$-distinct. Hence, by [8, A.4.2.4], $\mathscr{G}''$ is β-solvable. Thus $\mathscr{G}^{\#}_{3,m}$ is β-solvable. Let G be a β-solution for $\mathscr{G}^{\#}_{3,m}$.

Then $F \equiv \lambda t_0 t_1 \ldots t_m.\ G\langle t_0, t_1, \ldots, t_m\rangle\ t_1 \ldots t_m$ is a β-solution for $\mathscr{G}_{3,m}$. Let

$$\mathscr{G}_{4,m} = (\Gamma_{4,m}, \{x\}) = (\{xM_1 \ldots M_n = zM_1 \ldots M_n \mid xM_1 \ldots M_n = zM_1 \ldots M_n \in \mathscr{G} \text{ and } n > m\}, \{x\}).$$

Note that PFRLR*($\mathscr{G}_{4,m}$) is (right($\mathscr{G}_{4,m}$), $\emptyset$, 0)-distinct and $d(\mathscr{G}_{4,m}) < d(\mathscr{G})$. Thus by induction hypothesis $\mathscr{G}_{4,m}$ is β-solvable. Let H be a β-solution for $\mathscr{G}_{4,m}$. Then $D \equiv \lambda t_1 \ldots t_m.\ \mathrm{F}H t_1 \ldots t_m$ is a β-solution for $\mathscr{G}$. □

We give (A.4.2) a sufficient condition of β-solvability for GSL-systems.

A.4.2. Theorem. *Let $\mathscr{S} = (\Gamma, \{x\})$ be a GSL-system.*
If $\exists \mathscr{S}^1 \in \mathrm{canonical}(\mathscr{S})\ \exists \mathscr{S}^2 \in \mathrm{approx}(\mathscr{S}^1)\ \exists \mathscr{S}^3 \in \mathrm{dev}(\mathscr{S}^2)\ \exists \mathscr{S}^4 \in \mathrm{canonical}(\mathscr{S}^3)$

[PFRLR*($\mathscr{S}^4$) *is* (right($\mathscr{S}^4$), $\emptyset$, 0)-distinct]

then $\mathscr{S}$ is β-solvable.

Proof. Let $\mathscr{S}^1 \in \mathrm{canonical}(\mathscr{S})$, $\mathscr{S}^2 \in \mathrm{approx}(\mathscr{S}^1)$, $\mathscr{S}^3 \in \mathrm{dev}(\mathscr{S}^2)$ and $\mathscr{S}^4 \in$ canonical($\mathscr{S}^3$) s.t. PFRLR*($\mathscr{S}^4$) is (right($\mathscr{S}^4$), $\emptyset$, 0)-distinct. By A.4.1 $\mathscr{S}^3$ is β-solvable.

Let $D[\ ]$ be the β-solution for $\mathscr{S}^3$ obtained using the algorithm in A.4.1 and choosing the arbitrary values for the A_i's in the proof of [8, A.4.2.0] to be $\mathbf{D}_1$ (see A.0.0.). Then, by the construction of $\mathscr{S}^3$ from $\mathscr{S}^2$, $D[\ ]$ is also a β-solution for $\mathscr{S}^2$. Thus $D[\ ]$ is a β-solution of $\mathscr{S}$. □

A.4.3. Example. Let $\mathscr{S} = (\{x\mathbf{I}(\lambda b.\ x\mathbf{I}(bb))y = y\mathbf{I}(\lambda b.\ x\mathbf{I}(bb))y,\ x\mathbf{I}(\lambda b.\ x\mathbf{I}(\lambda a_1 \ldots a_{35}.\ z))\Omega = z\mathbf{I}(\lambda b.\ x\mathbf{I}(\lambda a_1 \ldots a_{35}.\ z))\Omega\}, \{x\})$. Let $\mathscr{S}^1 = \mathscr{S}^2 = \mathscr{S}$;

$$\begin{aligned}
\mathscr{S}^3 = (\{&x\mathbf{I}(\lambda bt.\ b\Omega_1\Omega_1 \ldots \Omega_8 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{17}.\ bbv_1 \ldots v_{17})t)y \\
&= y\mathbf{I}(\lambda bt.\ b\Omega_1\Omega_1 \ldots \Omega_8 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{17}.\ bbv_1 \ldots v_{17})t)y, \\
&x\mathbf{I}(\lambda v_1 \ldots v_{17}.\ g)t' = g\mathbf{I}(\lambda v_1 \ldots v_{17}.\ g)t', \\
&x\mathbf{I}(\lambda bta_{21} \ldots a_{35}.\ z)\Omega = z\mathbf{I}(\lambda bta_{21} \ldots a_{35}.\ z)\Omega, \\
&x\mathbf{I}(\lambda a_1 \ldots a_{17}.\ z')t = z'\mathbf{I}(\lambda a_1 \ldots a_{17}.\ z')t\}, \{x\}) \in \mathrm{dev}(3, \mathscr{S}), \\
\mathscr{S}^4 = (\{&x\mathbf{I}(\lambda bt.\ b\Omega_1\Omega_1 \ldots \Omega_8 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{17}.\ bbv_1 \ldots v_{17})t)y \\
&= y\mathbf{I}(\lambda bt.\ b\Omega_1\Omega_1 \ldots \Omega_8 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{17}.\ bbv_1 \ldots v_{17})t)y, \\
&x\mathbf{I}(\lambda v_1 \ldots v_{17}.\ g)\Omega = g\mathbf{I}(\lambda v_1 \ldots v_{17}.\ g)\Omega, \\
&x\mathbf{I}(\lambda bta_{21} \ldots a_{35}.\ z)\Omega = z\mathbf{I}(\lambda bta_{21} \ldots a_{35}.\ z)\Omega, \\
&x\mathbf{I}(\lambda a_1 \ldots a_{17}.\ z')\Omega = z'\mathbf{I}(\lambda a_1 \ldots a_{17}.\ z')\Omega\}, \{x\}) \in \mathrm{canonical}(\mathscr{S}^3).
\end{aligned}$$

Then PFRLR*($\mathscr{S}^4$) is (right($\mathscr{S}^4$), $\emptyset$, 0)-distinct. Hence, by A.4.2, $\mathscr{S}$ is β-solvable. A β-solution for $\mathscr{S}^4$ and for $\mathscr{S}$ is: $D \equiv \lambda t_1 t_2 t_3.\ t_2(\lambda a_1 \ldots a_{36}.\ t_3)\vec{\mathbf{D}}(16)t_1 t_2 t_3$.

Finally, we can give a sufficient condition of β-solvability for SL-systems (A.4.4).

A.4.4. Theorem. *Let $\mathscr{S} = (\Gamma, X)$ be an SL-system. If* $\exists\mathscr{S}^1 \in \text{canonical}(\text{gsl}^*(\text{single}(\mathscr{S})))$ $\exists\mathscr{S}^2 \in \text{approx}(\mathscr{S}^1)$ $\exists\mathscr{S}^3 \in \text{dev}(\mathscr{S}^2)$ $\exists\mathscr{S}^4 \in \text{canonical}(\mathscr{S}^3)$ $\text{PFRLR}^*(\mathscr{S}^4)$ *is* $(\text{right}(\mathscr{S}^4), \emptyset, 0)$-distinct] *then $\mathscr{S}$ is β-solvable.*

Proof. From A.4.2, A.0.7 and A.2.3. □

A.4.5. Example. Let $\mathscr{G} = (\{x(\lambda b.\, x(bb))(\lambda a_1 a_2 a_3.\, y) = y, \quad x(\lambda b.\, x(\lambda a_1 \dots a_{38}.\, z))\, \Omega = z\}, \{x\})$.
Then we have

$$\text{single}(\mathscr{G}) = (\{x\mathbf{I}(\lambda b.\, x\mathbf{I}(bb))(\lambda a_1 a_2 a_3.\, y) = y, x\mathbf{I}(\lambda b.\, x\mathbf{I}(\lambda a_1 \dots a_{38}.\, z))\Omega = z\}, \{x\}).$$

$\text{gsl}^*(\text{single}(\mathscr{G})) = \mathscr{S}$ (as in A.4.3).

Hence, by Example A.4.3 and Theorem A.4.4, $\mathscr{G}$ is β-solvable. A possible β-solution for $\text{gls}^*(\text{single}(\mathscr{G}))$ is D as in A.4.3. Thus, by A.0.7, D is also a β-solution for $\text{single}(\mathscr{G})$. Then, by A.2.3, $G \equiv D\mathbf{I}$ is a β-solution for $\mathscr{G}$.

There are cases in which *relaxations* [8, 5.1] can be done "from inside" the λ-calculus. In particular this happens with asg-like variables (A.2.6).

A.4.6. Proposition. *Let $\mathbb{T}$ be a λ-theory (e.g. β or $\beta\eta$) and $\mathscr{S} = (\Gamma_1 \cup \Gamma_2, \{x_1, \dots, x_n\})$ be an SL-system* s.t.:

H0. $\vec{y} \equiv y_0, \dots, y_{k-1} \in (\text{Var} - \{x_1, \dots, x_n\})$ *is ags-like in $\mathscr{S}$*
H1. *Each equation in Γ_1 has form $x\vec{M} = y$, where $y \in \{\vec{y}\}$.*
H2. *Each equation in Γ_2 has form $x\vec{M} = z$, where $z \notin \{\vec{y}\}$.*

Then:

0. *Let $f: \{\vec{y}\} \to \mathbb{N}$. Then $\mathscr{S}$ is $\mathbb{T}$-solvable iff $R(\vec{y}, f)[\mathscr{S}]$ is $\mathbb{T}$-solvable.*
1. *$\mathscr{S}$ is $\mathbb{T}$-solvable iff* Relax_asg_like$(\mathscr{S})$ *is $\mathbb{T}$-solvable.*

Proof. 0. ($\Rightarrow$) Let $D[\ ] \equiv (\lambda x_1 \dots x_n.\, [\ ])\, D_1 \dots D_n$ be a $\mathbb{T}$-solution for $\mathscr{S}$.
Let j be as in A.2.6. For all $i \in \{1, \dots, n\}$ define:
$G_i \equiv \lambda t_1 \dots t_j y_0 \dots y_{k-1}.\, D_i t_1 \dots t_j (y_0 \Omega_1 \dots \Omega_{f(y_0)}) \dots (y_{k-1} \Omega_1 \dots \Omega_{f(y_{k-1})})$.
Then $G[\ ] \equiv (\lambda x_1 \dots x_n.\, [\ ])\, G_1 \dots G_n$ is a $\mathbb{T}$-solution for $R(\vec{y}, f)[\mathscr{S}]$.
($\Leftarrow$) Let $D[\ ] \equiv (\lambda x_1 \dots x_n.\, [\ ]) D_1 \dots D_n$ be a $\mathbb{T}$-solution for $R(\vec{y}, f)[\mathscr{S}]$.
Let j be as in A.2.6. For all $i \in \{1, \dots, n\}$ define:

$$G_i \equiv \lambda t_1 \dots t_j y_0 \dots y_{k-1}.\, D_i t_1 \dots t_j (\lambda v_1 \dots v_{f(y_0)}.\, y_0) \dots (\lambda v_1 \dots v_{f(y_{k-1})}.\, y_{k-1}).$$

Then $G[\ ] \equiv (\lambda x_1 \dots x_n.\, [\ ]) G_1 \dots G_n$ is a $\mathbb{T}$-solution for $\mathscr{S}$.
1. From 0 and A.2.6. □

A.5. Proof of 2.4.0

We show (A.5.0) that any SL-system has an NP-regular relaxation. This shows that NP-regular SL-systems isolate the *source* of undecidability for SL-systems, namely: a shortage of abstractions on the LHS occurrences of the RHS variables.

As an immediate application of this result we give (A.5.8) a new proof of the decidability of the X-separability problem for λ-free sets (see [8, 3.1]).

A.5.0. Proof of 2.4.0. Let $\mathscr{S}$ be an SL-system s.t. NP_OK_NEC$(\mathscr{S})$ = true. Then, by A.1.0 and A.0.6, $\exists \mathscr{S}^1 \in \text{canonical}(\mathscr{S})$ $\exists \mathscr{S}^2 \in \text{approx}(\mathscr{S}^1)$ [$\mathscr{S}^2$ is GPFR and β-suitable]. Hence, by A.5.7.0, there exists an SL-system $\mathscr{S}'$ s.t.: relax$(\mathscr{S}, \mathscr{S}')$ and NP_OK_SUFF$(\mathscr{S}')$ = true. Thus NP_regular_SL$(\mathscr{S}')$ = true. Moreover, by (the proof of) A.5.5 $\mathscr{S}'$ can be found in Exponential Time. $\square$

In A.5.1–A.5.7 we prove A.5.5, A.5.7.

A.5.1. Definition. We extend relax (see [8, 5.1]) to GSL-systems in the obvious way. Let $\mathscr{G}_1 = (\Pi_1, X)$, $\mathscr{G}_2 = (\Pi_2, X)$ be GSL-systems. We define: relax_SL$(\mathscr{G}_1, \mathscr{G}_2)$ = relax_SL(sl$(\mathscr{G}_1)$, sl$(\mathscr{G}_2)$). Let $\mathscr{S}$ be an SL-system or a GSL-system. We write $\mathscr{P} \in$ relax$(\mathscr{S})$ for relax_SL$(\mathscr{S}, \mathscr{P})$.

A.5.2. Remark. Let $\mathscr{S} = (\Gamma, X)$ be an SL-system or a GSL-system. $\exists \mathscr{S}^1 \in \text{canonical}(\mathscr{S})$ $\exists \mathscr{S}^2 \in \text{approx}(\mathscr{S}^1)$ [$\mathscr{S}^2$ is GPFR and β-suitable] iff $\forall \mathscr{P} \in \text{relax}(\mathscr{S})$ $\exists \mathscr{P}^1 \in \text{canonical}(\mathscr{P})$ $\exists \mathscr{P}^2 \in \text{approx}(\mathscr{P}^1)$ [$\mathscr{P}^2$ is GPFR and β-suitable] iff $\exists \mathscr{P} \in \text{relax}(\mathscr{S})$ $\exists \mathscr{P}^1 \in \text{canonical}(\mathscr{P})$ $\exists \mathscr{P}^2 \in \text{approx}(\mathscr{P}^1)$ [$\mathscr{P}^2$ is GPFR and β-suitable].

W.l.o.g. we can restrict our attention to normal systems (A.5.3). This will simplify our proofs.

A.5.3. Definition. A system $\mathscr{S} = (\Gamma, X)$ is said to be normal iff [Card$(X) = 1$ and $\exists n \in \mathbb{N}$ $\forall M \in (\text{left}(\mathscr{S}) \cup \text{right}(\mathscr{S}))$ $\forall \alpha \in \text{BT}(M)$ [head$(M_\alpha) \in X \Rightarrow \exists$ $i \in \{1, \ldots, n+1\}$ $M_{\alpha \bullet \langle 0 \rangle} = \mathbf{U}_i^{n+1}$]].

Given a system $\mathscr{S} = (\Gamma, X)$ then single$(\mathscr{S})$ is a normal system. Thus, by A.2.4, it is not restrictive to consider only normal systems.

Lemma A.5.4 shows that the transformation in A.2.8.0 does not loose information.

A.5.4. Lemma. *Let* $\mathscr{S} = (\Gamma, \{x\})$ *be a normal GSL-system. If* $\exists \mathscr{S}^1 \in \text{canonical}(\mathscr{S})$ $\exists \mathscr{S}^2 \in \text{approx}(\mathscr{S}^1)$ [$\mathscr{S}^2$ *is* GPFR *and* β*-suitable*] *then* $\exists \mathscr{S}^1 \in \text{canonical}(\mathscr{S})$ $\exists \mathscr{S}^2 \in \text{approx}(\mathscr{S}^1)$ $\forall p \in \mathbb{N}$ $\exists \mathscr{S}^3 \in \text{dev}(p, \mathscr{S}^2)$ $\exists \mathscr{S}^4 \in \text{canonical}(\mathscr{S}^3)$ [$\mathscr{S}^4$ is PFR].

Proof. We divide the proof into 3 parts.

Part 0. Let $\mathscr{S}^1 \in \text{canonical}(\mathscr{S})$ and $\mathscr{S}^2 \in \text{approx}(\mathscr{S}^1)$ s.t. $\mathscr{S}^2$ is GPFR and β-suitable. Let $\mathscr{G} \in \text{version}(\{x\}, \text{crit}(\mathscr{S}^2))$ s.t. $(\text{unif}(\mathscr{S}^2) \cup \mathscr{G})$ is distinct (such a $\mathscr{G}$ exists since $\mathscr{S}^2$ is GPFR). Let $p \in \mathbb{N}$ arbitrarily chosen. Let $\mathscr{S}^3 \in \text{dev}(p, \mathscr{S}^2)$ constructed *according* to $\mathscr{G}$. An example will be sufficient to define this construction.

0. Example. Let $\mathscr{S}^2 = (\{x\mathbf{I}(\lambda a.\, x\mathbf{I}az) = z\mathbf{I}(\lambda a.\, x\mathbf{I}az)\}, \{x\})$. Then $\text{crit}(\mathscr{S}^2) = \{x\mathbf{I}azt\}$. The following are $\{x\}$-versions of $\text{crit}(\mathscr{S}^2)$:

$$\mathscr{G}_1 = \{x\mathbf{I}a\Omega\Omega\}, \quad \mathscr{G}_2 = \{x\mathbf{I}\Omega g\Omega\}, \quad \mathscr{G}_3 = \{x\mathbf{I}\Omega\Omega t\}.$$

Then the systems $\mathscr{S}_{3,i} \in \text{dev}(0, \mathscr{S}^2)$ constructed according to $\mathscr{G}_i$ $(i = 1, 2, 3)$ are:

$$\begin{aligned}\mathscr{S}_{3,1} = (\{&x\mathbf{I}(\lambda at.\, a\Omega_1 \ldots \Omega_5 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{14}.\, av_1 \ldots v_{14})zt)\\ &= z\mathbf{I}(\lambda at.\, a\Omega_1 \ldots \Omega_5 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{14}.\, av_1 \ldots v_{14})zt),\\ &\quad x\mathbf{I}(\lambda v_1 \ldots v_{14}.\, g)zt = g\mathbf{I}(\lambda v_1 \ldots v_{14}.\, g)zt\}, \{x\});\end{aligned}$$

$$\mathscr{S}_{3,2} = (\{x\mathbf{I}(\lambda at.\, z) = z\mathbf{I}(\lambda at.\, z), x\mathbf{I}a(z'\Omega_1 \ldots \Omega_4)t = z'\mathbf{I}a(z'\Omega_1 \ldots \Omega_4)t\}, \{x\})$$

(note that $\mathscr{S}_{3,2}$ is not β-suitable);

$$\begin{aligned}\mathscr{S}_{3,3} = (\{&x\mathbf{I}(\lambda at.\, t\Omega_1 \ldots \Omega_5 \vec{\mathbf{D}}(9)\mathbf{I}az(\lambda v_1 \ldots v_{14}.\, tv_1 \ldots v_{14}))\\ &= z\mathbf{I}(\lambda at.\, t\Omega_1 \ldots \Omega_5 \vec{\mathbf{D}}(9)\mathbf{I}az(\lambda v_1 \ldots v_{14}.\, tv_1 \ldots v_{14})),\\ &\quad x\mathbf{I}az(\lambda v_1 \ldots v_{14}.\, g) = g\mathbf{I}az(\lambda v_1 \ldots v_{14}.\, g)\}, \{x\}).\end{aligned} \qquad \diamond_0$$

Let $\mathscr{S}^4 \in \text{canonical}(\mathscr{S}^3)$. We show that $\mathscr{S}^4$ is PFR. Let $\mathfrak{F} = \text{unif}(\mathscr{S}^2) \cup \mathscr{G}$, $\mathfrak{F}' = \text{unif}(\mathscr{S}^4)$ and $\varphi : \mathfrak{F} \to \mathfrak{F}'$ the *natural* bijection between $\mathfrak{F}$ and $\mathfrak{F}'$. An example will be sufficient to define φ.

1. Example. Let $\mathscr{S}^2$ be as in Example 0. Hence $\text{unif}(\mathscr{S}^2) = \{x\mathbf{I}(\lambda a.\, x\mathbf{I}az)\Omega\Omega\}$. We use the same notation as in Example 0. Let $\mathfrak{F} = \text{unif}(\mathscr{S}^2) \cup \mathscr{G}_1 = \{x\mathbf{I}(\lambda a.\, x\mathbf{I}az)\Omega\Omega, x\mathbf{I}a\Omega\Omega\}$;

$$\begin{aligned}\mathscr{S}_{4,1} = (\{&x\mathbf{I}(\lambda at.\, a\Omega_1 \ldots \Omega_5 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{14}.\, av_1 \ldots v_{14})zt)\\ &= z\mathbf{I}(\lambda at.\, a\Omega_1 \ldots \Omega_5 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{14}.\, av_1 \ldots v_{14})zt),\\ &\quad x\mathbf{I}(\lambda v_1 \ldots v_{14}.\, g)\Omega\Omega = g\mathbf{I}(\lambda v_1 \ldots v_{14}.\, g)\Omega\Omega\}, \{x\}) \in \text{canonical}(\mathscr{S}_1^3);\end{aligned}$$

$$\begin{aligned}\mathfrak{F}' = \text{unif}(\mathscr{S}_{4,1}) = \{&x\mathbf{I}(\lambda at.\, a\Omega_1 \ldots \Omega_5 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{14}.\, av_1 \ldots v_{14})zt)\Omega\Omega,\\ &x\mathbf{I}(\lambda v_1 \ldots v_{14}.\, g)\Omega\Omega\}.\end{aligned}$$

Then the natural bijection $\varphi : \mathfrak{F} \to \mathfrak{F}'$ is defined as follows: $\varphi(x\mathbf{I}(\lambda a.\, x\mathbf{I}az)\Omega\Omega) = x\mathbf{I}(\lambda at.\, a\Omega_1 \ldots \Omega_5 \vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1 \ldots v_{14}.\, av_1 \ldots v_{14})zt)\Omega\Omega$, $\varphi(x, \mathbf{I}a\Omega\Omega) = x\mathbf{I}(\lambda v_1 \ldots v_{14}.\, g)\Omega\Omega$. $\diamond_1$

Let $M \in \mathfrak{F}$ and $\alpha \in \mathrm{Seq}$. Then α has a natural image $\delta(M, \alpha)$ in $\varphi(M)$. An example will be sufficient to define $\delta(M, \alpha)$.

2. Example. Let $\mathfrak{F}$ and $\mathfrak{F}'$ be as in Example 1. Let $M = x\mathbf{I}(\lambda a.\, x\mathbf{I}az)\Omega\Omega \in \mathfrak{F}$. Then: $\delta(M, \langle\ \rangle) = \langle\ \rangle$, $\delta(M, \langle 0 \rangle) = \langle 0 \rangle$, $\delta(M, \langle 1 \rangle) = \langle 1 \rangle$, $\delta(M, \langle 1, 0 \rangle) = \langle 1, 14 \rangle$, $\delta(M, \langle 1, 1 \rangle) = \langle 1, 15 \rangle$, $\delta(M, \langle 1, 2 \rangle) = \langle 1, 16 \rangle$, $\delta(M, \langle 1, 1, 1 \rangle) = \langle 1, 15, 1 \rangle$, $\delta(M, \langle 2 \rangle) = \langle 2 \rangle$. Let $N = x\mathbf{I}a\Omega\Omega \in \mathfrak{F}$. Then: $\delta(N, \langle\ \rangle) = \langle\ \rangle$, $\delta(N, \langle 0 \rangle) = \langle 0 \rangle$, $\delta(N, \langle 1 \rangle) = \langle 1 \rangle$, $\delta(N, \langle 2 \rangle) = \langle 2 \rangle$.

By definition we have: $\forall M \in \mathfrak{F}\ \forall \alpha \in \mathrm{Seq}\,[M|\alpha\downarrow \text{ iff } \varphi(M)|\delta(M,\alpha)\downarrow]$.

Part 1. Let $\mathscr{B} \subseteq \mathfrak{F}$ and α agt for $\mathscr{B}$. We prove that

$$[[\exists M, N \in \mathscr{B}\, \delta(M, \alpha) \neq \delta(N, \alpha)] \Rightarrow \exists \beta \text{ usf and agt for } \varphi(\mathscr{B})].$$

Let $P, Q \in \mathscr{B}$ s.t. $\delta(P, \alpha) \neq \delta(Q, \alpha)$. We prove that $\exists \beta$ usf and agt for $\varphi(\mathscr{B})$. Our hypothesis implies $\alpha \neq \langle\ \rangle$. Let θ be the longest node s.t. $\forall M \in \mathscr{B}\ \delta(M, \alpha) = \theta^* \gamma_M$. Since α is agt for $\mathscr{B}$ we have: $\forall M, N \in \mathscr{B}$ [length$(\delta(M, \alpha))$ = length$(\delta(N, \alpha))$]. We have: $\forall M \in \mathscr{B}[\gamma_M = \langle i_M \rangle^* \rho_M]$. W.l.o.g. we can assume $i_P = \min\{i_M | M \in \mathscr{B}\} < i_Q$. Let $M \in \mathscr{B}$, we have: $\varphi(M)_\theta = \lambda \vec{a}_M t_{q(M)+1} \ldots t_{\mathrm{sat}(\mathscr{S},x)}.\ b_M \Omega_1 \ldots \Omega_{h(M)} \Omega_{r(M)+1} \ldots \Omega_{\mathrm{top1}} \vec{\mathbf{D}}$ $(k_M \mathrm{top2})\ L_{M,1} \ldots L_{M,q(M)} t_{q(M)+1} \ldots t_{\mathrm{sat}(\mathscr{S},x)}$.

There exists $m \in \mathbb{N}$ (univocally determined by α) s.t. $\forall M \in \mathscr{B}\ [i_M = h(M) - r(M) + \mathrm{top1} + k_M \mathrm{top2} + m]$. Thus $h(P) - r(P) + \mathrm{top1} + k_P \mathrm{top2} < h(Q) - r(Q) + \mathrm{top1} + k_Q \mathrm{top2}$. Let $m^* = h(P) - r(P) + \mathrm{top1} + k_P \mathrm{top2} = \min\{h(M) - r(M) + \mathrm{top1} + k_M \mathrm{top2} | M \in \mathscr{B}\}$.

Suppose that there exists $L \in \mathscr{B}$ s.t. $m^* < h(L) - r(L) + \mathrm{top1}$. Then:

$$\mathrm{top2} < h(L) - r(L) + r(P) - h(P) \leqslant h(L) + r(P) \leqslant \mathrm{degrgh}(\mathscr{S}) + \mathrm{ordrgh}(\mathscr{S}).$$

Absurd, since $\mathrm{top2} = 2\,[\mathrm{degrgh}(\mathscr{S}) + \mathrm{ordrgh}(\mathscr{S})] + 1$.
Thus $\forall L \in \mathscr{B}[h(L) - r(L) + \mathrm{top1} \leqslant m^* \leqslant h(L) - r(L) + \mathrm{top1} + k_L \mathrm{top2}]$.
This implies that $\forall M \in \mathscr{B}[M | \theta^* \langle m^* \rangle \downarrow]$ and (A.5.3) $\exists n, b \in \mathbb{N}\ [\varphi(P)_{\theta^* \langle m^* \rangle} = \mathbf{U}^n_b$ and $[\varphi(Q)_{\theta^* \langle m^* \rangle} = \mathbf{D}_1 = \lambda ab.\, bab]$. Hence $\neg\, [\varphi(P) \sim_{\theta^* \langle m^* \rangle} \varphi(Q)]$ (see [8, A.0.0]).
Thus $\theta^* \langle m^* \rangle$ is usf for $\mathscr{B}$. Hence there exists $\beta \leqslant \theta^* \langle m^* \rangle$ s.t. β is usf and agt for $\mathscr{B}$.

Part 2. We show that $\mathfrak{F}'$ is distinct. This implies that $\mathscr{S}^4$ is PFR.
Suppose that $\mathfrak{F}'$ is not distinct. Then there exists $\mathfrak{E} \in \mathfrak{F}'/\mathrm{ind}(\mathfrak{F}')$ (see [8, A.0.0]) s.t. $\mathrm{Card}(\mathfrak{E}) > 1$. However, since $\mathfrak{F}$ is distinct, $\varphi^{-1}(\mathfrak{E})$ is distinct. Thus there exists α usf and agt for $\varphi^{-1}(\mathfrak{E})$. Hence $\forall M \in \varphi^{-1}(\mathfrak{E})[\varphi(M)|\delta(M, \alpha)\downarrow]$.

Case 0: $\exists M, N \in \varphi^{-1}(\mathfrak{E})[\delta(M, \alpha) \neq \delta(N, \alpha)]$. Then, by part 0, $\exists \beta$ usf and agt for $\mathfrak{E}$. Contradiction, since $\mathfrak{E} \in \mathfrak{F}'/\mathrm{ind}(\mathfrak{F}')$ implies $\neg \exists \beta$ usf and agt for $\mathfrak{E}$.

Case 1. $\forall M, N \in \varphi^{-1}(\mathfrak{E})[\delta(M, \alpha) = \delta(N, \alpha)]$.

Let $\theta \in \mathrm{Seq}$ s.t. $\forall M \in \varphi^{-1}(\mathfrak{E})\delta(M, \alpha) = \theta$. We have $\mathfrak{E}|\theta\downarrow$ and hence $\forall M, N \in \varphi^{-1}(\mathfrak{E})[\varphi(M) \sim_\theta \varphi(N)]$.
Let $M_1, M_2 \in \varphi^{-1}(\mathfrak{E})$. We show that $[\mathrm{ind}(\mathfrak{E}, \varphi(M_1), \varphi(M_2)) \Rightarrow M_1 \sim_\alpha M_2]$.
Since α is usf and agt for $\varphi^{-1}(\mathfrak{E})$ this implies $\exists M, N \in \varphi^{-1}(\mathfrak{E})\ \neg\mathrm{ind}(\mathfrak{E}, \varphi(M), \varphi(N))$.
This is a contradiction since $\mathfrak{E} \in \mathfrak{F}'/\mathrm{ind}(\mathfrak{F}')$. Note that by the structure of $\mathfrak{F}$ $\alpha \neq \langle\ \rangle$.

Case 1.0: $(M_1)_\alpha = \lambda\vec{a}_1 . xQ_{1,1} \dots Q_{1,q(1)}$, $(M_2)_\alpha = \lambda\vec{a}_2 . uQ_{2,1} \dots Q_{2,q(2)}$, $\neg u \equiv x$. Thus $\varphi(M_1)_\theta = \lambda\vec{a}_1 t_{q(1)+1} \dots t_{\mathrm{sat}(\mathscr{S},x)}$.

$$b\Omega_1 \dots \Omega_{h(1)}\Omega_{r(1)+1} \dots \Omega_{\mathrm{top}1}\, \vec{\mathbf{D}}(k_1\ \mathrm{top}2)Q_{1,1} \dots Q_{1,q(1)}t_{q(1)+1} \dots t_{\mathrm{sat}(\mathscr{S},x)} \quad \text{and}$$

$$\varphi(M_2)_\theta = \lambda\vec{a}_2\vec{c}.\, uQ_{2,1} \dots Q_{2,q(2)}\vec{c}.$$

From $\mathrm{ind}(\mathfrak{E}, \varphi(M_1), \varphi(M_2))$ it follows $\mathrm{head}(\varphi(M_1)_\theta) = \mathrm{head}(\varphi(M_2)_\theta)$. We have: $\varphi(M_1) \sim_\theta \varphi(M_2) \Rightarrow [\deg(\varphi(M_1)_\theta) - \mathrm{ord}(\varphi(M_1)_\theta) = \deg(\varphi(M_2)_\theta) - \mathrm{ord}(\varphi(M_2)_\theta) \Rightarrow [\mathrm{sat}(\mathscr{S}, x) + \mathrm{top}1 + k_1\,\mathrm{top}2 - r(1) + h(1) - \mathrm{sat}(\mathscr{S}, x) + q(1) - |\vec{a}_1| = q(2) - |\vec{a}_2|]$ $\Rightarrow$ $\mathrm{top}1 + k_1\,\mathrm{top}2 = q(2) - |\vec{a}_2| + r(1) - h(1) - q(1) + |\vec{a}_1| \leqslant q(2) + |a_1| + r(1) \leqslant \mathrm{degrgh}(\mathscr{S}) + 2\mathrm{ordrgh}(\mathscr{S})$.

This is absurd since $\mathrm{top}1 + \mathrm{top}2 = 3[\mathrm{degrgh}(\mathscr{S}) + \mathrm{ordrgh}(\mathscr{S})] + 2 + p$ and $k_1 \geqslant 1$.

Case 1.1: $(M_i)_\alpha = \lambda\vec{a}_i . xQ_{i,1} \dots Q_{i,q(i)}$, $i = 1, 2$. Thus $(i = 1, 2)$ $\varphi(M_i)_\theta = \lambda\vec{a}_i t_{q(i)+1} \dots t_{\mathrm{sat}(\mathscr{S},x)}.\ b_i\Omega_1 \dots \Omega_{h(i)}\Omega_{r(i)+1} \dots \Omega_{\mathrm{top}1}\, \vec{\mathbf{D}}(k_i\,\mathrm{top}2)Q_{i,1} \dots Q_{i,q(i)}t_{q(i)+1} \dots t_{\mathrm{sat}(\mathscr{S},x)}$.
From $\varphi(M_1) \sim_\theta \varphi(M_2)$ it follows $\mathrm{head}(\varphi(M_1)_\theta) = \mathrm{head}(\varphi(M_2)_\theta)$. We have: $\varphi(M_1) \sim_\theta \varphi(M_2) \Rightarrow [b_1 \equiv b_2$ and $\mathrm{top}1 + k_1\,\mathrm{top}2 - r(1) + h(1) + q(1) - |\vec{a}_1| = \mathrm{top}1 + k_2\,\mathrm{top}2 - r(2) + h(2) + q(2) - |\vec{a}_2|]$.

Case 1.1.0: $k_1 > k_2$. Let $k_1 = k_2 + j$ $(j > 0)$ $\therefore$ We have: $\mathrm{top}1 + k_2\,\mathrm{top}2 + j\,\mathrm{top}2 - r(1) + h(1) + q(1) - |\vec{a}_1| = \mathrm{top}1 + k_2\,\mathrm{top}2 - r(2) + h(2) + q(2) - |\vec{a}_2| \Rightarrow j\,\mathrm{top}2 = r(1) - h(1) - q(1) + |\vec{a}_1| - r(2) + h(2) + q(2) - |\vec{a}_2| \leqslant r(1) + |\vec{a}_1| + h(2) + q(2) \leqslant 2[\mathrm{ordrgh}(\mathscr{S}) + \mathrm{degrgh}(\mathscr{S})]$.
Absurd since $\mathrm{top}2 = 2[\mathrm{ordrgh}(\mathscr{S}) + \mathrm{degrgh}(\mathscr{S})] + 1$.

Case 1.1.1: $k_1 < k_2$. Analogous to case 1.1.0.

Case 1.1.2: $k_1 = k_2 = k$. Then $r(1) - h(1) - q(1) + |\vec{a}_1| = r(2) - h(2) - q(2) + |\vec{a}_2|$. Let $\varphi^{-1}(\mathfrak{E}) = \{M_1, M_2, \dots, M_m\}$. Thus $\forall i \in \{1, \dots, m\}$ we have: (taking into account cases 1.1.0, 1)

$$\varphi(M_i)_\theta = \lambda\vec{a}_i t_{q(i)+1} \dots\ t_{\mathrm{sat}(\mathscr{S},x)}.\ b_i\Omega_1 \dots \Omega_{h(i)}\Omega_{r(i)+1} \dots \Omega_{\mathrm{top}1}\, \vec{\mathbf{D}}(k\ \mathrm{top}2)$$
$$Q_{i,1} \dots Q_{i,q(i)}t_{q(i)+1} \dots t_{\mathrm{sat}(\mathscr{S},x)}.$$

Let $H = \min\{h(s) - r(s) \mid s \in \{1, \dots, m\}\}$ and $H^* = \mathrm{top}1 + k\,\mathrm{top}2 + H$.
$\forall i \in \{1, \dots, m\}$ define $H(i) = k\,\mathrm{top}2 + H - (h(i) - r(i)) + 1$. Let $i \in \{1, \dots, m\}$. By the choice of H we have: $H - (h(i) - r(i)) \leqslant 0$ and $\mathrm{top}1 + h(i) - r(i) + k\,\mathrm{top}2 \geqslant \mathrm{top}1 + H + k\,\mathrm{top}2$. Thus $H(i) \leqslant k\,\mathrm{top}2 + 1$. We show that $H(i) \geqslant 1$. Suppose $H(i) < 1$. Then: $1 > k\,\mathrm{top}2 + H - (h(i) - r(i)) + 1 \geqslant k\,\mathrm{top}2 - \mathrm{ordrgh}(\mathscr{S}) - \mathrm{degrgh}(\mathscr{S}) \geqslant \mathrm{top}2 - \mathrm{ordrgh}(\mathscr{S}) - \mathrm{degrgh}(\mathscr{S}) = \mathrm{degrgh}(\mathscr{S}) + \mathrm{ordrgh}(\mathscr{S}) + 1 \geqslant 1$. Contradiction. Hence $H(i) \geqslant 1$ and $H(i) - 1 = H + k\,\mathrm{top}2 - (h(i) - r(i)) \geqslant 0$. Thus: $\forall i \in \{1, \dots, m\}$ $[\mathrm{top}1 + h(i) - r(i) \leqslant \mathrm{top}1 + H + k\,\mathrm{top}2 = H^* \leqslant \mathrm{top}1 + h(i) - r(i) + k\,\mathrm{top}2]$. Hence $\forall i \in \{1, \dots, m\}$ $\exists a, b \in \mathbb{N}$ s.t.: (see 2.0)
$\varphi(M_i)_{\theta^* \langle H^* \rangle} = \textbf{if } h(i) - r(i) > H \textbf{ then } \mathbf{D}_1 \textbf{ else } \mathbf{U}^a_b$.

Case 1.1.2.0: $\forall i, s \in \{1, \dots, m\}$ $H(i) = H(s)$. Then $\forall i \in \{1, \dots, m\}$ $[H = h(i) - r(i)$ and $H(i) = k\ \mathrm{top}2 + 1]$. This implies $q(1) - |\vec{a}_1| = q(2) - |\vec{a}_2|$ and hence $M_1 \sim_\alpha M_2$. Contradiction.

Case 1.1.2.1: $\exists i, s \in \{1, \ldots, m\}$ $H(i) \neq H(s)$. Then $\exists s \in \{1, \ldots, m\}$ $H < h(s) - r(s)$. Let $i, s \in \{1, \ldots, m\}$ s.t. $[H = h(i) - r(i)$ and $H < h(s) - r(s)]$. Then $\exists a, b \in \mathbb{N}$ s.t.: $\varphi(M_s)_{\theta^* \langle H^* \rangle} = \mathbf{D}_1$ and $\varphi(M_i)_{\theta^* \langle H^* \rangle} = \mathbf{U}_b^a]$.
Thus $\theta^* \langle H^* \rangle$ is usf and agt for $\mathfrak{C}$. Hence $\exists \beta \leqslant 0^* \langle H^* \rangle$ usf and agt for $\mathfrak{C}$. Contradiction.

Case 1.2: $(M_i)_\alpha = \lambda \vec{a}_1 . u_i Q_{i,1} \ldots Q_{i,q(i)}$ $(i = 1, 2)$ and $u_1, u_2 \notin \{x\}$. Thus $(i = 1, 2)$ $\varphi(M_i)_\theta = \lambda \vec{a}_i . \vec{b}_i . u_i Q_{i,1} \ldots Q_{i,q(i)} \vec{b}_i$. From $\varphi(M_1) \sim_\theta \varphi(M_2)$ we have: $[u_1 \equiv u_2$ and $q(1) - |\vec{a}_1| = q(2) - |\vec{a}_2|]$. Hence $M_1 \sim_\alpha M_2$. Contradiction. □

The necessary condition in 2.0.2 becomes sufficient for a relaxation of a GSL-system $\mathscr{S}$ (A.2.2, A.5.5, A.5.6).

A.5.5. Theorem. *Let* $\mathscr{S} = (\Gamma, \{x\})$ *be a normal GSL-system. If* $\exists \mathscr{S}^1 \in \mathrm{canonical}(\mathscr{S})$ $\exists \mathscr{S}^2 \in \mathrm{approx}(\mathscr{S}^1)$ $[\mathscr{S}^2$ *is GPFR and* β*-suitable*$]$ *then* $\exists \mathscr{P} \in \mathrm{relax}(\mathscr{S})$ $\exists \mathscr{P}^1 \in \mathrm{canonical}(\mathscr{P})$ $\exists \mathscr{P}^2 \in \mathrm{approx}(\mathscr{P}^1)$ $\exists \mathscr{P}^3 \in \mathrm{dev}(\mathscr{P}^2)$ $\exists \mathscr{P}^4 \in \mathrm{canonical}(\mathscr{P}^3)$ $[\mathrm{PFRLR}^*(\mathscr{P}^4)$ *is* $(\mathrm{right}(\mathscr{P}^4), \emptyset, 0)$-distinct$]$.

Proof. We divide the proof into two parts.

Part 0. We give an algorithm to construct $\mathscr{P}$ from $\mathscr{S}$. Let $\mathscr{S}^1 \in \mathrm{canonical}(\mathscr{S})$ and $\mathscr{S}^2 \in \mathrm{approx}(\mathscr{S}^1)$ s.t. $[\mathscr{S}^2$ is GPFR and β-suitable$]$. Let (see A.0.1, A.1.2) $\mathfrak{G}$ an $\{x\}$-version of $\mathrm{crit}(\mathrm{sl}(\mathscr{S}^2))$ s.t. $(\mathrm{unif}(\mathrm{sl}(\mathscr{S}^2)) \cup \mathfrak{G})$ is distinct (such a $\mathfrak{G}$ exists since $\mathscr{S}^2$ is GPFR). Let $\mathscr{S}^3 \in \mathrm{dev}(0, \mathscr{S}^2)$ constructed according to $\mathfrak{G}$ (see Example 0 in the proof of A.5.4). Let $\mathscr{S}^4 \in \mathrm{canonical}(\mathscr{S}^3)$. From A.5.4 $\mathscr{S}^4$ is PFR, but in general it is not β-suitable (see Example 1 in the proof of A.5.4). The following definition will be useful.

0. Definition. Let $m \in \mathbb{N}$, $M \in \Lambda$, $z \in \mathrm{Var}$ and $\sigma \in \mathrm{Seq}$.

0. The node σ is said to be m-safe in M iff $\forall \beta \leqslant \sigma \deg(M_\beta) \leqslant m$.

1. The variable z is said to be m-safe in M iff $\forall \theta \in \mathrm{BT}(M)$ $[\mathrm{head}(M_\theta) \equiv z \Rightarrow \theta$ is m-safe in $M]$.

Let $\mathscr{P} \in \mathrm{relax}(\mathscr{S})$ (arbitrarily chosen). Then we can find $\mathscr{P}^1 \in \mathrm{canonical}(\mathscr{P})$ and $\mathscr{P}^2 \in \mathrm{approx}(\mathscr{P}^1)$ s.t. $[\mathscr{P}^2$ is GPFR and β-suitable$]$. Thus there exists $\mathfrak{G}^1 \in \mathrm{version}(\{x\}, \mathrm{crit}(\mathrm{sl}(\mathscr{P}^2)))$ s.t. $(\mathrm{unif}(\mathrm{sl}(\mathscr{P}^2)) \cup \mathfrak{G}^1)$ is distinct. Moreover, for any $p \in \mathbb{N}$ we can construct $\mathscr{P}^3 \in \mathrm{dev}(p, \mathscr{P}^2)$ according to $\mathfrak{G}^1$ and $\mathscr{P}^4 \in \mathrm{canonical}(\mathscr{P}^3)$.

Looking at $\mathscr{S}^4$ we can find (in Polynomial Time) $\mathscr{P} \in \mathrm{relax}(\mathscr{S})$ and $p \in \mathbb{N}$ s.t.:

P0. When we use the algorithm in A.2.8.0 to construct $\mathscr{P}^3 \in \mathrm{dev}(p, \mathscr{P}^2)$ we always have: (using the same notation as in A.2.8.0) $j = \mathrm{top} + \mathrm{sat}(\mathscr{P}^2, x)$;

P1. $\forall x\vec{M} = z\vec{M} \in \mathscr{P}^4$ $[z$ is $\mathrm{degrgh}(\mathscr{P}^2)$-safe in $(x\vec{M}) \Rightarrow \mathrm{ord}(x\vec{M}, z) = \mathrm{top1} + \mathrm{top2}]$.

An example will clarify the situation.

1. Example. Let $\mathscr{S} = (\{x\mathbf{I}(\lambda b.\, x\mathbf{I}(b\,b))y = y\mathbf{I}(\lambda b.\, x\mathbf{I}(b\,b))y,\ x\mathbf{I}(\lambda b.\, x\mathbf{I}(\lambda a_1 \ldots a_{33}.\, z))\Omega = z\mathbf{I}(\lambda b.\, x\mathbf{I}(\lambda a_1 \ldots a_{33}.\, z))\Omega\}, \{x\})$.

Let $\mathscr{S}^1 = \mathscr{S}^2 = \mathscr{S}$, $p = 0$,

$$\mathscr{S}^3 = (\{x\mathbf{I}(\lambda bta_{18}\ldots a_{33}.\, z)\Omega = z\mathbf{I}(\lambda bta_{18}\ldots a_{33}.\, z)\Omega,$$

$$x\mathbf{I}(\lambda a_1\ldots a_{14}.\, z')t = z'\mathbf{I}(\lambda a_1\ldots a_{14}.\, z')t,$$

$$x\mathbf{I}(\lambda v_1\ldots v_{14}.\, g)t' = g\mathbf{I}(\lambda v_1\ldots v_{14}.\, g)t',$$

$$x\mathbf{I}(\lambda bt.\, b\Omega_1\Omega_1\ldots\Omega_5\vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1\ldots v_{14}.\, bbv_1\ldots v_{14})t)y$$

$$= y\mathbf{I}(\lambda bt.\, b\Omega_1\Omega_1\ldots\Omega_5\vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1\ldots v_{14}.\, bbv_1\ldots v_{14})t)y\}, \{x\}) \in \mathrm{dev}(0, \mathscr{S}^2),$$

$$\mathscr{S}^4 = (\{x\mathbf{I}(\lambda bta_{18}\ldots a_{33}.\, z)\Omega = z\mathbf{I}(\lambda bta_{18}\ldots a_{33}.\, z)\Omega,$$

$$x\mathbf{I}(\lambda a_1\ldots a_{14}.\, z')\Omega = z'\mathbf{I}(\lambda a_1\ldots a_{14}.\, z')\Omega,$$

$$x\mathbf{I}(\lambda v_1\ldots v_{14}.\, g)\Omega = g\mathbf{I}(\lambda v_1\ldots v_{14}.\, g)\Omega,$$

$$x\mathbf{I}(\lambda bt.\, b\Omega_1\Omega_1\ldots\Omega_5\vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1\ldots v_{14}.\, bbv_1\ldots v_{14})t)y$$

$$= y\mathbf{I}(\lambda bt.\, b\Omega_1\Omega_1\ldots\Omega_5\vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1\ldots v_{14}.\, bbv_1\ldots v_{14})t)y\}, \{x\}).$$

We choose

$$\mathscr{P} = (\{x\mathbf{I}(\lambda b.xI(bb))(\lambda a_1\ldots a_{17}.\, y) = y\mathbf{I}(\lambda b.x\mathbf{I}(bb))(\lambda a_1\ldots a_{17}.\, y),$$

$$x\mathbf{I}(\lambda b.x\mathbf{I}(a_1\ldots a_{35}.\, z))\Omega = z\mathbf{I}(\lambda b.x\mathbf{I}(a_1\ldots a_{35}.\, z))\Omega\}, \{x\}),$$

$\mathscr{P}^1 = \mathscr{P}^2 = \mathscr{P}$, $p = 3$ (i.e. $\mathscr{P}^3 \in \mathrm{dev}(3, \mathscr{P}^2)$),

$$\mathscr{P}^3 = (\{x\mathbf{I}(\lambda bta_{21}\ldots a_{35}.\, z))\Omega = z\mathbf{I}(\lambda bta_{21}\ldots a_{35}.\, z)\Omega,$$

$$x\mathbf{I}(\lambda a_1\ldots a_{17}.\, z')t = z'\mathbf{I}(\lambda a_1\ldots a_{17}.\, z')t,$$

$$x\mathbf{I}(\lambda v_1\ldots v_{17}.\, g)t' = g\mathbf{I}(\lambda v_1\ldots v_{17}.\, g)t',$$

$$x\mathbf{I}(\lambda bt.\, b\Omega_1\Omega_1\ldots\Omega_8\vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1\ldots v_{17}.\, bbv_1\ldots v_{17})t)(\lambda a_1\ldots a_{17}.\, y)$$

$$= y\mathbf{I}(\lambda bt.\, b\Omega_1\Omega_1\ldots\Omega_8\vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1\ldots v_{17}.\, bbv_1\ldots v_{17})t)(\lambda a_1\ldots a_{17}.\, y)\}, \{x\}),$$

$$\mathscr{P}^4 = (\{x\mathbf{I}(\lambda bta_{21}\ldots a_{35}.\, z)\Omega = z\mathbf{I}(\lambda bta_{21}\ldots a_{35}.\, z)\Omega,$$

$$x\mathbf{I}(\lambda a_1\ldots a_{17}'.\, z')\Omega = z'\mathbf{I}(\lambda a_1\ldots a_{17}.\, z')\Omega,$$

$$x\mathbf{I}(\lambda v_1\ldots v_{17}.\, g)\Omega = g\mathbf{I}(\lambda v_1\ldots v_{17}.\, g)\Omega,$$

$$x\mathbf{I}(\lambda bt.\, b\Omega_1\Omega_1\ldots\Omega_8\vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1\ldots v_{17}.\, bbv_1\ldots v_{17})t)(\lambda a_1\ldots a_{17}.\, y)$$

$$= y\mathbf{I}(\lambda bt.\, b\Omega_1\Omega_1\ldots\Omega_8\vec{\mathbf{D}}(9)\mathbf{I}(\lambda v_1\ldots v_{17}.\, bbv_1\ldots v_{17})t)(\lambda a_1\ldots a_{17}.\, y)\}, \{x\}).$$

Then $\mathscr{P} \in \mathrm{relax}(\mathscr{S})$ and $\mathscr{P}^3$, $\mathscr{P}^4$ satisfy P0 and P1.

Note that we need the parameter $p \in \mathbb{N}$ to be able to satisfy P1.

Part 1. We show that PFRLR*($\mathscr{P}^4$) is ($\mathrm{right}(\mathscr{P}^4)$, $\emptyset$, 0)-distinct (see [8, A.2.1.4]).
Let $\mathfrak{F} \subset_{\mathrm{f}} \Lambda$ and $Z \subset_{\mathrm{f}} \mathrm{Var}$. We define $\mathrm{ind}(\mathfrak{F}, Z)$ as follows:
$\mathrm{ind}(\mathfrak{F}, Z) =$ **if** $\mathrm{Card}(\mathfrak{F}) = 1$

then if $[Z \neq \emptyset \Rightarrow \exists\alpha$ usf and agt for $\mathfrak{F}$ $\forall M \in \mathfrak{F}$ [head$(M_\alpha) \in Z$ and deg$(M_\alpha) = 0]$
  then $\{\mathfrak{F}\}$
  else $\{\emptyset\}$
else if $\exists\alpha$ usf and agt for $\mathfrak{F}$ [α is $(Z, \emptyset, 0)$-safe in $\mathfrak{F}$]
  then $\cup \{\text{ind}(\mathscr{B}, Z) \mid \mathscr{B} \in \mathfrak{F}/\sim_\alpha\}$
  else $\{\mathfrak{F}\}$.

Let $\mathfrak{F} = \text{PFRLR}^*(\mathscr{P}^4)$ and $Z = \text{right}(\mathscr{P}^4)$. Then ind$(\mathfrak{F}, Z)$ is a partition of $\mathfrak{F}$. Suppose that $\mathfrak{F}$ is not $(Z, \emptyset, 0)$-distinct. Then there exists $\mathfrak{E} \in \text{ind}(\mathfrak{F}, Z)$ s.t. Card$(\mathfrak{E}) > 1$. Since $\mathfrak{F}$ is distinct there is α usf and agt for $\mathfrak{E}$. However, since $\mathfrak{E} \in \text{ind}(\mathfrak{F}, Z)$ such an α is not $(Z, \emptyset, 0)$-safe in $\mathfrak{E}$. We show that if α is usf and agt for $\mathfrak{E}$ then α is $(Z, \emptyset, 0)$-safe in $\mathfrak{E}$. This contradiction implies that Card$(\mathfrak{E}) = 1$, i.e. $\mathfrak{F}$ is $(Z, \emptyset, 0)$-distinct.

Let α usf and agt for $\mathfrak{E}$ s.t. α is not $(Z, \emptyset, 0)$-safe in $\mathfrak{E}$. Then (see A.2.5 and [8, A.2.1.4]) there exists $N = \langle x, N_1, \ldots, N_m, H_1 \rangle \in \mathfrak{E}$ s.t. $N_\alpha = \lambda\vec{a}.z$ and $z \in Z$ and ord$(N_\alpha) <$ rad$(\emptyset, 0, \mathfrak{E}, \alpha)$. Thus there exists $Q = \langle x, Q_1, \ldots, Q_m, H_2 \rangle \in \mathfrak{E}$ s.t. rad$(\emptyset, 0, \mathfrak{E}, Q, \alpha) >$ ord(N_α).
Note that by the construction of $\mathscr{P}$ we have, for some $k \geqslant 0$, ord$(N_\alpha) = \text{top1} + (k+1)\,\text{top2}$.

Case 0. $Q_\alpha = \lambda\vec{b}.\,u\vec{L}$ and $u \notin Z$. Since α is agt for $\mathfrak{E}$ (and taking into account case 1.0 in the proof of A.5.4) we have: (see [8, A.2.1.0]) rad$(\emptyset, 0, \mathfrak{E}, Q, \alpha) \leqslant \text{degrgh}(\mathscr{S}) + \text{top1} + k\,\text{top2} + \text{sat}(\mathscr{S}, x) + 1 + \text{ordrgh}(\mathscr{S}) \leqslant \text{top1} + k\,\text{top2} + 2[\text{degrgh}(\mathscr{S}) + \text{ordrgh}(\mathscr{S})] + 1 = \text{top1} + (k+1)\,\text{top2} = \text{ord}(N_\alpha)$. Contradiction.

Case 1. $Q_\alpha = \lambda\vec{b}.u$ and $u \in Z$. Then ord$(Q_\alpha) = \text{top1} + (k+1)\,\text{top2} = \text{ord}(N_\alpha)$. Contradiction. Thus $\mathfrak{F}$ is $(Z, \emptyset, 0)$-distinct, i.e. PFRLR*$(\mathscr{P}^4)$ is (right$(\mathscr{P}^4), \emptyset, 0)$-distinct. □

A.5.6. Corollary. *Let* $\mathscr{S} = (\Gamma, \{x\})$ *be a normal GSL-system. If* $\exists\mathscr{S}^1 \in \text{canonical}(\mathscr{S})$ $\exists\mathscr{S}^2 \in \text{approx}(\mathscr{S}^1)$ [$\mathscr{S}^2$ *is GPFR and* β*-suitable*] *then* $\exists\mathscr{P} \in \text{relax}(\mathscr{S})$ [$\mathscr{P}$ *is* β*-solvable*].

Proof. From A.5.5 and A.4.2. □

The following corollary completes the Proof of A.5.0.

A.5.7. Corollary. *Let* $\mathscr{S} = (\Gamma, X)$ *be an SL-system.*

0. $\exists\mathscr{S}^1 \in \text{canonical}(\mathscr{S})$ $\exists\mathscr{S}^2 \in \text{approx}(\mathscr{S}^1)$ [$\mathscr{S}^2$ *is GPFR and* β*-suitable*] *iff* $\exists\mathscr{P} \in \text{relax}(\mathscr{S})$ NP_OK_SUFF$(\mathscr{P})$.

1. $\exists\mathscr{S}^1 \in \text{canonical}(\mathscr{S})$ $\exists\mathscr{S}^2 \in \text{approx}(\mathscr{S}^1)$ [$\mathscr{S}^2$ *is GPFR and* β*-suitable*] *iff* $\exists\mathscr{P} \in \text{relax}(\mathscr{S})$ [$\mathscr{P}$ *is* β*-solvable*].

Proof. 0. ($\Rightarrow$) Reasoning as in the proof of A.5.5 we can find $\mathscr{P} \in \text{relax}(\mathscr{S})$ s.t. $\exists\mathscr{P}^1 \in \text{canonical}(\text{gsl}^*(\text{single}(\text{Relax\_asg\_like}(\mathscr{S}))))$ $\exists\mathscr{P}^2 \in \text{approx}(\mathscr{P}^1)$ $\exists\mathscr{P}^3 \in \text{dev}(\mathscr{P}^2)$ $\exists\mathscr{P}^4 \in \text{canonical}(\mathscr{P}^3)$ [PFRLR*$(\mathscr{P}^4)$ is (right$(\mathscr{P}^4), \emptyset, 0)$-distinct)].

The thesis follows from 2.0.3, A.2.0.

($\Leftarrow$) From 2.0.2, A.0.6, A.5.2.

1. From 0, 2.0.2 and 2.0.3. $\square$

The following result (A.5.8) shows that the X-separability problem (see [8, 3.1]) for λ-free sets is decidable in any sms theory. This result was proved in [2], [3, Theorem 2.7] and [6, Theorem 2.1]. However, all together, the proof presented here is shorter then a full version of the proof sketched in [2, 3, 6].

Note that the X-separability problem [8, 3.1] for a λ-free set in the β-solvability problem for an SL-system satisfying the hypotheses of A.5.8.

A.5.8. Corollary. *Let $\mathbb{T}$ be a sms theory (e.g. β or $\beta\eta$) and $\mathscr{S} = (\Gamma, \{x_1, \ldots, x_n\})$ be an SL-system s.t.:*

H0. $\vec{y} \equiv y_0, \ldots, y_{k-1} \in (\mathrm{Var} - \{x_1, \ldots, x_n\})$ *is asg-like in* $\mathscr{S}$.

H1. *Each equation in $\mathscr{S}$ has form $x\vec{M} = y$, where $y \in \{\vec{y}\}$ and the variables in* $\mathrm{right}(\mathscr{S})$ *are pairwise distinct.*

Then $\mathscr{S}$ is $\mathbb{T}$-solvable iff $\exists \mathscr{S}^1 \in \mathrm{canonical}(\mathscr{S})\ \exists \mathscr{S}^2 \in \mathrm{approx}(\mathscr{S}^1)$ [$\mathscr{S}^2$ *is* GPFR]

Proof. ($\Rightarrow$) By 2.0.2.

($\Leftarrow$) By the structure of $\mathscr{S}$ we have: $\exists \mathscr{S}^1 \in \mathrm{canonical}(\mathscr{S})\ \exists \mathscr{S}^2 \in \mathrm{approx}(\mathscr{S}^1)$ [$\mathscr{S}^2$ is GPFR and β-suitable]. Thus, by A.5.7, $\exists \mathscr{P} \in \mathrm{relax}(\mathscr{S})$ [$\mathscr{P}$ is β-solvable]. Then for a suitable $f : \{\vec{y}\} \to \mathbb{N}$ we have: $R(\vec{y}, f)[\mathscr{S}] = \mathscr{P}$. Hence, by A.4.6, $\mathscr{S}$ is β-solvable. $\square$

Appendix B. Proof of 2.5

We show (B.0) that the X-separability problem for λ-free sets (see [8, 3.1]) is NP-complete. This implies (2.6) that the β-solvability problem for NP-regular SL-systems is NP-complete. Since NP-regular SL-systems are the core of our compiler we have (2.7, 3.0.0) that the existence of executable code satisfying a set of program specifications in our equational programming language is an NP-complete problem. This is the price that we have to pay for allowing an unrestrained presence of self-application.

B.0. Theorem. *Let $\mathbb{T}$ be a sms theory (e.g. β or $\beta\eta$). The $\mathbb{T}$-X-separability problem for λ-free sets is NP-complete.*

Proof. Let $\mathfrak{F}, \mathscr{B} \subset_{\mathrm{f}} \Lambda$, $X, Z \subset_{\mathrm{f}} \mathrm{Var}$.

0. Notation. (0) We say that $(\mathfrak{F}, X)$ is $\mathbb{T}$-solvable iff $\mathfrak{F}$ is $\mathbb{T}$-X-separable.

(1) $(\mathfrak{F}, X) \cup (\mathscr{B}, Z) = (\mathfrak{F} \cup \mathscr{B}, X \cup Z)$.

(2) $M \in (\mathfrak{F}, X)$ iff $M \in \mathfrak{F}$.

(3) $\mathrm{approx}((\mathfrak{F}, X)) = \{(\mathfrak{G}, X) \mid \mathfrak{G}$ is obtained from $\mathfrak{F}$ replacing each $M \in \mathfrak{F}$ with M' s.t.: $M' \sqsubseteq M\}$.

(4) $(\mathfrak{F}, X)$ is said to be GPFR iff $\mathscr{S} = (\{M = y \mid M \in \mathfrak{F}\}, X)$ is GPFR (see 3.2.3) (where y is a fresh variable). $\diamond_0$

1. Remark. If $\mathrm{FV}(\mathfrak{F}) \subseteq X$ from A.5.8 we have: $\mathfrak{F}$ is $\mathbb{T}$-X-separable iff $\exists \mathfrak{G} \in \mathrm{approx}(\mathfrak{F})$ [$\mathfrak{G}$ is GPFR). $\diamond_1$

We will codify the satisfiability problem for propositional formulas into an X-separability problem. Let PropForm (PropVar) be the set of propositional formulas (variables).

2. Definition. (0) $\mathrm{conc}(x\mathbf{U}_i^2\Omega, Q) = zQ\Omega$, $i = 1, 2$;
$\mathrm{conc}(x\Omega\mathbf{U}_i^2, Q) = z\Omega Q$, $i = 1, 2$.

(1) $\mathrm{switch}(x\mathbf{U}_i^2\Omega) = x\mathbf{U}_{3-i}^2\Omega$, $i = 1, 2$; $\mathrm{switch}(x\Omega\mathbf{U}_i^2) = x\Omega\mathbf{U}_{3-i}^2$, $i = 1, 2$. $\diamond_2$

We associate λ-terms to (propositional) formulas.

3. Definition. Let $A \in \mathrm{PropForm}$ and $\alpha \in \mathrm{Seq}$. We define:

(0) $\mathrm{true}(\alpha, A) =$

case

$A \equiv x_i \in \mathrm{PropVar}$	**then**	$(x_i\mathbf{U}_i^2\Omega)$;
$A \equiv \neg B$	**then**	$(\mathrm{not}_\alpha\mathbf{U}_1^2\Omega)$;
$A = B_1 \vee B_2$	**then**	$(\mathrm{or}_\alpha\mathbf{U}_1^2\Omega)$;
$A = B_1 \wedge B_2$	**then**	$(\mathrm{and}_\alpha\mathbf{U}_1^2\Omega)$;

end.

(1) $\mathrm{false}(\alpha, A) =$

case

$A \equiv x_i \in \mathrm{PropVar}$	**then**	$(x_i\Omega\mathbf{U}_1^2)$;
$A \equiv \neg B$	**then**	$(\mathrm{not}_\alpha\Omega\mathbf{U}_1^2)$;
$A = B_1 \vee B_2$	**then**	$(\mathrm{or}_\alpha\Omega\mathbf{U}_1^2)$;
$A = B_1 \wedge B_2$	**then**	$(\mathrm{and}_\alpha\Omega\mathbf{U}_1^2)$;

end. $\diamond_3$

We associate an X-separability problem to (propositional) formulas according to the following idea. The $\{x\}$-separability problem

$$(\mathfrak{G}, \{x\}) = (\{x(x\mathbf{U}_1^2\Omega)(x\Omega\mathbf{U}_1^2), x(x\mathbf{U}_2^2\Omega)(x\Omega\mathbf{U}_2^2)\}, \{x\})$$

has only two GPFR approximations:

$$(\mathfrak{F}_1, \{x\}) = (\{x(x\mathbf{U}_1^2\Omega)\Omega, x(x\mathbf{U}_2^2\Omega)\Omega\}, \{x\})$$

$$(\mathfrak{F}_2, \{x\}) = (\{x\Omega(x\Omega\mathbf{U}_1^2), x\Omega(x\Omega\mathbf{U}_2^2))\}, \{x\}).$$

We will codify the truth value True(T) with $(\mathfrak{F}_1, \{x\})$ and False(F) with $(\mathfrak{F}_2, \{x\})$.

4. Definition. Let $A \in \text{PropForm}$ and $\alpha \in \text{Seq}$. We define:

$\text{Syst}(\alpha, A) =$

case

$A \equiv x_i \in \text{PropVar}$ **then** $(\{x_i(x_i \mathbf{U}_1^2 \Omega)(x_i \Omega \mathbf{U}_1^2),\ x_i(x_i \mathbf{U}_2^2 \Omega), (x_i \Omega \mathbf{U}_2^2)\}, \{x_i\})$;

$A \equiv \neg B$ **then** $(\text{not}_\alpha(\text{not}_\alpha \mathbf{U}_1^2 \Omega)(\text{not}_\alpha \Omega \mathbf{U}_1^2),$
$\text{not}_\alpha(\text{not}_\alpha \mathbf{U}_2^2 \Omega)(\text{not}_\alpha \Omega \mathbf{U}_2^2),$
$\text{not}_\alpha \text{false}(\alpha^* \langle 0 \rangle, B)\, \text{true}(\alpha^* \langle 0 \rangle, B)\}, \{\text{not}_\alpha\}) \cup$
$\text{Syst}(\alpha^* \langle 0 \rangle, B)$;

$A = B_1 \vee B_2$ **then** $(\{\text{or}_\alpha(\lambda t.\, t\ \text{true}(\alpha^* \langle 0 \rangle, B_1)\, \text{true}(\alpha^* \langle 1 \rangle, B_2))$
$\text{conc}(\text{false}(\alpha^* \langle 0 \rangle, B_1), \text{false}(\alpha^* \langle 1 \rangle, B_2)),$
$\text{or}_\alpha\ (\text{or}_\alpha(\lambda t.\, t \mathbf{U}_1^2 \mathbf{U}_1^2)\Omega)$
$(\text{or}_\alpha \Omega\, \text{conc}(\text{false}(\alpha^* \langle 0 \rangle, B_1), \text{false}(\alpha^* \langle 1 \rangle, B_2))),$
$\text{or}_\alpha\ (\text{or}_\alpha(\lambda t.\, t \mathbf{U}_2^2 \mathbf{U}_2^2)\Omega)$
$(\text{or}_\alpha \Omega\, \text{conc}(\text{switch}(\text{false}(\alpha^* \langle 0 \rangle, B_1)),$
$\text{switch}(\text{false}(\alpha^* \langle 1 \rangle, B_2))))\}, \{\text{or}_\alpha\}) \cup$
$\text{Syst}(\alpha^* \langle 0 \rangle, B_1) \cup (\text{Syst}(\alpha^* \langle 1 \rangle, B_2)$;

$A = B_1 \wedge B_2$ **then** $(\{\text{and}_\alpha \text{conc}(\text{true}(\alpha^* \langle 0 \rangle, B_1), \text{true}(\alpha^* \langle 1 \rangle, B_2))$
$(\lambda t.\, t\ \text{false}(\alpha^* \langle 0 \rangle, B_1)\, \text{false}(\alpha^* \langle 1 \rangle, B_2)),$
$\text{and}_\alpha(\text{and}_\alpha \text{conc}(\text{true}(\alpha^* \langle 0 \rangle, B_1), \text{true}(\alpha^* \langle 1 \rangle, B_2))\Omega)$
$(\text{and}_\alpha \Omega(\lambda t.\, t \mathbf{U}_1^2 \mathbf{U}_1^2)),$
$\text{and}_\alpha(\text{and}_\alpha \text{conc}(\text{switch}(\text{true}(\alpha^* \langle 0 \rangle, B_1)),$
$\text{switch}(\text{true}(\alpha^* \langle 1 \rangle, B_2)))\Omega)$
$(\text{and}_\alpha \Omega(\lambda t.\, t \mathbf{U}_2^2 \mathbf{U}_2^2))\}, \{\text{and}_\alpha\}) \cup \text{Syst}(\alpha^* \langle 0 \rangle, B_1)$
$\cup (\text{Syst}(\alpha^* \langle 1 \rangle, B_2)$;

end. $\diamond_4$

5. Example. Let $A \equiv \neg x$. We have:

$$\text{Syst}(\langle\ \rangle, \neg x) = (\{\text{not}_{\langle\,\rangle}(\text{not}_{\langle\,\rangle} \mathbf{U}_1^2 \Omega), (\text{not}_{\langle\,\rangle} \Omega \mathbf{U}_1^2) \text{not}_{\langle\,\rangle}(\text{not}_{\langle\,\rangle} \mathbf{U}_2^2 \Omega)(\text{not}_{\langle\,\rangle} \Omega \mathbf{U}_2^2),$$
$$\text{not}_{\langle\,\rangle}(x \Omega \mathbf{U}_1^2)(x \mathbf{U}_1^2 \Omega),$$
$$x(x \mathbf{U}_1^2 \Omega)(x \Omega \mathbf{U}_1^2), x(x \mathbf{U}_2^2 \Omega)(x \Omega \mathbf{U}_2^2)\}, \{x, \text{not}_{\langle\,\rangle}\}).$$

$\text{Syst}(\langle\ \rangle, \neg x)$ has only two GPFR approximations:

$$(\mathfrak{A}_1, \{x, \text{not}_{\langle\,\rangle}\}) = (\{\text{not}_{\langle\,\rangle}(\text{not}_{\langle\,\rangle} \mathbf{U}_1^2 \Omega)\Omega, \text{not}_{\langle\,\rangle}(\text{not}_{\langle\,\rangle} \mathbf{U}_2^2 \Omega)\Omega,$$
$$\text{not}_{\langle\,\rangle}(x \Omega \mathbf{U}_1^2)\Omega, x\Omega(x \Omega \mathbf{U}_1^2) x \Omega(x \Omega \mathbf{U}_2^2)\}, \{x, \text{not}_{\langle\,\rangle}\}),$$

$$(\mathfrak{A}_2, \{x, \text{not}_{\langle\,\rangle}\}) = (\{\text{not}_{\langle\,\rangle} \Omega(\text{not}_{\langle\,\rangle} \Omega \mathbf{U}_1^2), \text{not}_{\langle\,\rangle} \Omega(\text{not}_{\langle\,\rangle} \Omega \mathbf{U}_2^2),$$
$$\text{not}_{\langle\,\rangle} \Omega(x \mathbf{U}_1^2 \Omega), x(x \mathbf{U}_1^2 \Omega)\Omega, x(x \mathbf{U}_2^2 \Omega)\Omega\}, \{x, \text{not}_{\langle\,\rangle}\}).$$

$(\mathfrak{A}_1, \{x, \text{not}_{\langle\,\rangle}\})$ corresponds to the assignment $x := \text{T}$, whereas $(\mathfrak{A}_2, \{x, \text{not}_{\langle\,\rangle}\})$ corresponds to the assignment $x := \text{F}$. $\diamond_5$

The following X-separability problem codifies the satisfability problem for propositional formulas.

6. Definition. Let $A \in \mathrm{PropForm}$ and $\alpha \in \mathrm{Seq}$. We define:
$\mathrm{Syst\,T}(\alpha, A) =$
case

$A \equiv x_i \in \mathrm{PropVar}$ **then** $(\{x_i(x_i \mathbf{U}_1^2 \Omega)\,\Omega,\ x_i(x_i \mathbf{U}_2^2 \Omega)\,\Omega\}, \{x_i\})$;

$A \equiv \neg B$ **then** $(\{\mathrm{not}_\alpha(\mathrm{not}_\alpha \mathbf{U}_1^2 \Omega), \Omega,$
$\quad \mathrm{not}_\alpha(\mathrm{not}_\alpha \mathbf{U}_2^2 \Omega), \Omega,$
$\quad \mathrm{not}_\alpha \mathrm{false}(\alpha^* \langle 0\rangle, B)\Omega\}, \{\mathrm{not}_\alpha\}) \cup \mathrm{Syst}(\alpha^* \langle 0\rangle, B)$;

$A = B_1 \vee B_2$ **then** $(\{\mathrm{or}_\alpha(\lambda t.\ t\ \mathrm{true}(\alpha^* \langle 0\rangle, B_1)\ \mathrm{true}(\alpha^* \langle 1\rangle, B_2))\Omega,$
$\quad \mathrm{or}_\alpha\ (\mathrm{or}_\alpha(\lambda t.\ t \mathbf{U}_1^2 \mathbf{U}_1^2)\Omega)\ \Omega,$
$\quad \mathrm{or}_\alpha\ (\mathrm{or}_\alpha(\lambda t.\ t \mathbf{U}_2^2 \mathbf{U}_2^2)\Omega)\ \Omega\}, \{\mathrm{or}_\alpha\}) \cup$
$\mathrm{Syst}(\alpha^* \langle 0\rangle, B_1) \cup \mathrm{Syst}(\alpha^* \langle 1\rangle, B_2)$;

$A = B_1 \wedge B_2$ **then** $(\{\mathrm{and}_\alpha \mathrm{conc}(\mathrm{true}(\alpha^* \langle 0\rangle, B_1), \mathrm{true}(\alpha^* \langle 1\rangle, B_2))\Omega,$
$\quad \mathrm{and}_\alpha(\mathrm{and}_\alpha \mathrm{conc}(\mathrm{true}(\alpha^* \langle 0\rangle, B_1), \mathrm{true}(\alpha^* \langle 1\rangle, B_2))\Omega)\Omega,$
$\quad \mathrm{and}_\alpha(\mathrm{and}_\alpha \mathrm{conc}(\mathrm{switch}(\mathrm{true}(\alpha^* \langle 0\rangle, B_1)),$
$\qquad \mathrm{switch}(\mathrm{true}(\alpha^* \langle 1\rangle, B_2)))\Omega)\Omega\}, \{\mathrm{and}_\alpha\})$
$\cup\ \mathrm{Syst}(\alpha^* \langle 0\rangle, B_1) \cup \mathrm{Syst}(\alpha^* \langle 1\rangle, B_2)$;

end. $\diamond_6$

7. Example. $\mathrm{SystT}(\langle\ \rangle, \neg x) = (\mathfrak{A}_1, \{x, \mathrm{not}_{\langle\ \rangle}\})$, where $(\mathfrak{A}_1, \{x, \mathrm{not}_{\langle\ \rangle}\})$ is as in example 5. $\diamond_7$

The following X-separability problem codifies the truth-value of a propositional formula under a given assignment for the propositional variables.

8. Definition. Let $A \in \mathrm{PropForm}$, $\alpha \in \mathrm{Seq}$ and $\varphi : \mathrm{PropVar} \to \{\mathrm{T}, \mathrm{F}\}$. We define:
$\mathrm{Syst}(\alpha, \varphi, A) =$
case

$A \equiv x_i \in \mathrm{PropVar}$
then if $\varphi(x_i) = T$
then $(\{x_i(x_i \mathbf{U}_1^2 \Omega)\,\Omega,\ x_i(x_i \mathbf{U}_2^2 \Omega)\,\Omega\}, \{x_i\})$
else $(\{x_i \Omega(x_i \Omega \mathbf{U}_1^2),\ x_i \Omega(x_i \Omega \mathbf{U}_2^2)\}, \{x_i\})$;

$A \equiv \neg B$
then if $\varphi(B) = \mathrm{T}$
then $(\{\mathrm{not}_\alpha \Omega(\mathrm{not}_\alpha \Omega \mathbf{U}_1^2), \mathrm{not}_\alpha \Omega(\mathrm{not}_\alpha \Omega \mathbf{U}_2^2), \mathrm{not}_\alpha \Omega\, \mathrm{true}(\alpha^* \langle 0\rangle, B)\},$
$\quad \{\mathrm{not}_\alpha\}) \cup \mathrm{Syst}(\alpha^* \langle 0\rangle, \varphi, B)$;
else $(\{\mathrm{not}_\alpha(\mathrm{not}_\alpha \mathbf{U}_1^2 \Omega)\Omega, \mathrm{not}_\alpha(\mathrm{not}_\alpha \mathbf{U}_2^2 \Omega)\Omega, \mathrm{not}_\alpha \mathrm{false}(\alpha^* \langle 0\rangle, B)\Omega\},$
$\quad \{\mathrm{not}_\alpha\}) \cup \mathrm{Syst}(\alpha^* \langle 0\rangle, \varphi, B)$;

$A = B_1 \vee B_2$
then case
$\varphi(B_1) = \varphi(B_2) = \mathrm{T}$

then $(\{\mathrm{or}_\alpha(\lambda t.\ t\ \mathrm{true}(\alpha^* \langle 0\rangle, B_1)\ \mathrm{true}(\alpha^* \langle 1\rangle, B_2))\Omega,$
$\mathrm{or}_\alpha\ (\mathrm{or}_\alpha(\lambda t.\ t\mathbf{U}_1^2\mathbf{U}_1^2)\Omega)\ \Omega,$
$\mathrm{or}_\alpha\ (\mathrm{or}_\alpha(\lambda t.\ t\mathbf{U}_2^2\mathbf{U}_2^2)\Omega)\ \Omega\}, \{\mathrm{or}_\alpha\}) \cup$
$\mathrm{Syst}(\alpha^* \langle 0\rangle, \varphi, B_1) \cup \mathrm{Syst}(\alpha^* \langle 1\rangle, \varphi, B_2);$

$\varphi(B_1) = \mathrm{T}$ and $\varphi(B_2) = \mathrm{F}$
then $(\{\mathrm{or}_\alpha(\lambda t.\ t\ \mathrm{true}(\alpha^* \langle 0\rangle, B_1)\Omega)\Omega,$
$\mathrm{or}_\alpha\ (\mathrm{or}_\alpha(\lambda t.\ t\mathbf{U}_1^2\mathbf{U}_1^2)\Omega)\ \Omega,$
$\mathrm{or}_\alpha\ (\mathrm{or}_\alpha(\lambda t.\ t\mathbf{U}_2^2\mathbf{U}_2^2)\Omega)\ \Omega\}, \{\mathrm{or}_\alpha\}) \cup$
$\mathrm{Syst}(\alpha^* \langle 0\rangle, \varphi, B_1) \cup \mathrm{Syst}(\alpha^* \langle 1\rangle, \varphi, B_2);$

$\varphi(B_1) = \mathrm{F}$ and $\varphi(B_2) = \mathrm{T}$
then $(\{\mathrm{or}_\alpha(\lambda t.\ t\Omega\ \mathrm{true}(\alpha^* \langle 1\rangle, B_2))\Omega,$
$\mathrm{or}_\alpha\ (\mathrm{or}_\alpha(\lambda t.\ t\mathbf{U}_1^2\mathbf{U}_1^2)\Omega)\ \Omega,$
$\mathrm{or}_\alpha\ (\mathrm{or}_\alpha(\lambda t.\ t\mathbf{U}_2^2\mathbf{U}_2^2)\Omega)\ \Omega\}, \{\mathrm{or}_\alpha\}) \cup$
$\mathrm{Syst}(\alpha^* \langle 0\rangle, \varphi, B_1) \cup \mathrm{Syst}(\alpha^* \langle 1\rangle, \varphi, B_2);$

$\varphi(B_1) = \varphi(B_2) = \mathrm{F}$
then $(\{\mathrm{or}_\alpha(\lambda t.\ t\Omega\Omega)\mathrm{conc}(\mathrm{false}(\alpha^* \langle 0\rangle, B_1), \mathrm{false}(\alpha^* \langle 1\rangle, B_2)),$
$\mathrm{or}_\alpha\Omega(\mathrm{or}_\alpha\Omega\mathrm{conc}(\mathrm{false}(\alpha^* \langle 0\rangle, B_1), \mathrm{false}(\alpha^* \langle 1\rangle, B_2))),$
$\mathrm{or}_\alpha\Omega(\mathrm{or}_\alpha\Omega\mathrm{conc}(\mathrm{switch}(\mathrm{false}(\alpha^* \langle 0\rangle, B_1)),$
$\mathrm{switch}(\mathrm{false}(\alpha^* \langle 1\rangle, B_2))))\}, \{\mathrm{or}_\alpha\}) \cup$
$\mathrm{Syst}(\alpha^* \langle 0\rangle, \varphi, B_1) \cup (\mathrm{Syst}(\alpha^* \langle 1\rangle, \varphi, B_2);$

end;

$A = B_1 \wedge B_2$
then case
$\varphi(B_1) = \varphi(B_2) = \mathrm{T}$
then $(\{\mathrm{and}_\alpha \mathrm{conc}(\mathrm{true}(\alpha^* \langle 0\rangle, B_1), \mathrm{true}(\alpha^* \langle 1\rangle, B_2))\Omega,$
$\mathrm{and}_\alpha(\mathrm{and}_\alpha \mathrm{conc}(\mathrm{true}(\alpha^* \langle 0\rangle, B_1), \mathrm{true}(\alpha^* \langle 1\rangle, B_2))\Omega)\Omega,$
$\mathrm{and}_\alpha(\mathrm{and}_\alpha \mathrm{conc}(\mathrm{switch}(\mathrm{true}(\alpha^* \langle 0\rangle, B_1)),$
$\mathrm{switch}(\mathrm{true}(\alpha^* \langle 1\rangle, B_2)))\Omega)\Omega\}, \{\mathrm{and}_\alpha\})$
$\cup\ \mathrm{Syst}(\alpha^* \langle 0\rangle, \varphi, B_1) \cup \mathrm{Syst}(\alpha^* \langle 1\rangle, \varphi, B_2);$

$\varphi(B_1) = \mathrm{T}$ and $\varphi(B_2) = \mathrm{F}$
then $(\{\mathrm{and}_\alpha\Omega(\lambda t.\ t\Omega\ \mathrm{false}(\alpha^* \langle 1\rangle, B_2)),$
$\mathrm{and}_\alpha\Omega\ (\mathrm{and}_\alpha\Omega(\lambda t.\ t\mathbf{U}_1^2\mathbf{U}_1^2)),$
$\mathrm{and}_\alpha\Omega\ (\mathrm{and}_\alpha\Omega(\lambda t.\ t\mathbf{U}_2^2\mathbf{U}_2^2))\}, \{\mathrm{and}_\alpha\}) \cup$
$\mathrm{Syst}(\alpha^* \langle 0\rangle, \varphi, B_1) \cup \mathrm{Syst}(\alpha^* \langle 1\rangle, \varphi, B_2);$

$\varphi(B_1) = \mathrm{F}$ and $\varphi(B_2) = \mathrm{T}$
then $(\{\mathrm{and}_\alpha\Omega(\lambda t.\ t\ \mathrm{false}(\alpha^* \langle 0\rangle, B_1)\Omega),$
$\mathrm{and}_\alpha\Omega\ (\mathrm{and}_\alpha\Omega(\lambda t.\ t\mathbf{U}_1^2\mathbf{U}_1^2)),$
$\mathrm{and}_\alpha\ (\mathrm{and}_\alpha\Omega(\lambda t.\ t\mathbf{U}_2^2\mathbf{U}_2^2))\}, \{\mathrm{and}_\alpha\}) \cup$
$\mathrm{Syst}(\alpha^* \langle 0\rangle, \varphi, B_1) \cup \mathrm{Syst}(\alpha^* \langle 1\rangle, \varphi, B_2);$

$\varphi(B_1) = \varphi(B_2) = \mathrm{F}$
then $(\{\mathrm{and}_\alpha\Omega(\lambda t.\ t\ \mathrm{false}(\alpha^* \langle 0\rangle, B_1)\ \mathrm{false}(\alpha^* \langle 1\rangle, B_2)),$
$\mathrm{and}_\alpha\Omega\ (\mathrm{and}_\alpha\Omega(\lambda t.\ t\mathbf{U}_1^2\mathbf{U}_1^2)),$

$\text{and}_\alpha \Omega\, (\text{and}_\alpha \Omega(\lambda t.\, t\mathbf{U}_2^2\mathbf{U}_2^2))\}, \{\text{and}_\alpha\}) \cup$
$\text{Syst}(\alpha^* \langle 0\rangle, \varphi, B_1) \cup \text{Syst}(\alpha^* \langle 1\rangle, \varphi, B_2)$;

end;

end. $\diamond_8$

9. Lemma. Let $A \in \text{PropForm}$ *and* $\alpha \in \text{Seq}$. *If* $\mathcal{S} \in \text{approx}(\text{Syst}(\alpha, A))$ *and* $\mathcal{S}$ *is* GPFR *then* $\exists\varphi : \text{PropVar} \to \{\text{T}, \text{F}\}$ s.t. $\mathcal{S} = \text{Syst}(\alpha, \varphi, A)$.

Proof. By induction on A.

Case 0: $A \equiv x_i \in \text{PropVar}$. Then $\text{Syst}(\alpha, A) = (\{x_i(x_i\mathbf{U}_1^2\Omega)(x_i\Omega\mathbf{U}_1^2),\ x_i(x_i\mathbf{U}_2^2\Omega)(x_i\Omega\mathbf{U}_2^2)\}, \{x_i\})$.

Case 0.0: $\mathcal{S} = (\{x_i(x_i\mathbf{U}_1^2\Omega)\Omega,\ x_i(x_i\mathbf{U}_2^2\Omega)\Omega\}, \{x_i\})$. Then choose $\varphi(x_i) = \text{T}$.

Case 0.1: $\mathcal{S} = (\{x_i\Omega(x_i\Omega\mathbf{U}_1^2),\ x_i\Omega(x_i\Omega\mathbf{U}_2^2)\}, \{x_i\})$. Then choose $\varphi(x_i) = \text{F}$.

Case 1: $A \equiv \neg B$. Then $\text{Syst}(\alpha, A) = (\{\text{not}_\alpha(\text{not}_\alpha\mathbf{U}_1^2\Omega)(\text{not}_\alpha\Omega\mathbf{U}_1^2),\ \text{not}_\alpha(\text{not}_\alpha\mathbf{U}_2^2\Omega)(\text{not}_\alpha\Omega\mathbf{U}_2^2),\ \text{not}_\alpha\text{false}(\alpha^*\langle 0\rangle, B)\ \text{true}(\alpha^*\langle 0\rangle, B)\}, \{\text{not}_\alpha\}) \cup \text{Syst}(\alpha^*\langle 0\rangle, B)$.

Case 1.0: $\mathcal{S} = (\{\text{not}_\alpha(\text{not}_\alpha\mathbf{U}_1^2\Omega)\Omega,\ \text{not}_\alpha(\text{not}_\alpha\mathbf{U}_2^2\Omega)\Omega,\ \text{not}_\alpha\text{false}(\alpha^*\langle 0\rangle, B)\Omega\}, \{\text{not}_\alpha\}) \cup \mathcal{S}'$, where $\mathcal{S}' \in \text{approx}(\text{Syst}(\alpha^*\langle 0\rangle, B))$ and $\mathcal{S}'$ is GPFR.

By induction hypothesis there exists φ s.t. $\mathcal{S}' = \text{Syst}(\alpha^*\langle 0\rangle, \varphi, B)$. Since $\mathcal{S}$ is GPFR and by the definition of $\text{Syst}(\alpha^*\langle 0\rangle, \varphi, B)$ we have $\varphi(B) = \text{F}$. Hence $\varphi(A) = \text{T}$. Thus $\mathcal{S} = \text{Syst}(\alpha, \varphi, A)$.

Case 1.1: $\mathcal{S} = (\{\text{not}_\alpha\Omega(\text{not}_\alpha\Omega\mathbf{U}_1^2),\ \text{not}_\alpha\Omega(\text{not}_\alpha\Omega\mathbf{U}_2^2),\ \text{not}_\alpha\Omega\ \text{true}(\alpha^*\langle 0\rangle, B)\}, \{\text{not}_\alpha\}) \cup \mathcal{S}'$, where $\mathcal{S}' \in \text{approx}(\text{Syst}(\alpha^*\langle 0\rangle, B))$ and $\mathcal{S}'$ is GPFR.

Analogous to case 1.0.

Case 2: $A = B_1 \vee B_2$. Analogous to case 1.

Case 3: $A = B_1 \wedge B_2$. Analogous to case 1. $\diamond_9$

10. Lemma. Let $A \in \text{PropForm}$ *and* $\alpha \in \text{Seq}$ *and* $\varphi : \text{PropVar} \to \{\text{T}, \text{F}\}$. *Then* $\text{Syst}(\alpha, \varphi, A)$ *is* GPFR.

Proof. By induction on A. $\diamond_{10}$

11. Lemma. Let $A \in \text{PropForm}$. *Then*:
A is satisfiable iff $\exists\mathcal{S} \in \text{approx}(\text{SystT}(\langle\ \rangle, A))$ [$\mathcal{S}$ *is* GPFR].

Proof. ($\Rightarrow$) Let $\varphi : \text{PropVar} \to \{\text{T}, \text{F}\}$ s.t. $\varphi(A) = \text{T}$. Then, by Lemma 10, $\text{Syst}(\langle\ \rangle, \varphi, A)$ is GPFR and, since $\varphi(A) = \text{T}$, $\text{Syst}(\langle\ \rangle, \varphi, A) \in \text{approx}(\text{SystT}(\langle\ \rangle, A))$.
($\Leftarrow$). Let $\mathcal{S} \in \text{approx}(\text{SystT}(\langle\ \rangle, A))$ s.t. [$\mathcal{S}$ is GPFR]. Then, by Lemma 9, there exists φ: $\text{PropVar} \to \{\text{T}, \text{F}\}$ s.t. $\mathcal{S} = \text{Syst}(\langle\ \rangle, \varphi, A)$. Then, by definition 8, $\varphi(A) = \text{T}$. $\diamond_{11}$

12. Lemma. The $\mathbb{T}$*-X-separability problem for λ-free sets is* NP*-hard.*

Proof. From Lemma 11 and Remark 1. $\diamond_{12}$

Since the problem of deciding if an SL-system is GPFR is in NP the thesis follows from Lemma 12. □

Appendix C. Proof of 3.0

In C.2 we prove 3.0. To this end we need to show that results analogous to [8, C.7, C.9] hold when unrestrained self-application is allowed. This is done in C.0, C.1.

Using C.0, C.1 we can take an argument of a function symbol from the LHS to the RHS.

C.0. Lemma. *Let* $\mathcal{S} = (\Gamma_1 \cup \Gamma_2, \{\vec{x}\})$ *be a system and* $\vec{y} \in (\mathrm{Var} - \{\vec{x}\})$ *s.t.:*

H0. *Each equation in* Γ_1 *has form* $x\vec{M} = y\vec{M}$, *where* $x \in \{\vec{x}\}$ *and* $y \in \{\vec{y}\}$.

H1. $\mathrm{head}(\mathrm{right}(\Gamma_1)) \subseteq \{\vec{y}\}$ *and* $\vec{y} \equiv y_1, \ldots, y_k$ *is assignable in* $\mathcal{S}$ *(see* [8, 6.0]).

H2. $\forall x\vec{M} = y\vec{M},\ x\vec{N} = y'\vec{N} \in \Gamma_1\ [y \equiv y' \Rightarrow \deg(y\vec{M}) = \deg(y'(\vec{N})]$.

H3. *Each equation in* Γ_2 *has form* $x\vec{M} = z$, *where* $x \in \{\vec{x}\}$ *and* $z \notin \{\vec{x}, \vec{y}\}$.

H4. $\mathcal{S}_\Omega$ *is* NP-*regular* ($\mathcal{S}_\Omega$ *is defined in* [8, C.0]).

Then: $\mathcal{S}$ *is* β*-solvable iff* $\mathrm{NP\_OK\_NEC}(\mathcal{S}_\Omega)$

Proof. ($\Rightarrow$) By [8, C.1] and 2.0.2.

($\Leftarrow$) As in [8, C.7], but using 2.0, 2.1 instead of, respectively, [8, 5.2, 5.3]. □

C.1. Theorem. *Let* $\mathcal{S} = (\Gamma_1 \cup \Gamma_2, \{\vec{x}\})$ *be a system and* $\vec{y} \in (\mathrm{Var} - \{\vec{x}\})$ *s.t.:*

H0. $\vec{y} \equiv y_1, \ldots, y_k$ *is assignable in* $\mathcal{S}$ (see [8, 6.0]);

H1. *Each equation in* Γ_1 *has form* $x\vec{M} = y(\vec{x}:\vec{y})\vec{M}\,\vec{z}$, (see [8, A.0.0]) *where:* $x \in \{\vec{x}\}$, $y \in \{\vec{y}\}$, $\vec{z} \notin \{\vec{x}, \vec{y}\}$ *and the variables in* $\mathrm{head}(\mathrm{right}(\Gamma_1))$ *are pairwise distinct;*

H2. *Each equation in* Γ_2 *has form* $x\vec{M} = z$, *where* $x \in \{\vec{x}\}$ *and* $z \notin \{\vec{x}, \vec{y}\}$;

H3. $\mathcal{S}^{\#} = ((\{x\vec{M} = y \mid x\vec{M} = y(\vec{x}:\vec{y})\vec{M}\vec{z} \in \Gamma_1\} \cup \{u_{yz}\vec{M} = z \mid x\vec{M} = y(\vec{x}:\vec{y})\vec{M}\vec{z} \in \Gamma_1$ *and* $z \in \{\vec{z}\}$ *and* $u_{yz} \in \{\vec{u}\}\} \cup \Gamma_2), \{\vec{x}, \vec{u}\})$ *is regular, where* $\vec{u}$ *is a sequence of fresh variables s.t.:* $\{\vec{u}\} = \{u_{yz} \mid x\vec{M} = y(x_1\vec{y})\ldots(x_n\vec{y})\vec{z} \in \Gamma_1$ *and* $z \in \{\vec{z}\}\}$.

0. $\mathcal{S}$ *is* β*-solvable iff* $\mathrm{NP\_OK\_NEC}(\mathcal{S}^{\#}) = \mathit{true}$.

1. *If* $\mathcal{S}$ *is* β*-solvable and* $\mathrm{Card}(\Gamma_1 \cup \Gamma_2) > 1$ *then* $\mathcal{S}$ *has a* β*-solution having normal form.*

Proof. 0. ($\Rightarrow$) As in [8, C.9] but using 2.0.2 instead of [8, 5.2.2].

($\Leftarrow$) As in [8, C.9] but using 2.0, C.0 instead of, respectively, [8, 5.2, C.7].

1. As in [8, C.7]. □

C.2. Proof of 3.0. 0. ($\Rightarrow$) As in C.1. ($\Leftarrow$) By C.1 and [8, C.1].

1. From the constructions in C.0, C.1 and in Section A.4.

2. As in C.1. □

References

[1] H.P. Barendregt, *The Lambda Calculus* (North-Holland, Amsterdam, 1984).

[2] C. Böhm and A. Piperno, Characterizing X-separability and one-side invertibility in $\lambda\beta\Omega$-calculus, *Proc. LICS 88*, Edinburgh, July 4–8, 1988 (IEEE, Computer Soc. Press, Silver Spring, MD, 1988)

[3] C. Böhm, A. Piperno and E. Tronci, Solving equations in λ-calculus, *Proc. Logic Colloquium 88*, Padova (North-Holland, Amsterdam, 1989).

[4] C. Böhm and E. Tronci, X-Separability and left-invertibility in lambda-calculus, *Proc. LICS 87*, Ithaca NY, June 22–25, 1987, (IEEE, Computer Soc. Press, Silver Spring, MD, 1987) 320–328.

[5] C. Böhm and E. Tronci, About systems of equations, X-separability and left-invertibility in the λ-calculus, *Inform. Comput.* **90** (1991) 1–32.

[6] A. Piperno and E. Tronci, Regular systems of equations in the λ-calculus, *Int. J. Found. Comput. Sci.* **1** (1990) 325–339.

[7] E. Tronci, Equational programming in λ-calculus via SL-systems, Ph.D. Thesis, Carnegie Mellon University, 1991 (advisor: R. Statman).

[8] E. Tronci, Equational Programming in λ-calculus via SL-systems. Part 1, *Theoret. Comput. Sci.* **160** (this Vol.) (1996) 145–184.