JFP **16** (2): 137–155, 2006. © 2003 Cambridge University Press
doi:10.1017/S0956796803004726 First published online 16 September 2003 Printed in the United Kingdom

Remarks on the equational theory of non-normalizing pure type systems

GILLES BARTHE

INRIA Sophia-Antipolis, France
(*e-mail:* `Gilles.Barthe@inria.fr`)

THIERRY COQUAND

Chalmers University, Göteborg, Sweden
(*e-mail:* `coquand@cs.chalmers.se`)

Abstract

Pure Type Systems (PTS) come in two flavours: domain-free systems with untyped λ-abstractions (i.e. of the form $\lambda x\,.\,M$); and domain-free systems with typed λ-abstractions (i.e. of the form $\lambda x : A\,.\,M$). Both flavours of systems are related by an erasure function $|.|$ that removes types from λ-abstractions. *Preservation of Equational Theory*, which states the equational theories of both systems coincide through the erasure function, is a property of functional and normalizing PTSs. In this paper we establish that Preservation of Equational Theory fails for some non-normalizing PTSs, including the PTS with $* : *$. The gist of our argument is to exhibit a typable expression Y_H whose erasure $|Y|$ is a fixpoint combinator, but which is not a fixpoint combinator itself.

1 Introduction

The simply typed λ-calculus comes in two flavours (see figure 1):

- the simply typed λ-calculus *à la* Church, which features an *explicitely typed* λ-abstraction, and in which one writes $\lambda x : A\,.\,M$ for the function of domain A which sends x to M;
- the simply typed λ-calculus *à la* Curry, which features an *implicitely typed* λ-abstraction, and in which one writes $\lambda x\,.\,M$ for the function of domain A which sends x to M.

The two formulations are equivalent in the sense that typing and conversion are preserved by the obvious erasure function $|.|$ which maps λ-terms *à la* Church to λ-terms *à la* Curry. More precisely, the following two properties hold:

Preservation of typing – if $\Gamma \vdash M : A$ then $\Gamma \Vdash |M| : A$. Conversely, if $\Gamma \Vdash M : A$ then $\Gamma \vdash M' : A$ for some $M' \in \Lambda$ such that $|M'| = M$.

Preservation of equational theory – if $\Gamma \vdash M : A$ and $\Gamma \vdash M' : A$ then $M =_\beta M'$ if and only if $|M| =_{\underline{\beta}} |M'|$.

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

We assume two countably infinite sets $\mathbb{B}$ and V of base types and variables, respectively.

Simply typed λ-calculus à la Church

- *Types* $\mathbb{T} = \mathbb{B} \mid \mathbb{T} \to \mathbb{T}$
- *Expressions* $\Lambda = V \mid \Lambda\,\Lambda \mid \lambda V : \mathbb{T} . \Lambda$
- *Reduction* $\to_\beta$ is the compatible closure of the contraction

$$(\lambda x : A . M)\, N \quad \to_\beta \quad M\{x := N\}$$

- *Conversion* $=_\beta$ is the reflexive-symmetric-transitive closure of $\to_\beta$
- *Typing*

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A . M : A \to B}$$

with the usual restrictions on contexts.

Simply typed λ-calculus *à la* Curry

- *Types* $\mathbb{T} = \mathbb{B} \mid \mathbb{T} \to \mathbb{T}$
- *Expressions* $\underline{\Lambda} = V \mid \underline{\Lambda}\,\underline{\Lambda} \mid \lambda V . \underline{\Lambda}$
- *Reduction* $\to_{\underline{\beta}}$ is the compatible closure of the contraction

$$(\lambda x . M)\, N \quad \to_{\underline{\beta}} \quad M\{x := N\}$$

- *Conversion* $=_{\underline{\beta}}$ is the reflexive-symmetric-transitive closure of $\to_{\underline{\beta}}$
- *Typing*

$$\frac{(x : A) \in \Gamma}{\Gamma \Vdash x : A} \qquad \frac{\Gamma \Vdash M : A \to B \quad \Gamma \Vdash N : A}{\Gamma \Vdash M\,N : B} \qquad \frac{\Gamma, x : A \Vdash M : B}{\Gamma \Vdash \lambda x . M : A \to B}$$

with the usual restrictions on contexts.

Erasure

$$\begin{array}{rcl} |.| & : & \Lambda \to \underline{\Lambda} \\ |x| & = & x \\ |M\,N| & = & |M|\,|N| \\ |\lambda x : A . M| & = & \lambda x . |M| \end{array}$$

Fig. 1. Simply typed λ-calculus.

The distinction between typed λ-calculi *à la* Church and *à la* Curry carries over to more powerful type disciplines, and it is natural to study the equivalence between the two flavours for such type disciplines. These equivalences can be analyzed conveniently in the setting of Pure Type Systems (PTS) (Barendregt, 1992; Geuvers & Nederhof, 1991), which provide a generic framework for the description of typed λ-calculi *à la* Church, and Domain-Free Pure Type Systems (DFPTS) (Barthe & Sørensen, 2000), which play a similar role for typed λ-calculi *à la* Curry – DFPTSs are not to be confused with Type Assignment Systems (Bakel *et al.*, 1997). Recall that PTSs and DFPTSs are parametrized by the notion of specification, which fixes the typing discipline under consideration. The equivalence properties for a given specification **S** are stated as follows, where $\Vdash$ and $\vdash$ denote the typing relation for DFPTSs and PTSs respectively; see section 2 for further notations.

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

Preservation of typing – if $\Gamma \vdash_S M : A$ then $\Gamma \Vdash_S |M| : A$. Conversely, if $\Gamma \Vdash_S M : A$ then $\Gamma' \vdash_S M' : A'$ for some Γ', M' and A' such that $|\Gamma'| = \Gamma$, $|M'| = M$ and $|A'| = A$.

Preservation of Equational Theory – if $\Gamma \vdash_S M : A$ and $\Gamma \vdash_S M' : A$ then $M =_\beta M'$ if and only if $|M| =_{\underline{\beta}} |M'|$.

A previous analysis (Barthe & Sørensen, 2000) shows that Preservation of Typing and Preservation of Equational Theory hold for *normalizing* and functional Pure Type Systems. This leaves open the question of equivalence for functional and non-normalizing Pure Type Systems such as $\lambda*$ and λU^-.

The main result of this note is that Preservation of Equational Theory fails for these two systems. The failure of Preservation of Equational Theory for U^- is shown by defining an expression Y_H, that is derived from the paradox of Hurkens (1995), and verifies:

1. the domain free erasure Y of Y_H is a fixpoint combinator, i.e. verifies $Y\ A\ f =_{\underline{\beta}} f\ (Y\ A\ f)$;
2. Y_H is not a fixpoint combinator, i.e. does not satisfy $Y_H\ A\ f =_\beta f\ (Y_H\ A\ f)$.

Establishing the second fact is elementary but tedious: it requires to characterize the possible reducts of $Y_H\ A\ f$. Using an idea of Geuvers & Werner (1994), we also provide a direct proof of the failure of Preservation of Equational Theory for $\lambda*$. In particular, the proof only requires to know that Y is a fixpoint combinator and does not analyze the possible reducts of $Y_H\ A\ f$.

The other results of this note are concerned with a variant of PTSs and DFPTSs, where the convertibility relation is extended to $\beta\eta$-equality and $\underline{\beta\eta}$-equality, respectively. We show that:

- confluence of $\beta\eta$-reduction on typable terms may fail for a variant of PTSs with $\beta\eta$-conversion;
- subject reduction η-reduction may fail for a variant of DFPTSs with $\underline{\beta\eta}$-conversion.

For this second result, we use the fact that strengthening may fail for DFPTSs. The latter, which is shown in Lemma 6, answers in the negative another question left open in Barthe & Sørensen (2000).

Contents This note is organized as follows. In section 2, we briefly review the notions of PTSs and DFPTSs, and the erasure function that connects the two frameworks. In section 3, we present the looping combinator Y_H associated to Hurkens' paradox in λU^-, and show that its erasure Y is a fixpoint combinator. In section 4, we establish the failure of Preservation of Equational Theory for λU^- and $\lambda*$. In section 5, we derive from our results that $\beta\eta$-reduction is not confluent on legal terms of the variant of $\lambda*$, where the conversion rule is based on $\beta\eta$-convertibility. Finally, we conclude in section 6. In the appendix we prove that a restricted form of strengthening holds for DFPTSs and that it implies soundness of DFPTSs with $\beta\eta$-conversion.

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

Preliminaries We assume the reader to be familiar with λ-calculus (Barendregt, 1984) and abstract rewriting systems (Klop, 1992). In particular, we use the following standard notation and terminology of abstract rewriting systems: we let $\to_{ij}$ denote the union of two relations $\to_i$ and $\to_j$; we let $\twoheadrightarrow_i^+$, $\twoheadrightarrow_i$, and $=_i$ denote the transitive, reflexive-transitive, and reflexive-symmetric-transitive closure of $\to_i$ respectively; we let $\downarrow_i$ denote the composition of $\twoheadrightarrow_i$ with its converse, i.e. $a \downarrow_i b$ if there exists c such that $a \twoheadrightarrow_i c$ and $b \twoheadrightarrow_i c$; as usual, we say that $\to_i$ is confluent if $=_i$ and $\downarrow_i$ coincide. Finally, we say that:

- a is in normal form w.r.t. $\to_i$, written $a \in \mathrm{NF}_i$, if there is no b s.t. $a \to_i b$;
- a is weakly normalizing w.r.t. $\to_i$, written $a \in \mathrm{WN}_i$, if $a \twoheadrightarrow_i b$ for some $b \in \mathrm{NF}_i$;
- a is strongly normalizing w.r.t. $\to_i$, written $a \in \mathrm{SN}_i$, if all reduction sequences starting from a are finite.

2 (Domain-Free) Pure Type Systems

We review the definition of DFPTSs (Barthe & Sørensen, 2000) and PTSms (Barendregt, 1992; Geuvers & Nederhof, 1991).

2.1 Specifications

Both frameworks are parameterized by the notion of specification.

Definition 1
A *specification* is a triple $\mathbf{S} = (\mathscr{S}, \mathscr{A}, \mathscr{R})$ where

1. $\mathscr{S}$ is a set of *sorts*;
2. $\mathscr{A} \subseteq \mathscr{S} \times \mathscr{S}$ is a set of *axioms*;
3. $\mathscr{R} \subseteq \mathscr{S} \times \mathscr{S} \times \mathscr{S}$ is a set of *rules*.

As usual, a rule of form (s_1, s_2, s_2) is also written (s_1, s_2).

The analysis of Barthe & Sørensen (2000) focuses on functional specifications.

Definition 2
$\mathbf{S} = (\mathscr{S}, \mathscr{A}, \mathscr{R})$ is *functional* if for every $s_1, s_2, s_2', s_3, s_3' \in \mathscr{S}$,

$$\begin{array}{lcll} (s_1, s_2) \in \mathscr{A} & \wedge & (s_1, s_2') \in \mathscr{A} & \Rightarrow \quad s_2 = s_2' \\ (s_1, s_2, s_3) \in \mathscr{R} & \wedge & (s_1, s_2, s_3') \in \mathscr{R} & \Rightarrow \quad s_3 = s_3' \end{array}$$

In this paper, our analysis focuses on the (functional) specifications $*$ and U^-. These specifications respectively correspond to Martin-Löf's original inconsistent type theory (Martin-Löf, 1971) and to Girard's System U^- (Girard, 1972).

Definition 3

1. The specification $U^- = (\mathscr{S}_{U^-}, \mathscr{A}_{U^-}, \mathscr{R}_{U^-})$ is defined by the clauses:
   - $\mathscr{S}_{U^-} = \{*, \Box, \triangle\}$
   - $\mathscr{A}_{U^-} = \{(*, \Box), (\Box, \triangle)\}$
   - $\mathscr{R}_{U^-} = \{(*, *), (\Box, \Box), (\Box, *), (\triangle, \Box)\}$
2. The specification $*$ is defined by the triple $(\{*\}, \{(*, *)\}, \{(*, *)\})$.

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

The specifications U^- and $*$ are examples of impredicative specifications, i.e. specifications that contain as a subsystem the polymorphic λ-calculus of Girard and Reynolds.

Definition 4

A specification $\mathbf{S} = (\mathscr{S}, \mathscr{A}, \mathscr{R})$ is *impredicative* w.r.t. the sorts $*$ and $\square$ if $* : \square$ is an axiom and $(*, *)$ and $(\square, *)$ are rules.

In the sequel, we shall often talk about impredicative specifications, leaving $*$ and $\square$ implicit.

2.2 Domain-Free Pure Type Systems

Every specification $\mathbf{S} = (\mathscr{S}, \mathscr{A}, \mathscr{R})$ yields a DFPTS $\underline{\lambda}\mathbf{S}$ as follows.

Definition 5

Let V denote a fixed, countably infinite, set of variables.

1. The set $\underline{\mathscr{E}}$ of *(domain-free) expressions* is given by the abstract syntax
$$\underline{\mathscr{E}} = V \mid \mathscr{S} \mid \underline{\mathscr{E}}\,\underline{\mathscr{E}} \mid \lambda V.\underline{\mathscr{E}} \mid \Pi V : \underline{\mathscr{E}}.\underline{\mathscr{E}}$$
We use $A \to B$ as an abbreviation for $\Pi x{:}A.\, B$ when $x \notin \mathrm{FV}(B)$.
2. A *(domain-free) context* is a finite sequence of assertions of the form $x{:}A$ with $x \in V$ and $A \in \underline{\mathscr{E}}$. If $\Gamma = x_1{:}A_1, \ldots, x_n{:}A_n$ is a context, we let $\mathrm{dom}(\Gamma)$ denote the set $\{x_1, \ldots, x_n\}$.
3. $\underline{\beta}$*-reduction* on $\underline{\mathscr{E}}$ is defined as the compatible closure of the contraction rule
$$(\lambda x.M)\ N \to_{\underline{\beta}} M\{x := N\}$$
where $\bullet\{\bullet := \bullet\}$ is the obvious substitution operator.
4. The *derivability* relation $\Vdash$ is given by the rules of figure 2. If $\Gamma \Vdash A : B$ then Γ, A and B are *legal*. Sometimes we write $\Gamma \Vdash_{\mathbf{S}} M : A$ instead of $\Gamma \Vdash M : A$ to make the dependence of the derivability relation on $\mathbf{S}$ explicit.

Barthe & Sørensen (2000) show that DFPTSs enjoy most, but not all, properties properties of PTSs. We complete their analysis by showing that strengthening fails for some DFPTSs; in contrast, recall that strengthening holds for an arbitrary Pure Type System (van Benthem Jutting, 1993).

Lemma 6

There exist specifications $\mathbf{S}$ for which the following implication fails:

$$\Gamma_1, x : A, \Gamma_2 \Vdash_{\mathbf{S}} M : B \quad \wedge \quad x \notin \mathrm{FV}(\Gamma_2) \cup \mathrm{FV}(M) \cup \mathrm{FV}(B) \quad \Rightarrow \quad \Gamma_1, \Gamma_2 \Vdash_{\mathbf{S}} M : B$$

Proof

Consider the PTS $\underline{\lambda}E$ given by the specification $E = (\mathscr{S}_E, \mathscr{A}_E, \mathscr{R}_E)$:

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

(axiom)	$\Vdash s_1 : s_2$	if $(s_1, s_2) \in \mathscr{A}$
(start)	$\dfrac{\Gamma \Vdash A : s}{\Gamma, x{:}A \Vdash x : A}$	if $x \notin \mathrm{dom}(\Gamma)$
(weakening)	$\dfrac{\Gamma \Vdash A : B \qquad \Gamma \Vdash C : s}{\Gamma, x{:}C \Vdash A : B}$	if $x \notin \mathrm{dom}(\Gamma)$
(product)	$\dfrac{\Gamma \Vdash A : s_1 \qquad \Gamma, x{:}A \Vdash B : s_2}{\Gamma \Vdash (\Pi x{:}A.\, B) : s_3}$	if $(s_1, s_2, s_3) \in \mathscr{R}$
(application)	$\dfrac{\Gamma \Vdash F : (\Pi x{:}A.\, B) \qquad \Gamma \Vdash a : A}{\Gamma \Vdash F\, a : B\{x := a\}}$	
(abstraction)	$\dfrac{\Gamma, x{:}A \Vdash b : B \qquad \Gamma \Vdash (\Pi x{:}A.\, B) : s}{\Gamma \Vdash \lambda x\,.\, b : \Pi x{:}A.\, B}$	
(conversion)	$\dfrac{\Gamma \Vdash A : B \qquad \Gamma \Vdash B' : s}{\Gamma \Vdash A : B'}$	if $B =_{\underline{\beta}} B'$

Fig. 2. Domain-Free Pure Type Systems.

- $\mathscr{S}_E = \{*^s, \Box^s, *^p, \Box^p\}$
- $\mathscr{A}_E = \{(*^s, \Box^s), (*^p, \Box^p)\}$
- $\mathscr{R}_E = \{(*^s, *^s), (*^s, *^p)\}$

Now take $M = (\lambda z\,.\, w)\, \lambda y\,.\, y$, and $\Gamma_1 = B : *^p, w : B$, and $A = *^s$ and Γ_2 to be the empty sequence.

- $B : *^p, w : B, x : *^s \Vdash_{\mathbf{S}} M : B$ is derivable, with intermediate steps

$$B : *^p, w : B, x : *^s \Vdash_{\mathbf{S}} \lambda z\,.\, w : (x \to x) \to B$$
$$B : *^p, w : B, x : *^s \Vdash_{\mathbf{S}} \lambda y\,.\, y : x \to x$$

- $B : *^p, w : B \Vdash_{\mathbf{S}} M : B$ is not derivable because there is no C such that $B : *^p, w : B \Vdash_{\mathbf{S}} \lambda y\,.\, y : C$ is derivable.

□

On the positive side, we show in Appendix B that strengthening holds for $M \in \mathrm{NF}_\beta$.

2.3 *Pure Type Systems*

Every specification $\mathbf{S} = (\mathscr{S}, \mathscr{A}, \mathscr{R})$ yields a Pure Type System $\lambda\mathbf{S}$ as follows.

Definition 7
Let V denote a fixed, countably infinite, set of variables.

1. The set $\mathscr{E}$ of *(domain-full) expressions* is given by the abstract syntax:

$$\mathscr{E} = V \mid \mathscr{S} \mid \mathscr{E}\mathscr{E} \mid \lambda V : \mathscr{E}.\mathscr{E} \mid \Pi V : \mathscr{E}.\mathscr{E}$$

   We use $A \to B$ as an abbreviation for $\Pi x{:}A.\, B$ when $x \notin \mathrm{FV}(B)$.

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

2. A *(domain-full) context* is a finite sequence of assertions of the form $x{:}A$ with $x \in V$ and $A \in \mathscr{E}$. If $\Gamma = x_1{:}A_1, \ldots, x_n{:}A_n$ is a context, we let $\mathrm{dom}(\Gamma)$ denote the set $\{x_1, \ldots, x_n\}$.
3. *β-reduction* $\rightarrow_\beta$ on $\mathscr{E}$ is defined as the compatible closure of the contraction

$$(\lambda x : A.M)\ N \rightarrow_\beta M\{x := N\}$$

where $\bullet\{\bullet := \bullet\}$ is the obvious substitution operator.
4. The *derivability* relation $\vdash$ is given by the rules of figure 2, except for the (abstraction) rule which becomes

$$\frac{\Gamma, x{:}A \vdash b : B \qquad \Gamma \vdash (\Pi x{:}A.\,B) : s}{\Gamma \vdash \lambda x{:}A\,.\,b : \Pi x{:}A.\,B}$$

and the (conversion) rule which becomes

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s}{\Gamma \vdash A : B'} \qquad \text{if } B =_\beta B'$$

We adopt the same conventions as for domain-free derivability.

In the sequel, we focus on normalizing PTSs.

Definition 8

1. A PTS $\lambda\mathbf{S}$ is β-weakly normalizing (or normalizing for short) if

$$\Gamma \vdash M : A \quad \Rightarrow \quad M \in \mathrm{WN}(\beta)$$

2. A PTS $\lambda\mathbf{S}$ is β-weakly normalizing on types (or type-normalizing for short) if

$$[\Gamma \vdash M : s \quad \wedge \quad s \in \mathscr{S}] \quad \Rightarrow \quad M \in \mathrm{WN}(\beta)$$

Most systems that appear in the literature are normalizing, but the PTSs λU^- and $\lambda *$ are not. In fact, these two systems are well-known examples of inconsistent PTSs, i.e. systems for which one can construct an expression H such that $\Gamma \vdash H : \Pi\alpha{:}*.\,*$, see Girard (1972) for the original construction. Such an expression H is a proof of inconsistency, or a paradox, because $\Pi\alpha{:}*.\,\alpha$ is the encoding of falsity in these systems. As is well-known, every inconsistent PTS is also non-normalizing hence λU^- and $\lambda *$ are non-normalizing. However, λU^- is type-normalizing.

2.4 Erasure and equivalence properties

There is an obvious erasure function $|.| : \mathscr{E} \rightarrow \underline{\mathscr{E}}$ which removes domains from λ-abstractions:

$$\begin{array}{rcl} |x| & = & x \\ |s| & = & s \\ |M\ N| & = & |M|\,|N| \\ |\lambda x{:}A\,.\,M| & = & \lambda x\,.\,|M| \\ |\Pi x{:}A.\,B| & = & \Pi x{:}|A|.\,|B| \end{array}$$

$|.|$ can be extended to contexts in the obvious way and preserves typing in the sense that for every specification $\mathbf{S}$,

$$\Gamma \vdash M : A \quad \Rightarrow \quad |\Gamma| \vdash |M| : |A|$$

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

The erasure function is useful for stating the equivalence properties between PTSs and DFPTSs.

Preservation of Typing A specification **S** enjoys Preservation of Typing, written PTY(**S**), if, for every derivable judgement $\Gamma \Vdash_{\mathbf{S}} M : A$ there exists a derivable judgement $\Gamma' \vdash_{\mathbf{S}} M' : A'$ with $|\Gamma'| = \Gamma$, $|M'| = M$ and $|A'| = A$.

Preservation of Equational Theory A specification **S** enjoys Preservation of Equational Theory, written PET(**S**), if:

$$[\Gamma \vdash_{\lambda\mathbf{S}} M : A \quad \wedge \quad \Gamma \vdash_{\lambda\mathbf{S}} N : A \quad \wedge \quad |M| =_{\underline{\beta}} |N|] \quad \Rightarrow \quad M =_{\beta} N$$

Lemma 9

A specification **S** enjoys Preservation of Equational Theory iff

$$[\Gamma \vdash_{\lambda\mathbf{S}} M : A \quad \wedge \quad \Gamma \vdash_{\lambda\mathbf{S}} N : A \quad \wedge \quad |M| = |N|] \quad \Rightarrow \quad M =_{\beta} N$$

Proof

Only the reverse implication is not trivial. So assume

$$\Gamma \vdash_{\lambda\mathbf{S}} M : A \quad \wedge \quad \Gamma \vdash_{\lambda\mathbf{S}} N : A \quad \wedge \quad |M| =_{\underline{\beta}} |N|$$

By confluence of $\underline{\beta}$-reduction, $|M| =_{\underline{\beta}} |N|$ iff $|M| \twoheadrightarrow_{\underline{\beta}} P$ and $|N| \twoheadrightarrow_{\underline{\beta}} P$. We proceed by induction on the length of the reduction sequences. In case the length of both sequences is 0, we are done by assumption. Otherwise one reduction sequence, say that of $|M|$, is of the form $|M| \rightarrow_{\underline{\beta}} M' \twoheadrightarrow_{\underline{\beta}} P$. Now there exists $M'' \in \mathscr{E}$ such that $M \rightarrow_{\beta} M''$ and $|M''| = M'$. By Subject Reduction $\Gamma \vdash_{\lambda\mathbf{S}} M'' : A$ so we can apply our induction hypothesis to conlude $M'' =_{\beta} M'$, from which $M =_{\beta} M'$ follows. □

A previous analysis (Barthe & Sørensen, 2000) establishes the equivalence properties for some classes of functional PTSs.

Proposition 10

Let $\lambda\mathbf{S}$ be a functional PTS.

1. If $\lambda\mathbf{S}$ is normalizing then PET(**S**).
2. If $\lambda\mathbf{S}$ is type-normalizing then PTY(**S**).

The question arises whether Preservation of Equational Theory holds for non-normalizing PTSs. In section 4 we give a negative answer by showing that erasure does not reflect the equational theory of λU^- and $\lambda *$. That is, we show $\neg$PET(U^-) and $\neg$PET($*$).

3 A fixpoint combinator for λU^-

We define an expression Y_H that is a looping combinator, but not a fixpoint combinator for λU^-. Furthermore the erasure Y of Y_H is a fixpoint combinator for $\underline{\lambda} U^-$. The expression Y_H is derived from Hurkens' paradox for λU^- (Hurkens, 1995).

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

3.1 Fixpoint and looping combinators

We begin by reviewing the notions of fixpoint and looping combinators. The definitions are given for DFPTSs, but they adapt readily to PTSs.

Definition 11
Let $\mathbf{S} = (\mathscr{S}, \mathscr{A}, \mathscr{R})$ be an impredicative specification.

1. A *fixpoint combinator* is an expression Y such that
$$\Vdash_{\mathbf{S}} Y : \Pi A{:}*. (A \to A) \to A$$
and $Y\ A\ f =_{\underline{\beta}} f\ (Y\ A\ f)$ for every $A, f \in \underline{\mathscr{E}}$ (it would be equivalent to require $A, f \in V$). We write $\mathsf{FIX}(\underline{\lambda}\mathbf{S})$ if there exists a fixpoint combinator in $\underline{\lambda}\mathbf{S}$.
2. A *looping combinator* is a family of expressions $(Y_n)_{n \in I\!N}$ such that for every $i \in I\!N$
$$\Vdash_{\mathbf{S}} Y_i : \Pi A{:}*. (A \to A) \to A$$
and $Y_n\ A\ f =_{\underline{\beta}} f\ (Y_{n+1}\ A\ f)$ for every $A, f \in \underline{\mathscr{E}}$ (it would be equivalent to require $A, f \in V$). By abuse of notation, we say that an expression Y is a looping combinator if there exists a looping combinator $(Y_n)_{n \in I\!N}$ with $Y = Y_0$. We write $\mathsf{LOOP}(\underline{\lambda}\mathbf{S})$ if there exists a looping combinator in $\underline{\lambda}\mathbf{S}$.

The notion of looping combinator originates from the work of Meyer & Reinhold (1986), who suggested a method to transform Girard's paradox in $\lambda*$ (Girard, 1972) into a fixpoint combinator for $\lambda*$. This transformation was carried out by Howe (1987), who also established that the resulting term is in fact a *looping* combinator and not a *fixpoint* combinator; this is closely connected to the fact that the term G does not reduce to itself, but to a more complicated term (Coquand, 1986). However, the existence of a looping combinator is enough to get undecidability of type checking in $\lambda*$, see Reinhold (1989) which also provides an interesting account of related work up to 1989. More recently, Coquand & Herbelin (1994) presented a systematic way to show the existence of a looping combinator in so-called inconsistent logical PTSs, and showed that under certain conditions these inconsistent logical PTSs have undecidable type-checking.

3.2 PET, fixpoint and looping combinators

Erasure preserves fixpoint and looping combinators (Barthe & Sørensen, 2000).

Lemma 12
Let $\mathbf{S}$ be an impredicative specification.

1. If $\mathsf{FIX}(\lambda\mathbf{S})$ then $\mathsf{FIX}(\underline{\lambda}\mathbf{S})$.
2. If $\mathsf{LOOP}(\lambda\mathbf{S})$ then $\mathsf{LOOP}(\underline{\lambda}\mathbf{S})$.

A form of converse holds for those specifications that enjoy Preservation of Equational Theory.

Lemma 13
Let $\mathbf{S}$ be an impredicative specification such that $\mathsf{PET}(\mathbf{S})$.

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

1. If $\Gamma \vdash_S Y : \Pi A{:}*. (A \to A) \to A$ and $|Y|$ is a fixpoint combinator in $\underline{\lambda}\mathbf{S}$ then Y is a fixpoint combinator in $\lambda\mathbf{S}$.
2. If $\Gamma \vdash_S Y : \Pi A{:}*. (A \to A) \to A$ and $|Y|$ is a looping combinator in $\underline{\lambda}\mathbf{S}$ then Y is a looping combinator in $\lambda\mathbf{S}$.

Proof

We only prove 1, but 2 is proved in a similar way. Assume that $\Gamma \vdash Y : \Pi A{:}*. (A \to A) \to A$ and $|Y|$ is a fixpoint combinator in $\underline{\lambda}\mathbf{S}$. Clearly $|Y\ A\ f| =_{\underline{\beta}} |f\ (Y\ A\ f)|$ for every $A, f \in \mathscr{E}$. It follows from PET(**S**) that $Y\ A\ f =_\beta f\ (Y\ A\ f)$ so that Y is a fixpoint combinator for $\lambda\mathbf{S}$. □

3.3 A fixpoint combinator for λU^-

Hurkens (1995) presents a remarkably simple paradox H for λU^-, with the striking property that H reduces to itself (non-trivially). One can transform H into a looping combinator Y_H by using the idea of Meyer and Reinhold. In a nutshell, their idea is to replace $\neg\ \phi$ by $\phi \to A$, where A is a fresh type variable, and to introduce a fresh function symbol f of type $A \to A$ at appropriate places in H.

Definition 14

1. The (domain-full) expression Y_H is defined as $\lambda A{:}*. \lambda f{:}A \to A. L\ R$, where L and R are given in figure 3. One has $\vdash_{U^-} Y_H : \Pi A{:}*. (A \to A) \to A$.
2. The (domain-free) expression Y is defined as $|Y_H|$. One has $\Vdash_{U^-} Y : \Pi A{:}*. (A \to A) \to A$.

As already noted by Geuvers and Pollack in 1994, Y is a fixpoint combinator for $\underline{\lambda} U^-$. However, Y_H is a looping combinator but not a fixpoint combinator for λU^-.

Theorem 15

1. Y is a fixpoint combinator for $\underline{\lambda} U^-$.
2. Y_H is a looping combinator for λU^-.
3. Y_H is not a fixpoint combinator for λU^-.

Proof

1. We look at the corresponding domain free erasure of the terms of figure 3. By abuse of notation we write B, C and ρ for $|B|$, $|C|$ and $|\rho|$, respectively. We have:

$$\begin{aligned} Y &= \lambda A. \lambda f. L\ R \\ L &= \lambda d. d\ B\ M\ (\lambda p. d\ (\lambda y. p\ (\rho\ y))) \\ M &= \lambda x. \lambda k. \lambda l. f\ (l\ B\ k\ (\lambda p. l\ (\lambda y. p\ (\rho\ y)))) \\ R &= \lambda p. \lambda h. h\ C\ (\lambda x. h\ (\rho\ x)) \end{aligned}$$

A simple calculation shows that Y is a fixpoint combinator:

$$\begin{aligned} Y\ A\ f &\twoheadrightarrow_{\underline{\beta}} L\ R \\ &\to_{\underline{\beta}} R\ B\ M\ (\lambda p. R\ (\lambda y. p\ (\rho\ y))) \\ &\to_{\underline{\beta}} R\ B\ M\ R \end{aligned}$$

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

All terms are given with their types in the context $A : *, f : A \to A$:

Preliminary definitions

$$\begin{array}{lcll}
\mathscr{P} & : & \Box \to \Box & = \lambda X : \Box.\ X \to * \\
\mathscr{P}^2 & : & \Box \to \Box & = \lambda X : \Box. \mathscr{P}\ (\mathscr{P}\ X) \\
\neg & : & * \to * & = \lambda \phi : *.\ \phi \to A
\end{array}$$

Definition of a paradoxical universe

$$\begin{array}{lcll}
\mathscr{U} & : & \Box & = \Pi X : \Box.\ ((\mathscr{P}^2\ X) \to X) \to (\mathscr{P}^2\ X) \\
\tau & : & (\mathscr{P}^2\ \mathscr{U}) \to \mathscr{U} & = \lambda t : \mathscr{P}^2\ \mathscr{U}.\ \lambda X : \Box.\ \lambda g : (\mathscr{P}^2\ X) \to X.\ \lambda p : \mathscr{P}\ X. \\
 & & & \quad t\ (\lambda x : \mathscr{U}.\ p\ (g\ (x\ X\ g))) \\
\sigma & : & \mathscr{U} \to (\mathscr{P}^2\ \mathscr{U}) & = \lambda s : \mathscr{U}.\ s\ \mathscr{U}\ \tau \\
\rho & : & \mathscr{U} \to \mathscr{U} & = \lambda y : \mathscr{U}.\ \tau\ (\sigma\ y)
\end{array}$$

Intermediate Definitions

$$\begin{array}{lcll}
E & : & * & = \Pi p : \mathscr{P}\ \mathscr{U}.\ (\sigma\ x\ p) \to p\ (\tau\ (\sigma\ x)) \\
Q & : & \mathscr{P}\ (\mathscr{P}\ \mathscr{U}) & = \lambda p : \mathscr{P}\ \mathscr{U}.\ \Pi x : \mathscr{U}.\ (\sigma\ x\ p) \to (p\ x) \\
B & : & \mathscr{P}\ \mathscr{U} & = \lambda x : \mathscr{U}.\ \neg(E\ x) \\
C & : & \mathscr{U} & = \tau\ Q \\
D & : & * & = \Pi p : \mathscr{P}\ \mathscr{U}.\ Q\ p \to p\ C
\end{array}$$

Definition of the paradox

$$\begin{array}{lcll}
M & : & Q\ B & = \lambda x : \mathscr{U}.\ \lambda k : \sigma\ x\ B.\ \lambda l : E. \\
 & & & \quad f\ (l\ B\ k\ (\lambda p : \mathscr{P}\ \mathscr{U}.\ l\ (\lambda y : \mathscr{U}.\ p\ (\rho\ y)))) \\
L & : & \neg D & = \lambda d : D.\ d\ B\ M\ (\lambda p : \mathscr{P}\ \mathscr{U}.\ d\ (\lambda y : \mathscr{U}.\ p\ (\rho\ y))) \\
R & : & D & = \lambda p : \mathscr{P}\mathscr{U}.\ \lambda h : Q\ p.\ h\ C\ (\lambda x : \mathscr{U}.\ h\ (\rho\ x))
\end{array}$$

Fig. 3. The expression Y_H.

$$\begin{array}{ll}
\twoheadrightarrow_{\underline{\beta}} & M\ C\ (\lambda y.\ M\ (\rho\ y))\ R \\
\to_{\underline{\beta}} & M\ C\ M\ R \\
\twoheadrightarrow_{\underline{\beta}} & f\ (R\ B\ M\ (\lambda p.\ R\ (\lambda y.\ p\ (\rho\ y)))) \\
\to_{\underline{\beta}} & f\ (R\ B\ M\ R)
\end{array}$$

As the reduction sequence contains $Y\ A\ f \twoheadrightarrow_{\underline{\beta}} R\ B\ M\ R$ as a subsequence, one concludes that $Y\ A\ f =_{\underline{\beta}} f\ (Y\ A\ f)$.

2. By a simple calculation.
3. See Appendix A.

□

The results adapt readily to $\lambda *$ and $\underline{\lambda} *$.

Proposition 16

Let Y'_H and Y' be the expressions obtained from Y_H and Y by replacing every occurence of □ by $*$.

1. Y' is a fixpoint combinator for $\underline{\lambda} *$.
2. Y'_H is a looping combinator for $\lambda *$.
3. Y'_H is not a fixpoint combinator for $\lambda *$.

Proof

One can repeat the calculations and analyses of Theorem 15. Alternatively, one can derive 1 and 2 from Theorem 15.1 and Theorem 15.2 by noting that Y' and Y'_H

are the images of Y and Y_H under the unique PTS-morphism from U^- to $*$ and by using the fact that PTS-morphisms preserve typing and convertibility (e.g. see Geuvers (1993)). □

In the sequel, we blur the distinction between Y and Y' (resp. Y_H and Y'_H), and simply write Y (resp. Y_H) for both.

4 Applications to non-preservation of equational theories

The results of the previous section imply that Preservation of Equational Theory fails both for U^- and $*$.

Theorem 17

1. $\neg\mathsf{PET}(U^-)$
2. $\neg\mathsf{PET}(*)$

Proof

1. Assume $\mathsf{PET}(U^-)$. By Lemma 13.1 and Theorem 15.1, Y_H is a fixpoint combinator for λU^-, which contradicts Theorem 15.3.
2. Proceed as in 1 or use a more direct argument that does not rely on Proposition 16.3, whose proof is extremly tedious. The direct proof proceeds in two steps: first, one proves that $\mathsf{PET}(*) \Rightarrow \mathsf{FIX}(\lambda *)$ as in 1 above. Then one uses $\mathsf{FIX}(\lambda *) \Rightarrow \neg\, \mathsf{PET}(*)$, see Lemma 18 below, to conclude that $\mathsf{PET}(*) \Rightarrow \neg\mathsf{PET}(*)$, hence $\neg\mathsf{PET}(*)$.

□

The next lemma, which is used in the proof of Theorem 17, is adapted from Geuvers & Werner (1994) – we review their original argument in section 5.

Lemma 18
$\mathsf{FIX}(\lambda *) \Rightarrow \neg\, \mathsf{PET}(*)$

Proof
Assume that Y is a fixpoint combinator in $\lambda *$. Then let $\Gamma = \alpha : *, \alpha' : *, \delta : *$ and define

$$\begin{aligned} A &= Y\ *\ (\lambda\epsilon : *.\ \epsilon \to (\alpha \to \alpha) \to \delta) \\ A' &= Y\ *\ (\lambda\epsilon : *.\ \epsilon \to (\alpha' \to \alpha') \to \delta) \end{aligned}$$

By definition of a fixpoint combinator, one has

$$\begin{aligned} A &=_\beta A \to (\alpha \to \alpha) \to \delta \\ A' &=_\beta A' \to (\alpha' \to \alpha') \to \delta \end{aligned}$$

Now let

$$\begin{aligned} M &= \lambda y : A.\ y\ y \\ M' &= \lambda y : A'.\ y\ y \\ N &= M\ M\ (\lambda z : \alpha.\ z) \\ N' &= M'\ M'\ (\lambda z : \alpha'.\ z) \end{aligned}$$

Clearly

$$|N| = (\lambda y.\ y\ y)\ (\lambda y.\ y\ y)\ (\lambda z.\ z) = |N'|$$

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

but we do not have $N =_\beta N'$. Indeed, every β-reduct of N is of the form

$$(\lambda y : A_1.\ y\ y)\ (\lambda y : A_2.\ y\ y)\ (\lambda z : \alpha.\ z)$$

and similarly every β-reduct of N' is of the form

$$(\lambda y : A_1.\ y\ y)\ (\lambda y : A_2.\ y\ y)\ (\lambda z : \alpha'.\ z)$$

so N and N' cannot have a common reduct under β-reduction. By confluence of β-reduction, it follows that N and N' are not β-convertible. □

5 Applications to non-confluence of $\beta\eta$-reduction in $* : *$

5.1 Definitions and general results

Following Geuvers (1992, 1993), we associate to every specification **S** an Extensional Pure Type System $\lambda^{\beta\eta}\mathbf{S}$ whose conversion rule is modified to deal with a richer convertibility relation that includes η-conversion.

Definition 19

1. η-reduction $\to_\eta$ on $\mathscr{E}$ is defined as the compatible closure of the contraction

$$\lambda x : A\,.\,M\,x \quad \to_\eta \quad M \quad \text{if } x \notin \mathrm{FV}(M)$$

2. $\beta\eta$-reduction $\to_{\beta\eta}$ is defined as $\to_\beta \cup \to_\eta$.
3. The derivability relation $\vdash^{\beta\eta}$ is given by the rules of PTSs, except for the (conversion) rule which becomes

$$\frac{\Gamma \vdash^{\beta\eta} A : B \qquad \Gamma \vdash^{\beta\eta} B' : s}{\Gamma \vdash^{\beta\eta} A : B'} \qquad \text{if } B =_{\beta\eta} B'$$

We adopt the same conventions as for other derivability relations.

The following well-known example, due to Nederpelt, shows that $\beta\eta$-reduction is not confluent on the set $\mathscr{E}$ of expressions

$$\lambda x : A\,.\,M \quad {}_\eta\!\leftarrow \quad \lambda x : B\,.\,\lambda x : A\,.\,M\,x \quad \to_\beta \quad \lambda x : B\,.\,M$$

One may wonder if confluence holds for legal terms.

Definition 20

A Pure Type System $\lambda^{\beta\eta}\mathbf{S}$ enjoys confluence of $\beta\eta$-reduction on legal terms, written $\mathsf{CON}(\lambda^{\beta\eta}\mathbf{S})$, if

$$[\Gamma \vdash^{\beta\eta}_{\mathbf{S}} M : A \quad \wedge \quad \Gamma \vdash^{\beta\eta}_{\mathbf{S}} M' : A \quad \wedge \quad M =_{\beta\eta} N] \quad \Rightarrow \quad M \downarrow_{\beta\eta} N$$

Geuvers (1992; 1993) showed $\mathsf{CON}(\lambda^{\beta\eta}\mathbf{S})$ for functional and β-weakly normalizing PTSs $\lambda^{\beta\eta}\mathbf{S}$, and also $\mathsf{CON}(\lambda^{\beta\eta}U)$, where the specification U is obtained from U^- by adding the rule $(\triangle, *)$; in fact, Geuvers' argument for $\lambda^{\beta\eta}U$ also applies to $\lambda^{\beta\eta}U^-$. Summarizing

Theorem 21 (*Geuvers*)

1. If $\lambda^{\beta\eta}\mathbf{S}$ is functional and β-weakly normalizing, then $\mathsf{CON}(\lambda^{\beta\eta}\mathbf{S})$.
2. $\mathsf{CON}(\lambda^{\beta\eta}U^-)$ and $\mathsf{CON}(\lambda^{\beta\eta}U)$.

We can also drop the assumption about functionality in the first item of the above theorem, see (Barthe, 1999).

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

5.2 *Non-confluence of $\beta\eta$-reduction in $\lambda^{\beta\eta}*$*

The question arises whether a similar result holds for $\lambda^{\beta\eta}*$. Using Theorem 15, we give a negative answer to the question by establishing $\neg\mathsf{CON}(\lambda^{\beta\eta}*)$. Our proof relies on two observations:

- the first observation, due to Geuvers (1993, Lemma 4.4.16, p. 98), is that Nederpelt's counter-example provides an immediate proof that any two domain-full expressions whose erasure $\underline{\beta}$-convertible are themselves $\beta\eta$-convertible. Lemma 22 is a corollary of Geuvers' observation.

  Lemma 22 (*Geuvers*)
  For every $M, N \in \mathscr{E}$,

  $$|M| =_{\underline{\beta}} |N| \quad \Rightarrow \quad M =_{\beta\eta} N$$

  This observation allows to deduce that $\lambda^{\beta\eta}U^-$ and $\lambda^{\beta\eta}*$ have a fixpoint combinator.

  Lemma 23
  $\mathsf{FIX}(\lambda^{\beta\eta}U^-)$ and $\mathsf{FIX}(\lambda^{\beta\eta}*)$.

  Proof
  We only prove $\mathsf{FIX}(\lambda^{\beta\eta}U^-)$ as the proof of $\mathsf{FIX}(\lambda^{\beta\eta}*)$ is similar. By Theorem 15, the expression Y_H is such that $|Y_H| = Y$ is a fixpoint combinator and verifies $\vdash_{U^-} Y_H : \Pi A:*.\,(A \to A) \to A$. The derivability relation $\vdash^{\beta\eta}$ extends the derivability relation $\vdash$, hence $\vdash^{\beta\eta}_{U^-} Y_H : \Pi A:*.\,(A \to A) \to A$. Furthermore, assume $A, f \in \mathscr{E}$. We have

  $$|Y_H\ A\ f| = Y\ A\ f =_{\underline{\beta}} f\ (Y\ A\ f) = |f\ (Y_H\ A\ f)|$$

  and hence by Lemma 22, $Y_H\ A\ f =_{\beta\eta} f\ (Y_H\ A\ f)$. It follows that Y_H is a fixpoint combinator for $\lambda^{\beta\eta}U^-$. □

- the second observation, due to Geuvers & Werner (1994), is that $\mathsf{CON}(\lambda^{\beta\eta}*)$ and $\mathsf{FIX}(\lambda^{\beta\eta}*)$ exclude each other. The proof uses the definitions of Lemma 18 – recall that the proof of the latter is an adaptation of Geuvers and Werner's original proof of this lemma.

  Lemma 24 (*Geuvers and Werner*)
  $\mathsf{FIX}(\lambda^{\beta\eta}*) \quad \Rightarrow \quad \neg\mathsf{CON}(\lambda^{\beta\eta}*)$

  Proof
  Observe that the forms $(\lambda y : A_1.\ y\ y)\ (\lambda y : A_2.\ y\ y)\ (\lambda z : \alpha.\ z)$ and $(\lambda y : A_1.\ y\ y)\ (\lambda y : A_2.\ y\ y)\ (\lambda z : \alpha'.\ z)$ are closed under $\beta\eta$-reduction so N and N' cannot have a common reduct under $\beta\eta$-reduction. However $|N| = |N'|$, hence by Lemma 22, $N =_{\beta\eta} N'$. □

The non-confluence of $\beta\eta$-reduction on legal terms of $\lambda^{\beta\eta}*$ is an immediate corollary of Lemmas 23 and 24.

Corollary 25
$\neg\mathsf{CON}(\lambda^{\beta\eta}*)$

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

5.3 Domain-Free Pure Type Systems with $\underline{\beta\eta}$-conversion?

Corollary 25 and more generally the difficulties with $\beta\eta$-reduction in a domain-full setting make it tempting to switch to a domain-free setting, where confluence of $\underline{\beta\eta}$-reduction holds. Every specification **S** yields a DFPTS with $\underline{\beta\eta}$-conversion $\underline{\lambda}^{\underline{\beta\eta}}\mathbf{S}$ as follows.

Definition 26

1. $\underline{\eta}$-reduction $\rightarrow_{\underline{\eta}}$ on $\underline{\mathscr{E}}$ is defined as the compatible closure of the contraction

$$\lambda x\,.\,M\,x \quad \rightarrow_{\underline{\eta}} \quad M \quad \text{if } x \notin \mathrm{FV}(M)$$

2. $\underline{\beta\eta}$-reduction $\rightarrow_{\underline{\beta\eta}}$ is defined as $\rightarrow_{\underline{\beta}} \cup \rightarrow_{\underline{\eta}}$.
3. The derivability relation $\Vdash^{\underline{\beta\eta}}$ is given by the rules of DFPTSs, except for the (conversion) rule which becomes

$$\frac{\Gamma \Vdash^{\underline{\beta\eta}} A : B \qquad \Gamma \Vdash^{\underline{\beta\eta}} B' : s}{\Gamma \Vdash^{\underline{\beta\eta}} A : B'} \qquad \text{if } B =_{\underline{\beta\eta}} B'$$

   We adopt the same conventions as for other derivability relations.

Unfortunately, subject reduction fails for some DFPTSs with $\underline{\beta\eta}$-conversion.

Lemma 27
There exist specifications **S** for which the following implication fails:

$$\Gamma \Vdash^{\underline{\beta\eta}}_{\mathbf{S}} M : B \;\wedge\; M \rightarrow_{\underline{\eta}} M' \quad \Rightarrow \quad \Gamma \Vdash^{\underline{\beta\eta}}_{\mathbf{S}} M' : B$$

Proof
Consider the PTS $\underline{\lambda}^{\underline{\beta\eta}}F$ given by the specification $F = (\mathscr{S}_F, \mathscr{A}_F, \mathscr{R}_F)$:

- $\mathscr{S}_F = \{*, \square, \triangle\}$
- $\mathscr{A}_F = \{(*, \square)\}$
- $\mathscr{R}_F = \{(*, *), (\square, *, \triangle), (*, \triangle)\}$

Now set $\bot = \Pi A{:}*.\,A$ and $M = \lambda A\,.\,(\lambda z\,.\,x)\,(\lambda y\,.\,y)\,A$, and $M' = (\lambda z\,.\,x)\,(\lambda y\,.\,y)$, and $\Gamma = x : \bot$ and $B = \bot$. We have:

- $x : \bot \Vdash^{\underline{\beta\eta}}_{F} M : \bot$ is derivable with intermediate steps

$$x : \bot\, A : * \Vdash^{\underline{\beta\eta}} \lambda z\,.\,x : (A \rightarrow A) \rightarrow \bot$$
$$A : * \Vdash^{\underline{\beta\eta}} \lambda y\,.\,y : A \rightarrow A$$

- $x : \bot \Vdash^{\underline{\beta\eta}}_{F} M' : \bot$ is not derivable as there is no C s.t. $x : \bot \Vdash^{\underline{\beta\eta}} \lambda y\,.\,y : C$.

□

On the positive side, we show in the appendix that $\underline{\eta}$-reduction holds provided $M \in \mathrm{NF}_{\underline{\beta}}$ and that DFPTSs with $\underline{\beta\eta}$-conversion are sound.

6 Conclusion

This paper contributes to the study of fixpoint combinators in inconsistent PTSs and resolves some of the questions that arose from previous work on DFPTSs, in

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

particular the preservation of equational theory for $*$ and the existence of a fixpoint combinator in $\underline{\lambda}*$. However, the question of the existence of a fixpoint combinator in $\lambda*$, i.e. $\mathsf{FIX}(\lambda*)$, remains open.

Appendix A: proof that Y_H is not a fixpoint combinator

We begin with some preliminary definitions and notations. Recall that weak-head reduction $\rightarrow_{wh}$ is the smallest relation such that

$$(\lambda x : A.\ P)\ P'\ \vec{R} \quad \rightarrow_{wh} \quad P\{x := P'\}\ \vec{R}$$

(Weak-head reduction differs from β-reduction by applying only at the top-level.)

Then we define for every $X, Y \subseteq \mathscr{E}$ the following sets and proposition:

- $X\ Y = \{H\ H' \mid H \in X\ \wedge\ H' \in Y\}$;
- $\lambda x : X.\ Y = \{\lambda x : H.\ H' \mid H \in X\ \wedge\ H' \in Y\}$;
- $\lambda x.\ Y = \{\lambda x : H.\ H' \mid H \in \mathscr{E}\ \wedge\ H' \in Y\}$;
- $X =_\beta Y$ iff $\exists H \in X,\ H' \in Y.\ H =_\beta H'$.

Next we give a necessary and sufficient condition for two expressions to be β-convertible.

Definition 28
Let $M, M' \in \mathscr{E}$. We say that $M \sim M'$ if:

- M is of the form $P_0\ \ldots\ P_k$,
- M' is of the form $P'_0\ \ldots\ P'_k$,
- both P_0 and P'_0 are not applications,
- $P_i =_\beta P_i$ for $1 \leqslant i \leqslant k$.

The next fact follows from the standardization theorem.

Fact
For every pseudo-terms $M, M' \in \mathscr{E}$, $M =_\beta M'$ iff there exists $N, N' \in \mathscr{E}$ such that

$$N \sim N'\ \wedge\ M \twoheadrightarrow_{wh} N\ \wedge\ M' \twoheadrightarrow_{wh} N'$$

We shall use this characterization to show that $Y_H\ A\ f \neq_\beta f\ (Y_H\ A\ f)$. We first define:

$$\begin{array}{rcl} N & = & \{H \in \mathscr{E} \mid H \twoheadrightarrow_{wh} \lambda k.\ \lambda l.\ f\ (l\ B\ k\ (\lambda p.\ l\ (\lambda y.\ p\ (\rho\ y))))\} \\ S & = & \{H \in \mathscr{E} \mid H \twoheadrightarrow_{wh} \lambda h.\ h\ C\ (\lambda x.\ h\ (\rho\ x))\} \\ S_n(p) & = & \{H \in \mathscr{E} \mid H \twoheadrightarrow_{wh} \lambda h : Q_n(p).\ h\ C\ (\lambda x.\ h\ (\rho\ x))\} \end{array}$$

where

$$\begin{array}{rcl} Q_0(p) & = & Qp \\ Q_{n+1}(p) & = & Q_n(\rho\ p) \end{array}$$

Note that for $Q_m(p) =_\beta Q_n(p)$ iff $m = n$.

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

Lemma 29

$$S\ (\lambda x\,.\,N)\ (\lambda p.\ S_m(p)) =_\beta S\ (\lambda x\,.\,N)\ (\lambda p.\ S_n(p)) \Leftrightarrow m = n$$

Proof
Consider the weak-head reduction sequence starting from a term of the form:

$$S\ (\lambda x\,.\,N)\ (\lambda p.\ S_n(p))$$

By a direct inspection it follows that the only reducts appearing in such a sequence are of the form:

- $S\ (\lambda x\,.\,N)\ (\lambda p.\ S_n(p))$;
- $X\ (\lambda x\,.\,N)\ (\lambda p\,.\,S_n(p))$;
- $(\lambda l.\ f\ (l\ B\ (\lambda x\,.\,N)\ (\lambda p.\ l\ (\lambda y.\ p\ (\rho\ y))))) \ (\lambda p.\ S_n(p))$;
- $f\ (S\ (\lambda x\,.\,N)\ (\lambda p.\ S_{n+1}(p)))$.

As $\lambda p.\ S_m(p) =_\beta \lambda p.\ S_n(p)$ iff $m = n$, the lemma follows from the fact. □

Now to conclude that

$$S\ (\lambda x\,.\,N)\ (\lambda p.\ S_0(p)) \neq_\beta f\ (S\ (\lambda x\,.\,N)\ (\lambda p.\ S_0(p)))$$

it is enough to notice that

$$S\ (\lambda x\,.\,N)\ (\lambda p.\ S_0(p)) \twoheadrightarrow_{wh} f\ (S\ (\lambda x\,.\,N)\ (\lambda p.\ S_1(p)))$$

and

$$S\ (\lambda x\,.\,N)\ (\lambda p.\ S_0(p)) \neq_\beta S\ (\lambda x\,.\,N)\ (\lambda p.\ S_1(p))$$

from the above lemma. Finally,

$$Y_H\ A\ f \neq_\beta f\ (Y_H\ A\ f)$$

follows from the fact that

$$Y_H\ A\ f =_\beta S\ (\lambda x\,.\,N)\ (\lambda p.\ S_0(p))$$

Remark The construction of Y_H relies on the existence of a paradoxical universe, that is of a type $\mathscr{U} : \Box$ and two functions $\sigma : (\mathscr{U} \to *) \to \mathscr{U}$ and $\tau : \mathscr{U} \to \mathscr{U} \to *$ such that

$$(\sigma(\tau\ X))\ y \Leftrightarrow \exists x : \mathscr{U}.\ X(x) \wedge y = \tau(\sigma x)$$

The results of this paper are independent of the choice of a specific paradoxical universe. In particular, to prove that Y_H is not a fixpoint combinator it is enough to know that we do not have

$$\tau(\sigma x) \twoheadrightarrow_\beta x$$

which allows us to deduce that $Q_n(p) =_\beta Q_m(p)$ iff $n = m$. In fact, Hurkens observes that the existence of a fixpoint combinator would follow from the existence of a paradoxical universe s.t.

$$\tau(\sigma x) \twoheadrightarrow_\beta x$$

but it is direct to show that no such paradoxical universe exists in λU^-, since all propositions are normalizable.

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

Appendix B: further remarks on strengthening and soundness in DFPTSs

Lemma 30
For every DFPTS $\underline{\lambda}\mathbf{S}$:

$$[\Gamma_1, x : A, \Gamma_2 \Vdash M : B \wedge M \in \mathrm{NF}_{\underline{\beta}} \wedge x \notin \mathrm{FV}(\Gamma_2) \cup \mathrm{FV}(M) \cup \mathrm{FV}(B)] \quad \Rightarrow \quad \Gamma_1, \Gamma_2 \Vdash M : B$$

Proof
The proof proceeds by induction on the structure of $M \in \mathrm{NF}_{\underline{\beta}}$. We first prove

$$\left.\begin{array}{c} \Gamma_1, x : A, \Gamma_2 \vdash M : B \\ x \notin \mathrm{FV}(\Gamma_2) \cup \mathrm{FV}(M) \cup \mathrm{FV}(B) \\ M \in \mathrm{NF}_{\underline{\beta}} \end{array}\right\} \quad \Rightarrow \quad [\exists B' \in \underline{\mathscr{E}}.\ B \twoheadrightarrow_{\underline{\beta}} B' \wedge \Gamma_1, \Gamma_2 \Vdash M : B'] \quad (\&)$$

For it, we use the following observation

$$\Delta \Vdash y\ \vec{P} : C \quad \Rightarrow \quad [\exists C', D \in \underline{\mathscr{E}}.\ (y : D) \in \Delta \wedge C \twoheadrightarrow_{\underline{\beta}} C' \wedge \mathrm{FV}(C') \subseteq \mathrm{FV}(D) \cup \mathrm{FV}(\vec{P})]$$

Then we conclude as follows: assume $\Gamma_1, x : A, \Gamma_2 \Vdash M : B$ with $M \in \mathrm{NF}_{\underline{\beta}}$ and $x \notin \mathrm{FV}(\Gamma_2) \cup \mathrm{FV}(M) \cup \mathrm{FV}(B)$. By correctness of types, $\Gamma_1, x : A, \Gamma_2 \Vdash B : s$ and by (&), $\Gamma_1, \Gamma_2 \Vdash B : s$. By (&) again, there exists $B' \in \underline{\mathscr{E}}$ such that $B \twoheadrightarrow_{\underline{\beta}} B'$ and $\Gamma_1, \Gamma_2 \Vdash M : B'$. By conversion, we are done. □

Define $\underline{\eta}_0$-reduction $\rightarrow_{\underline{\eta}_0}$ by

$$\lambda x . M\, x \quad \rightarrow_{\underline{\eta}_0} \quad M \qquad \text{if } M \in \mathrm{NF}_{\underline{\beta}}$$

We have

Proposition 31
If $\Gamma \Vdash^{\underline{\beta\eta}} M : A$ and $M \rightarrow_{\underline{\eta}_0} M'$ then $\Gamma \Vdash^{\underline{\beta\eta}} M' : A$.

Proof
Standard, using Lemma 30. □

This result is enough to ensure *soundness* (Geuvers & Werner, 1994) of normalizing DFPTSs with $\underline{\beta\eta}$-conversion, in the sense that for such systems any two well-typed and convertible expressions are convertible through well-typed terms.

Proposition 32
Assume that $\underline{\lambda}^{\underline{\beta\eta}}\mathbf{S}$ is normalizing. If $\Gamma \Vdash^{\underline{\beta\eta}} M : A$ and $\Gamma \Vdash^{\underline{\beta\eta}} M' : A'$ with $M =_{\underline{\beta\eta}} M'$ then there exists $M_1 \ \dots\ M_n$ such that

$$M \rightarrow_{\underline{\beta\eta}} M_1 \rightarrow_{\underline{\beta\eta}} \ \dots \ \rightarrow_{\underline{\beta\eta}} M_k \ {}_{\underline{\beta\eta}}\!\leftarrow \ \dots \ {}_{\underline{\beta\eta}}\!\leftarrow M_n \ {}_{\underline{\beta\eta}}\!\leftarrow M'$$

with $\Gamma \Vdash^{\underline{\beta\eta}} M_i : A$ for $1 \leqslant i \leqslant k$ and $\Gamma \Vdash^{\underline{\beta\eta}} M_i : A'$ for $k \leqslant i \leqslant n$.

Proof
By confluence of $\underline{\beta\eta}$-reduction, postponement of $\underline{\eta}$-reduction and normalization of $\underline{\lambda}^{\underline{\beta\eta}}\mathbf{S}$, if M and M' are well-typed and $M =_{\beta\eta} M'$ then there exists $M_1 \ \dots M_n$ with $M_j,\ M_l \in \mathrm{NF}_\beta$ and

$$\begin{array}{l} M \rightarrow_\beta M_1 \rightarrow_\beta \ \dots \ \rightarrow_\beta M_j \rightarrow_{\eta 0} \ \dots \ \rightarrow_{\eta 0} M_k \\ M' \rightarrow_\beta M_n \rightarrow_\beta \ \dots \ \rightarrow_\beta M_l \rightarrow_{\eta 0} \ \dots \ \rightarrow_{\eta 0} M_k \end{array}$$

The result then follows immediately from subject reduction for $\underline{\beta}$-reduction and $\underline{\eta}_0$-reduction. □

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press

References

Bakel, S. van, Liquori, L., Ronchi della Rocca, S. and Urzyczyn, P. (1997) Comparing cubes of typed and type assignment systems. *Ann. Pure Appl. Logic*, **86**(3), 267–303.

Barendregt, H. (1984) *The lambda calculus: Its syntax and semantics (revised edn)*. Studies in Logic and the Foundations of Mathematics, vol. 103. North-Holland.

Barendregt, H. (1992) Lambda calculi with types. In: Abramsky, S., Gabbay, D. and Maibaum, T., editors, *Handbook of Logic in Computer Science*, vol. 2, pp. 117–309. Oxford Science Publications.

Barthe, G. (1999) Existence and uniqueness of normal forms in pure type systems with $\beta\eta$-conversion. In: Gottlob, G., Grandjean, E. and Seyr, K., editors, *Proceedings CSL'98: Lecture Notes in Computer Science 1584*, pp. 241–259. Springer-Verlag.

Barthe, G. and Sørensen, M. H. (2000) Domain-free pure type systems. *J. Functional Program.* **10**(5), 417–452.

Coquand, T. (1986) An analysis of Girard's paradox. *Pages 227–236 of: Proceedings of LICS'86*. IEEE Press.

Coquand, T. and Herbelin, H. (1994) A-translation and looping combinators in pure type systems. *J. Functional Program.* **4**(1), 77–88.

Geuvers, H. (1992) The Church-Rosser property for $\beta\eta$-reduction in typed λ-calculi. *Proceedings LICS'92*, pp. 453–460. IEEE Press.

Geuvers, H. (1993) *Logics and type systems*. PhD thesis, University of Nijmegen, The Netherlands.

Geuvers, H. and Nederhof, M. J. (1991) A modular proof of strong normalisation for the Calculus of Constructions. *J. Functional Program.* **1**(2), 155–189.

Geuvers, H. and Werner, B. (1994) On the Church-Rosser property for expressive type systems and its consequence for their metatheoretic study. *Proceedings LICS'94*, pp. 320–329. IEEE Press.

Girard, J.-Y. (1972) Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur. Thèse d'Etat, Université Paris 7, France.

Howe, D. (1987) The computational behaviour of Girard's paradox. *Proceedings LICS'87*, pp. 205–214. IEEE Press.

Hurkens, A. (1995) A Simplification of Girard's Paradox. In: Dezani-Ciancaglini, M. and Plotkin, G., editors, *Proceedings TLCA'95: Lecture Notes in Computer Science 902*, pp. 266–278. Springer-Verlag.

Klop, J. W. (1992) Term-rewriting systems. In: Abramsky, S., Gabbay, D. and Maibaum, T., editors, *Handbook of Logic in Computer Science*, vol. 2, pp. 1–116. Oxford Science Publications.

Martin-Löf, P. (1971) *A theory of types*. Technical Report, Stockholm University, Sweden. February.

Meyer, A. R. and Reinhold, M. B. (1986) "Type" is not a type. *Proceedings POPL'86*, pp. 287–295. ACM Press.

Reinhold, M. B. (1989) *Typechecking is undecidable 'TYPE' is a type*. Technical Report MIT/LCS/TR-458, Massachusetts Institute of Technology.

van Benthem Jutting, L. S. (1993) Typing in pure type systems. *Infor. & Computation*, **105**(1), 30–41.

https://doi.org/10.1017/S0956796803004726 Published online by Cambridge University Press