The Theory of Call-by-Value Solvability

Beniamino Accattoli\textsuperscript{1} \hspace{1cm} Giulio Guerrieri\textsuperscript{2}

\textsuperscript{1}Inria Saclay, France

\textsuperscript{2}Huawei Edinburgh Research Centre, United Kingdom

June 20, 2022
Outline

1 Introduction

2 Call-by-Value $\lambda$-calculus and solvability

3 Our Contribution
Table of Contents

1. Introduction

2. Call-by-Value $\lambda$-calculus and solvability

3. Our Contribution
Meaningful and meaningless in the $\lambda$-calculus

A semantics of the (untyped) $\lambda$-calculus \(\approx\) an equational theory over $\lambda$-terms.

Which equational theories collapse all meaningless $\lambda$-terms?
What does it mean being meaningless?
Meaningful and meaningless in the $\lambda$-calculus

A semantics of the (untyped) $\lambda$-calculus $\approx$ an \textit{equational theory} over $\lambda$-terms.

- It induces some equivalence classes on $\lambda$-terms.
- $\lambda$-terms in the same equivalence class share the same “meaning”.

Which equational theories collapse all meaningless $\lambda$-terms?
What does it mean being meaningless?
Meaningful and meaningless in the $\lambda$-calculus

A semantics of the (untyped) $\lambda$-calculus $\approx$ an equational theory over $\lambda$-terms.

- It induces some equivalence classes on $\lambda$-terms.
- $\lambda$-terms in the same equivalence class share the same “meaning”.

A reasonable approach to give a meaning to $\lambda$-terms:

Which equational theories collapse all meaningless $\lambda$-terms?
What does it mean being meaningless?
Meaningful and meaningless in the $\lambda$-calculus

A semantics of the (untyped) $\lambda$-calculus $\approx$ an equational theory over $\lambda$-terms.
- It induces some equivalence classes on $\lambda$-terms.
- $\lambda$-terms in the same equivalence class share the same “meaning”.

A reasonable approach to give a meaning to $\lambda$-terms:
- Each equivalence class must be stable by $\beta$-conversion;
- There are many different equivalence classes of meaningful $\lambda$-terms;
- **Collapse**: all meaningless $\lambda$-terms should be equated.

Which equational theories collapse all meaningless $\lambda$-terms?
What does it mean being meaningless?
A naive theory: meaningless $\equiv$ non-normalizable

Idea:

1. A $\beta$-normal form is the result of a computation;
2. $\beta$-normalizing $\lambda$-terms are meaningful ($\approx$ defined partial recursive functions);
3. $\beta$-diverging $\lambda$-terms are meaningless ($\approx$ undefined partial recursive functions).

Drawbacks of collapsing all $\beta$-diverging $\lambda$-terms [Bar74,Wad76]:

1. The representation of partial recursive functions is not stable by composition;
2. Inconsistency: the theory equates all $\lambda$-terms! (it collapses everything!)

Moral:

1. Being $\beta$-normalizable is not a meaningful predicate.
2. $\beta$-normalizing terms are not the only meaningful $\lambda$-terms.
A naive theory: meaningless \(\equiv\) non-normalizable

Idea:

1. A \(\beta\)-normal form is the result of a computation;
2. \(\beta\)-normalizing \(\lambda\)-terms are meaningful (\(\approx\) defined partial recursive functions);
3. \(\beta\)-diverging \(\lambda\)-terms are meaningless (\(\approx\) undefined partial recursive functions).

Drawbacks of collapsing all \(\beta\)-diverging \(\lambda\)-terms [Bar74, Wad76]:

1. The representation of partial recursive functions is not stable by composition;
2. Inconsistency: the theory equates all \(\lambda\)-terms! (it collapses everything!)
A naive theory: meaningless $\equiv$ non-normalizable

Idea:
1. A $\beta$-normal form is the result of a computation;
2. $\beta$-normalizing $\lambda$-terms are meaningful ($\approx$ defined partial recursive functions);
3. $\beta$-diverging $\lambda$-terms are meaningless ($\approx$ undefined partial recursive functions).

Drawbacks of collapsing all $\beta$-diverging $\lambda$-terms [Bar74,Wad76]:
1. The representation of partial recursive functions is not stable by composition;
2. Inconsistency: the theory equates all $\lambda$-terms! (it collapses everything!)

Moral:
1. Being $\beta$-normalizable is not a meaningful predicate.
2. $\beta$-normalizing terms are not the only meaningful $\lambda$-terms.
A sensible theory: meaningless = unsolvable

**Definition:** A λ-term $t$ is solvable if there is a head context $H$ sending $H\langle t \rangle$ to the identity $I = \lambda x.x$, that is, such that $H\langle t \rangle \rightarrow^* \beta I$.

**Idea:** A solvable term $t$ might be divergent but all its diverging sub-terms are removable without discarding the whole $t$.

Example: Let $\delta = \lambda z.z$. Then, $\Omega = \delta \delta$ is unsolvable. But $x\Omega$ is solvable!

Let $H = (\lambda x.\langle \cdot \rangle)\lambda y.I$: $H\langle x\Omega \rangle = (\lambda x.\Omega)\lambda y.I \rightarrow^* \beta I$. 

Theorem [Bar74]: Collapsing all unsolvable terms is consistent (sensible theories).

Examples: $H$, theories induced by models (Scott’s $D\infty$, relational semantics, etc.).
Definition: A $\lambda$-term $t$ is solvable if there is a head context $H$ sending $H\langle t \rangle$ to the identity $I = \lambda x. x$, that is, such that $H\langle t \rangle \rightarrow^*_\beta I$.

Idea: A solvable term $t$ might be divergent but all its diverging sub-terms are removable without discarding the whole $t$.

Example: Let $\delta = \lambda z.zz$. Then, $\Omega = \delta\delta$ is unsolvable. But $x\Omega$ is solvable!
Let $H = (\lambda x.\langle \cdot \rangle)\lambda y. I$: $H\langle x\Omega \rangle = (\lambda x.x\Omega)\lambda y. I \rightarrow^*_\beta (\lambda y.I)\Omega \rightarrow^*_\beta I$. 

Theorem [Bar74]: Collapsing all unsolvable terms is consistent (sensible theories).
Examples: $H\langle \cdot \rangle$, theories induced by models (Scott's $D_\infty$, relational semantics, etc.).
A sensible theory: meaningless $\equiv$ unsolvable

Definition: A $\lambda$-term $t$ is solvable if there is a head context $H$ sending $H\langle t \rangle$ to the identity $I = \lambda x.x$, that is, such that $H\langle t \rangle \rightarrow^* I$.

Idea: A solvable term $t$ might be divergent but all its diverging sub-terms are removable without discarding the whole $t$.

Example: Let $\delta = \lambda z.zz$. Then, $\Omega = \delta\delta$ is unsolvable. But $x\Omega$ is solvable! Let $H = (\lambda x.\langle \cdot \rangle)\lambda y.I$: $H\langle x\Omega \rangle = (\lambda x.x\Omega)\lambda y.I \rightarrow_\beta (\lambda y.I)\Omega \rightarrow_\beta I$.

Theorem [Bar74]: Collapsing all unsolvable terms is consistent (sensible theories).

Examples: $\mathcal{H}$, theories induced by models (Scott’s $D_\infty$, relational semantics, etc.).
Characterizations of solvability: a beautiful theory

Definition of solvability is not handy (how to find an head context?)
Characterizations of solvability: a beautiful theory

Definition of solvability is not handy (how to find an head context?)
Characterizations of solvability: a beautiful theory

Definition of solvability is not handy (how to find an head context?)

**Theorem [Operational characterization, Bar74]:** $t$ is solvable iff head reduction terminates on $t$.

**Corollary:** The class of solvable terms strictly includes the $\beta$-normalizing ones. Morally, unsolvable means “heavily divergent”.

Characterizations of solvability: a beautiful theory

Definition of solvability is not handy (how to find an head context?)

**Theorem** [Operational characterization, Bar74]: \( t \) is solvable iff head reduction terminates on \( t \).

**Corollary**: The class of solvable terms strictly includes the \( \beta \)-normalizing ones. Morally, unsolvable means “heavily divergent”.

**Theorem** [Type-theoretic characterizations, [CopDez80, deC07]]: \( t \) is solvable iff \( t \) is typable in a (idempotent or non-idempotent) intersection type system.
Characterizations of solvability: a beautiful theory

Definition of solvability is not handy (how to find an head context?)

Theorem [Operational characterization, Bar74]: \( t \) is solvable iff head reduction terminates on \( t \).

Corollary: The class of solvable terms strictly includes the \( \beta \)-normalizing ones. Morally, unsolvable means “heavily divergent”.

Theorem [Type-theoretic characterizations, [CopDez80,deC07]: \( t \) is solvable iff \( t \) is typable in a (idempotent or non-idemptotent) intersection type system.

Theorem [Genericity, Bar84]: Let \( t \) be unsolvable, \( u \) be \( \beta \)-normal, \( C \) be a context. If \( C\langle t \rangle \rightarrow^*_\beta u \) then \( C\langle s \rangle \rightarrow^*_\beta u \) for every term \( s \).

Idea: \( C\langle t \rangle \) normalizes and has a unsolvable subterm \( t \), so \( t \) is discarded.
1 Introduction

2 Call-by-Value λ-calculus and solvability

3 Our Contribution
Plotkin’s Call-by-Value $\lambda$-calculus [Plo75]

Terms $s, t, u ::= v \mid tu$

Values $v ::= x \mid \lambda x.t$

CbV reduction $(\lambda x.t)v \rightarrow_{\beta_v} t[v/x]$

It is closer to real implementation of most programming languages.
The semantics is completely different from standard (Call-by-Name) $\lambda$-calculus.
Plotkin’s Call-by-Value $\lambda$-calculus [Plo75]

Terms $s, t, u ::= v \mid tu$

Values $v ::= x \mid \lambda x.t$

CbV reduction $(\lambda x.t)v \rightarrow_{\beta_v} t\{v/x\}$

It is closer to real implementation of most programming languages. The semantics is completely different from standard (Call-by-Name) $\lambda$-calculus.
Plotkin’s Call-by-Value $\lambda$-calculus [Plo75]

Terms $s, t, u ::= v \mid tu$

Values $v ::= x \mid \lambda x. t$

CbV reduction $(\lambda x. t)v \rightarrow_{\beta_v} t\{v/x\}$

It is closer to real implementation of most programming languages. The semantics is completely different from standard (Call-by-Name) $\lambda$-calculus.

Examples:

1. $(\lambda x. \delta)(xx)\delta$ is $\beta_v$-normal but $\beta$-divergent!
Definition: A head context is a context defined by $H ::= \langle \cdot \rangle \mid \lambda x.H \mid Ht$.

A $\lambda$-term $t$ is $\beta_v$-solvable if there is a head context $H$ sending $H\langle t \rangle$ to the identity $I = \lambda x.x$, that is, such that $H\langle t \rangle \rightarrow_{\beta_v}^* I$. 

Examples:
1. $\Omega$ is $\beta_v$-unsolvable, because $\Omega$ cannot be erased (but it is $\beta$-solvable).
2. $(\lambda x.\delta)(xx)\delta$ is $\beta_v$-normal but $\beta_v$-unsolvable.
3. No operational characterization of $\beta_v$-solvability inside Plotkin’s calculus! What a mess!
Call-by-Value solvability

Definition: A head context is a context defined by $H ::= \langle \cdot \rangle \mid \lambda x. H \mid Ht$.

A $\lambda$-term $t$ is $\beta_v$-solvable if there is a head context $H$ sending $H\langle t \rangle$ to the identity $I = \lambda x.x$, that is, such that $H\langle t \rangle \rightarrow_{\beta_v}^* I$.

Examples:

1. $x\Omega$ is $\beta_v$-unsolvable, because $\Omega$ cannot be erased (but it is $\beta$-solvable).
2. $(\lambda x.\delta)(xx)\delta$ is $\beta_v$-normal but $\beta_v$-unsolvable.
3. No operational characterization of $\beta_v$-solvability inside Plotkin’s calculus!
Call-by-Value solvability

Definition: A head context is a context defined by $H ::= \langle \cdot \rangle \mid \lambda x.H \mid Ht$.

A λ-term $t$ is $\beta_v$-solvable if there is a head context $H$ sending $H\langle t \rangle$ to the identity $I = \lambda x.x$, that is, such that $H\langle t \rangle \rightarrow_{\beta_v}^* I$.

Examples:

1. $x\Omega$ is $\beta_v$-unsolvable, because $\Omega$ cannot be erased (but it is $\beta$-solvable).
2. $(\lambda x.\delta)(xx)\delta$ is $\beta_v$-normal but $\beta_v$-unsolvable.
3. No operational characterization of $\beta_v$-solvability inside Plotkin’s calculus!

What a mess!
Alternative CbV $\lambda$-calculus: Value Substitution [AccPao12]

Terms: $s, t, u ::= v \mid tu \mid t[u/x]$  
Values: $v ::= x \mid \lambda x.t$

Substitution contexts: $L ::= [t_1/x_1] \ldots [t_n/x_n]$

Reductions:  
$(\lambda x.t)Lu \rightarrow_m t[u/x]L$  
$t[vL/x] \rightarrow_e t[v/x]L$
Alternative CbV $\lambda$-calculus: Value Substitution [AccPao12]

Terms: $s, t, u ::= \nu \mid tu \mid t[u/x]$  
Values: $\nu ::= x \mid \lambda x. t$

Substitution contexts: $L ::= \lfloor t_1/x_1 \rfloor \ldots \lfloor t_n/x_n \rfloor$

Reductions:
1. $\beta_\nu$-reduction can be simulated into VSC.

   $$(\lambda x. t) \nu \rightarrow_m t[v/x] \rightarrow_e t\{v/x\} \quad L$$

2. VSC extends $\beta_\nu$-reduction:

   $$(\lambda x. \delta)(xx)\delta \rightarrow_m \delta[xx/x]\delta \rightarrow_m (zz)[\delta/z][xx/x] \rightarrow_e \delta\delta[xx/x] \rightarrow \cdots$$
Operational internal characterization of VSC-Solvability

Theorem [AccPao12]: \( t \) is VSC-solvable iff solving reduction terminates on \( t \).

Solving reduction: restriction of VSC not firing under \( \lambda \) on the left of application.

Corollary: The set of VSC-scrutable terms strictly includes the VSC-solvable ones.
Theorem [AccPao12]: \( t \) is VSC-solvable iff solving reduction terminates on \( t \).

**Solving reduction:** restriction of VSC not firing under \( \lambda \) on the left of application.

Theorem [AccPao12]: \( t \) is VSC-scrutable iff weak reduction terminates on \( t \).

**Weak reduction:** restriction of VSC not firing under \( \lambda \).

**Scrutability:** \( t \) is VSC-scrutable (aka VSC-potentially valuable) if there are values \( v, v_1, \ldots, v_n \) such that \( t\{v_1/x_1, \ldots, v_n/x_n\} \rightarrow^*_\text{VSC} v \).

**Corollary:** The set of VSC-scrutable terms strictly includes the VSC-solvable ones.
# Table of Contents

1. Introduction

2. Call-by-Value $\lambda$-calculus and solvability

3. Our Contribution
Some results

Theorem [Robustness]:

1. $t$ is VSC-scrutable iff $t$ is $\beta_v$-scrutable.
2. $t$ is VSC-solvable iff $t$ is $\beta_v$-solvable.

The notions are robust in CbV, do not depend on the (CbV) calculus. VSC is a tool to study them!
Some results

Theorem [Robustness]:
1. $t$ is VSC-scrutable iff $t$ is $\beta_v$-scrutable.
2. $t$ is VSC-solvable iff $t$ is $\beta_v$-solvable.

The notions are robust in CbV, do not depend on the (CbV) calculus. VSC is a tool to study them!

Theorem [(Un-)Collapsibility]:
1. Collapsing all CbV-unsolvable terms is inconsistent.
2. Collapsing all CbV-inscrutable terms is consistent.

In CbV, meaningless = unscrutable. In CbN, meaningless = unsolvable.
Some results

**Theorem [Robustness]:**
1. $t$ is VSC-scrutable iff $t$ is $\beta_v$-scrutable.
2. $t$ is VSC-solvable iff $t$ is $\beta_v$-solvable.

The notions are robust in CbV, do not depend on the (CbV) calculus. VSC is a tool to study them!

**Theorem [(Un-)Collapsibility]:**
1. Collapsing all CbV-unsolvable terms is inconsistent.
2. Collapsing all CbV-inscrutable terms is consistent.

In CbV, meaningless = unscrutable. In CbN, meaningless = unsolvable.

**Theorem [Non Genericity]:** Genericity does not hold with CbV solvability.

**Conjecture [Genericity]:** Genericity does hold with CbV scrutability.
Type-theoretic characterization of solvability/scrutability

Theorem

1. $t$ is CbV-scrutable iff $t$ is typable in a (suitable) non-idempotent intersection type system.

2. $t$ is CbV-solvable iff $t$ is typable in a (suitable) restriction of the non-idempotent intersection type system.