The theory of meaningfulness in the call-by-name and call-by-value λ-calculi

a long journey with Simona Ronchi Della Rocca, Luca Paolini, Beniamino Accattoli, Delia Kesner, Victor Arrial, and many others

Giulio Guerrieri

LIS, Aix-Marseille Université, France

Workshop in honour of Antonio Bucciarelli's 60th birthday
Paris, June 20, 2023

Outline

Table of Contents

Looking for a meaning

In every language there are sentences that are

- syntactically well-formed, but
- semantically nonsensical.

Example: "Colorless green ideas sleep furiously" (Chomsky).

Looking for a meaning

In every language there are sentences that are

- syntactically well-formed, but
- semantically nonsensical.

Example: "Colorless green ideas sleep furiously" (Chomsky).

Question: How to distinguish meaningful and meaningless sentences?

Looking for a meaning

In every language there are sentences that are

- syntactically well-formed, but
- semantically nonsensical.

Example: "Colorless green ideas sleep furiously" (Chomsky).

Question: How to distinguish meaningful and meaningless sentences?

We address this question when the language is the untyped pure λ-calculus.

Meaningful and meaningless in the λ-calculus

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

Meaningful and meaningless in the λ-calculus

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

- It induces some equivalence classes on λ-terms.
- λ-terms in the same equivalence class share the same "meaning".

Meaningful and meaningless in the λ-calculus

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

- It induces some equivalence classes on λ-terms.
- λ-terms in the same equivalence class share the same "meaning".

A reasonable approach to give a meaning to λ-terms:

Meaningful and meaningless in the λ-calculus

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

- It induces some equivalence classes on λ-terms.
- λ-terms in the same equivalence class share the same "meaning".

A reasonable approach to give a meaning to λ-terms:

- Each equivalence class must be stable by β-conversion and contextual closure;
- There are many different equivalence classes of meaningful λ-terms;

Meaningful and meaningless in the λ-calculus

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

- It induces some equivalence classes on λ-terms.
- λ-terms in the same equivalence class share the same "meaning".

A reasonable approach to give a meaning to λ-terms:

- Each equivalence class must be stable by β-conversion and contextual closure;
- There are many different equivalence classes of meaningful λ-terms;
- Collapse: all meaningless λ-terms should be equated;
- Genericity: meaningless subterms are irrelevant in the evaluation of normalizing terms.

Meaningful and meaningless in the λ-calculus

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

- It induces some equivalence classes on λ-terms.
- λ-terms in the same equivalence class share the same "meaning".

A reasonable approach to give a meaning to λ-terms:

- Each equivalence class must be stable by β-conversion and contextual closure;
- There are many different equivalence classes of meaningful λ-terms;
- Collapse: all meaningless λ-terms should be equated;
- Genericity: meaningless subterms are irrelevant in the evaluation of normalizing terms.

Crucial questions in this line of research:

1. Which equational theories collapse all meaningless λ-terms?
2. Which (sub)terms are generic (irrelevant in evaluating normalizing terms)?
3. What does it mean being meaningless (and so, meaningful)?

A naive theory: meaningless = non-normalizable

Idea (naive):

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

A naive theory: meaningless = non-normalizable

Idea (naive):

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

Drawbacks of the theory $\mathcal{T}_{\mathrm{nf}}$ collapsing all β-diverging λ-terms [Bare'74,Wads'76]:

1. The representation of partial recursive functions is not stable by composition;
2. Inconsistency: the theory $\mathcal{T}_{\mathrm{nf}}$ equates all λ-terms! (it collapses everything!) Indeed, for every terms t and s, we have $\lambda x.xt\Omega =_{\mathcal{T}_{\mathrm{nf}}} \lambda x.xs\Omega$, and so

$$t =_\beta (\lambda x.xt\Omega)(\lambda z.\lambda y.z) =_{\mathcal{T}_{\mathrm{nf}}} (\lambda x.xs\Omega)(\lambda z.\lambda y.z) =_\beta s$$

A naive theory: meaningless = non-normalizable

Idea (naive):

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

Drawbacks of the theory $\mathcal{T}_{\mathrm{nf}}$ collapsing all β-diverging λ-terms [Bare'74,Wads'76]:

1. The representation of partial recursive functions is not stable by composition;
2. Inconsistency: the theory $\mathcal{T}_{\mathrm{nf}}$ equates all λ-terms! (it collapses everything!) Indeed, for every terms t and s, we have $\lambda x.xt\Omega =_{\mathcal{T}_{\mathrm{nf}}} \lambda x.xs\Omega$, and so

$$t =_\beta (\lambda x.xt\Omega)(\lambda z.\lambda y.z) =_{\mathcal{T}_{\mathrm{nf}}} (\lambda x.xs\Omega)(\lambda z.\lambda y.z) =_\beta s$$

Moral:

1. Being β-normalizable is not a meaningful predicate.
2. β-normalizing terms are not the only meaningful λ-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 $H\langle t\rangle \to_\beta^* I$, where

$$\text{head contexts} \quad H ::= \langle\cdot\rangle \mid \lambda x.H \mid Ht$$

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

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 $H\langle t\rangle \to_\beta^* I$, where

$$\text{head contexts} \quad H ::= \langle\cdot\rangle \mid \lambda x.H \mid Ht$$

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 $H = (\lambda x.\langle\cdot\rangle)\lambda y.I$: then $H\langle x\Omega\rangle = (\lambda x.x\Omega)\lambda y.I \to_\beta (\lambda y.I)\Omega \to_\beta I$.

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 $H\langle t\rangle \to_\beta^* I$, where

$$\text{head contexts} \quad H ::= \langle\cdot\rangle \mid \lambda x.H \mid Ht$$

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 $H = (\lambda x.\langle\cdot\rangle)\lambda y.I$: then $H\langle x\Omega\rangle = (\lambda x.x\Omega)\lambda y.I \to_\beta (\lambda y.I)\Omega \to_\beta I$.

Example: $\lambda x.xt\Omega$ and $\lambda x.xs\Omega$ are solvable (and so t and s are not collapsed)!
Take $H = \langle\cdot\rangle\lambda y.\lambda z.I$

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 $H\langle t\rangle \to_\beta^* I$, where

$$\text{head contexts} \quad H ::= \langle\cdot\rangle \mid \lambda x.H \mid Ht$$

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 $H = (\lambda x.\langle\cdot\rangle)\lambda y.I$: then $H\langle x\Omega\rangle = (\lambda x.x\Omega)\lambda y.I \to_\beta (\lambda y.I)\Omega \to_\beta I$.

Example: $\lambda x.xt\Omega$ and $\lambda x.xs\Omega$ are solvable (and so t and s are not collapsed)!
Take $H = \langle\cdot\rangle\lambda y.\lambda z.I$

Theorem [Bar'74]: Collapsing all unsolvable terms is consistent (sensible theories).

Examples: $\mathcal{H}$, theories induced by models (Scott's D_∞, relational semantics, etc.).

Characterizations of solvability: a beautiful theory 1/2

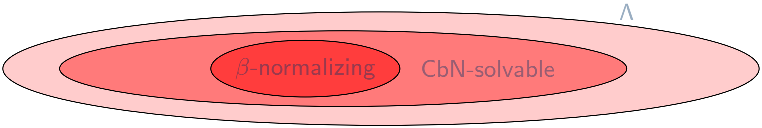
Definition of solvability is not handy (how to find an head context?)

Characterizations of solvability: a beautiful theory 1/2

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

Theorem (Operational characterization) [Bare'74]: t is solvable iff head reduction terminates on t.

Corollary: The class of solvable terms strictly includes the β-normalizing ones. Morally, unsolvable means "heavily divergent".



Characterizations of solvability: a beautiful theory 2/2

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

Characterizations of solvability: a beautiful theory 2/2

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 β-normal, C be a context. If $C\langle t\rangle \to_\beta^* u$ then $C\langle s\rangle \to_\beta^* u$ for every term s.

Idea: $C\langle t\rangle$ normalizes and has an unsolvable subterm t, so t is discarded.
$\rightsquigarrow$ Unsolvable subterms are irrelevant in the evaluation of normalizing terms.

Table of Contents

Plotkin's Call-by-Value λ-calculus [Plo75]

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

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

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

Plotkin's Call-by-Value λ-calculus [Plo75]

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

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

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

Examples (with duplicator $\delta = \lambda z.zz$ and identity $I = \lambda z.z$):

1. $\Omega = \delta\delta \to_{\beta_v} \delta\delta \to_{\beta_v} \delta\delta \to_{\beta_v} \dots$

Plotkin's Call-by-Value λ-calculus [Plo75]

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

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

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

Examples (with duplicator $\delta = \lambda z.zz$ and identity $I = \lambda z.z$):

1. $\Omega = \delta\delta \to_{\beta_v} \delta\delta \to_{\beta_v} \delta\delta \to_{\beta_v} \dots$
2. $\delta(\delta I) \to_{\beta_v} \delta(II) \to_{\beta_v} \delta I \to_{\beta_v} II \to_{\beta_v} I$ but $\delta(\delta I) \not\to_{\beta_v} (\delta I)(\delta I)$.
3. $(\lambda x.\delta)(xx)\delta$ is β_v-normal but β-divergent!
4. $(\lambda x.I)\Omega$ is β_v-divergent but β-normalizing!

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 β_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 $H\langle t\rangle \to^*_{\beta_v} I$.

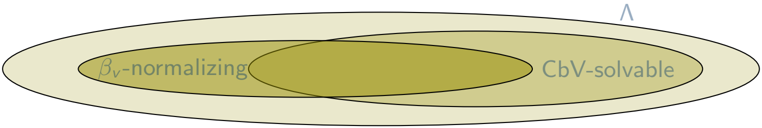
Call-by-Value solvability

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

A λ-term t is β_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 $H\langle t\rangle \to^*_{\beta_v} I$.

Examples:

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



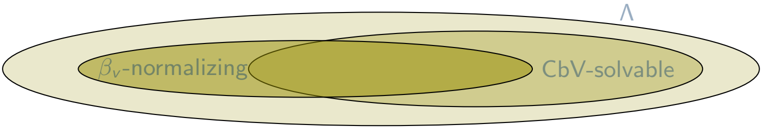
Call-by-Value solvability

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

A λ-term t is β_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 $H\langle t\rangle \to^*_{\beta_v} I$.

Examples:

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



What a mess!

Alternative CbV λ-calculus: Value Substitution [AccPao12]

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

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

Reductions: $(\lambda x.t)Ls \to_m t[s/x]L$ $t[vL/x] \to_e t\{v/x\}L$

Alternative CbV λ-calculus: Value Substitution [AccPao12]

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

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

Reductions: $(\lambda x.t)Ls \to_m t[s/x]L$ $t[vL/x] \to_e t\{v/x\}L$

1. β_v-reduction can be simulated in the Value Substitution Calculus (VSC).

$$(\lambda x.t)v \to_m t[v/x] \to_e t\{v/x\}$$

2. VSC extends β_v-reduction: $(\lambda x.\delta)(xx)\delta$ is β_v-normal but

$$(\lambda x.\delta)(xx)\delta \to_m \delta[xx/x]\delta \to_m (zz)[\delta/z][xx/x] \to_e \delta\delta[xx/x] \to \cdots$$

Operational internal characterization of VSC-Solvability

Theorem [AccattPaol12]: t is VSC-solvable iff solving reduction terminates on t.

Solving reduction: restriction of VSC not firing under λ on the right of application.

Operational internal characterization of VSC-Solvability

Theorem [AccattPaol12]: t is VSC-solvable iff solving reduction terminates on t.

Solving reduction: restriction of VSC not firing under λ on the right of application.

Theorem [AccattPaol12]: t is VSC-scrutable iff weak reduction terminates on t.

Weak reduction: restriction of VSC not firing under λ.
Scrutability: t is VSC-scrutable (aka VSC-potentially valuable) if there are values $v, v_1, \ldots, v_n$ such that $t\{v_1/x_1, \ldots, v_n/x_n\} \to^*_{VSC} v$.

Operational internal characterization of VSC-Solvability

Theorem [AccattPaol12]: t is VSC-solvable iff solving reduction terminates on t.

Solving reduction: restriction of VSC not firing under λ on the right of application.

Theorem [AccattPaol12]: t is VSC-scrutable iff weak reduction terminates on t.

Weak reduction: restriction of VSC not firing under λ.
Scrutability: t is VSC-scrutable (aka VSC-potentially valuable) if there are values $v, v_1, \ldots, v_n$ such that $t\{v_1/x_1, \ldots, v_n/x_n\} \rightarrow^*_{VSC} v$.

Corollary: The set of VSC-scrutable terms strictly includes the VSC-solvable ones.

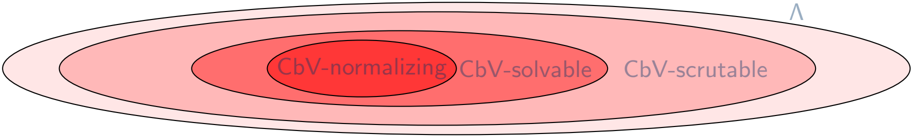


Table of Contents

Open questions we answer

Much was still not well understood about CbV meaningfulness before our papers.

1. There are characterizations via types, but they all contain mistakes.
2. Is CbV solvability a meaning predicate? If not, then what?
   - Can we collapse CbV unsolvable terms? If not, what can we collapse?
   - Are CbV unsolvable terms generic? If not, what is generic in CbV?
3. Is CbV solvability in Plotkin's calculus the same as in the VSC?
4. ...

Some results: robustness

Theorem (Robustness) [GuerrPaolRonchi'17,AccattGuerr'22]:

1. t is VSC-scrutable iff t is β_v-scrutable.
2. t is VSC-solvable iff t is β_v-solvable.

Some results: robustness

Theorem (Robustness) [GuerrPaolRonchi'17,AccattGuerr'22]:

1. t is VSC-scrutable iff t is β_v-scrutable.
2. t is VSC-solvable iff t is β_v-solvable.

The notions are robust in CbV, do not depend on the (CbV) calculus.

- VSC is a tool to study them! (Not the only one, other–equivalent–extensions)
- Change syntax (not semantics) of CbV to have good operational properties.

Some results: collapse

Theorem ((Un-)Collapsibility) [AccattGuerr'22,ArrialKesnerGuerr'??]:

Some results: collapse

Theorem ((Un-)Collapsibility) [AccattGuerr'22,ArrialKesnerGuerr'??]:

1. Collapsing all CbV-unsolvable terms is inconsistent.
2. Collapsing all CbV-inscrutable terms is consistent.

Moral: In CbV, meaningless = inscrutable. In CbN, meaningless = unsolvable.

Some results: collapse

Theorem ((Un-)Collapsibility) [AccattGuerr'22,ArrialKesnerGuerr'??]:

1. Collapsing all CbV-unsolvable terms is inconsistent.
2. Collapsing all CbV-inscrutable terms is consistent.

Moral: In CbV, meaningless = inscrutable. In CbN, meaningless = unsolvable.

Actually, we have some analogue consistent equational theories in CbN and CbV:

	CbN	CbV
$\mathcal{H}$	collapsing CbN unsolvable terms [Bare'74,Wads'76]	collapsing CbV inscrutable terms [ArrialKesnerGuerr'??]
$\mathcal{H}^*$	the only maximal consistent extension of CbN $\mathcal{H}$ [Bare'84]	the only maximal consistent extension of CbV $\mathcal{H}$ [ArrialKesnerGuerr'??]

Some results: genericity

Theorem (Non Genericity in CbV) [AccattGuerr'22]:
Genericity does not hold with CbV (un)solvability.

Some results: genericity

Theorem (Non Genericity in CbV) [AccattGuerr'22]:
Genericity does not hold with CbV (un)solvability.

Theorem (Genericity in CbV) [ArrialKesnerGuerr'??]:
Genericity does hold with CbV (in)scrutability.

Some results: genericity

Theorem (Non Genericity in CbV) [AccattGuerr'22]:
Genericity does not hold with CbV (un)solvability.

Theorem (Genericity in CbV) [ArrialKesnerGuerr'??]:
Genericity does hold with CbV (in)scrutability.

Genericity (in CbN and CbV) is a non-trivial result.

There are well-known techniques to prove genericity in CbN:

1. a very sophisticated proof by means of a topological approach [Bare'84];
2. a simpler proof by based on an operational approach [Takah'94];
3. a proof based on the powerful notion of Taylor expansion [BarbaManzo'20].

They do not work in CbV, or it is not easy to adapt them to CbV.

Some results: genericity

Theorem (Non Genericity in CbV) [AccattGuerr'22]:
Genericity does not hold with CbV (un)solvability.

Theorem (Genericity in CbV) [ArrialKesnerGuerr'??]:
Genericity does hold with CbV (in)scrutability.

Genericity (in CbN and CbV) is a non-trivial result.

There are well-known techniques to prove genericity in CbN:

1. a very sophisticated proof by means of a topological approach [Bare'84];
2. a simpler proof by based on an operational approach [Takah'94];
3. a proof based on the powerful notion of Taylor expansion [BarbaManzo'20].

They do not work in CbV, or it is not easy to adapt them to CbV.

Our proof is based on a calculus of approximants, it works for both CbN and CbV.

Type-theoretic characterization of solvability/scrutability

Theorem [AccattGuerr'22,ArrialKesnerGuerr'??]

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.

Type-theoretic characterization of solvability/scrutability

Theorem [AccattGuerr'22,ArrialKesnerGuerr'??]

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.

Both results are refined as to be quantitative:
$\rightsquigarrow$ Type derivations give the time cost of the weak/solving strategy.

Still open questions about CbV

There are all the ingredients for a theory for CbV λ-calculus as elegant as for CbN.

But there still are at least two open (and challenging) questions:

Still open questions about CbV

There are all the ingredients for a theory for CbV λ-calculus as elegant as for CbN.

But there still are at least two open (and challenging) questions:

1. What is a denotational model of the CbV λ-calculus?
   What is the CbV equivalent of a reflexive object in a CCC?

Still open questions about CbV

There are all the ingredients for a theory for CbV λ-calculus as elegant as for CbN.

But there still are at least two open (and challenging) questions:

1. What is a denotational model of the CbV λ-calculus?
   What is the CbV equivalent of a reflexive object in a CCC?
2. Meaningfulness seems to have a different definition in CbN and CbV.
   Is there a general framework where meaningfulness can be defined uniformly?
   1. In the CbN fragment of that general framework, meaningful = solvable;
   2. In the CbV fragment of that general framework, meaningful = scrutable.

Thank you!

Buon compleanno, Antonio!