Intersection Types as Evaluation Types

Pablo Barenbaum[1]   Eduardo Bonelli[2]   Mariana Milicich[3*†]

[1]UNQ (CONICET) and ICC, UBA, Argentina

[2]Stevens Institute of Technology, USA

[3]Université Paris-Cité, CNRS, IRIF, France

Workshop on Intersection Types and Related Systems
09/07/2024

* *This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 945332.*

† *Work by this author was partially supported by ICC, UBA, Argentina.*

Motivation

Theorem (Intersection types characterise termination)

Type systems $\quad P \text{ types} \iff P \text{ terminates} \quad$ **Programming languages**

Motivation

Theorem (Intersection types characterise termination)

Type systems P *types* $\iff$ P *terminates* **Programming languages**

Is it possible to
use intersection types as a **notation**
for normalisation sequences?

Motivation

Theorem (Intersection types characterise termination)

Type systems $\quad P$ *types* $\iff P$ *terminates* $\quad$ **Programming languages**

given a normal form N,
typing with $N \iff$ terminating in N

Our Contributions

For both (weak) **call-by-name** and (weak closed) **call-by-value**:

- Non-idempotent intersection type systems where **terms are typed with their normal form**: **Evaluation types**
- **One-to-one correspondence** between normalisation sequences and typing derivations.

Our Contributions

For both (weak) **call-by-name** and (weak closed) **call-by-value**:

- Non-idempotent intersection type systems where **terms are typed with their normal form**: **Evaluation types**
- **One-to-one correspondence** between normalisation sequences and typing derivations.

Related work

A. Bernadet and S. Graham-Lengrand proposed an non-idempotent intersection type system for **strong call-by-name** (2013).

- Typing with the *structure* of the normal form.
- Focus on the quantitative aspects of the type system.

Weak Call-by-Name

Syntax

$$\begin{array}{rrcll} \text{(Terms)} & t,s & ::= & x \mid \lambda x.t \mid t\,s & \\ \text{(Answers)} & \mathtt{a} & ::= & \lambda x.t \mid x\,t_1 \ldots t_n & (n \geq 0) \end{array}$$

Some answers:

$\mathtt{x}$ $\mathtt{id}$ $x\,y\,\mathtt{id}$

Weak Call-by-Name

Syntax

$$
\begin{array}{rrcll}
\text{(Terms)} & t, s & ::= & x \mid \lambda x.t \mid t\,s & \\
\text{(Answers)} & \mathtt{a} & ::= & \lambda x.t \mid x\,t_1 \ldots t_n & (n \geq 0)
\end{array}
$$

Some answers:

$\mathtt{x}$ $\mathtt{id}$ $x\,y\,\mathtt{id}$

Operational Semantics

$$
\frac{}{(\lambda x.t)\,s \to t\{x := s\}} \qquad \frac{t \to t'}{t\,s \to t'\,s}
$$

Evaluation Types for Call-by-Name

$$A ::= \mathtt{a} \mid t.M \to A$$
$$M ::= [A_1, \ldots, A_n] \qquad (n \geq 0)$$

- **Normal forms** and β-**redexes** are typed with the **term** $\mathtt{a}$.
  - $x : x$ if x is **free**, but otherwise $x : \mathtt{a}$, $\mathtt{a} \neq x$
  - $(\lambda x.x\,y)\,z : z\,y$
  - $x\,y\,\mathtt{id} : x\,y\,\mathtt{id}$
- Left subterms of a **beta-redex** are typed with $t.M \to A$.
  - Typing $\lambda x.x\,x\,z$ in $(\lambda x.x\,x\,z)\,y$:

$$\lambda x.x\,x\,z : y.[y, y] \to y\,y\,z$$

Evaluation Types for Call-by-Name

Typing judgements:

$$\sigma \,;\Gamma \vdash t \Downarrow A$$

- σ: lists of substitutions $(x_1 : t_1, \ldots, x_n : t_n)$
- Γ: function mapping variables to multisets of types

Evaluation Types for Call-by-Name

Typing judgements:

$$\sigma\,;\Gamma \vdash t \Downarrow A$$

- σ: lists of substitutions $(x_1 : t_1, \ldots, x_n : t_n)$
- Γ: function mapping variables to multisets of types

Two typing rules for each term in the syntax to distinguish:

- **Variables** bound by abstractions from those that are not.
- **Abstractions** to the left of a β-redex from those that are not.
- β-redexes from **applications** whose head term is a variable.

Evaluation Types for Call-by-Name

Typing Rules

$$\frac{x \notin \mathrm{dom}(\sigma)}{\sigma\,;\varnothing \vdash x \Downarrow x} \qquad \frac{x \in \mathrm{dom}(\sigma) \qquad \mathrm{fv}(A) \# \mathrm{dom}(\sigma)}{\sigma\,;x:[A] \vdash x \Downarrow A}$$

$$\frac{}{\sigma\,;\varnothing \vdash \lambda x.\,t \Downarrow \lambda x.\,t^{\sigma}} \qquad \frac{\sigma, x : s\,; \Gamma, x : M \vdash t \Downarrow A}{\sigma\,;\Gamma \vdash \lambda x.\,t \Downarrow s.M \to A}$$

$$\frac{\sigma\,;\Gamma \vdash t \Downarrow x\, t_1 \ldots t_n}{\sigma\,;\Gamma \vdash t\, s \Downarrow x\, t_1 \ldots t_n\, s^{\sigma}}$$

$$\frac{\sigma\,;\Gamma \vdash t \Downarrow s^{\sigma}.[A_1, \ldots, A_n] \to B \quad (\sigma\,;\Delta_i \vdash s \Downarrow A_i)_{i=1}^{n}}{\sigma\,;\Gamma +_{i=1}^{n} \Delta_i \vdash t\, s \Downarrow B}$$

Evaluation Types for Call-by-Name

Example

Let $\mathtt{id} := \lambda x_1 . x_1$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : \mathtt{id}\, z ; x : [z] \vdash x \Downarrow z}}{x : \mathtt{id}\, z ; x : [z] \vdash x\, y \Downarrow z\, y}}{\varnothing ; \varnothing \vdash \lambda x . x\, y \Downarrow (\mathtt{id}\, z).[z] \to z\, y} \qquad \dfrac{\dfrac{\dfrac{}{x_1 : z ; x_1 : [z] \vdash x_1 \Downarrow z}}{\varnothing ; \varnothing \vdash \mathtt{id} \Downarrow z.[z] \to z} \qquad \dfrac{}{\varnothing ; \varnothing \vdash z \Downarrow z}}{\varnothing ; \varnothing \vdash \mathtt{id}\, z \Downarrow z}}{\varnothing ; \varnothing \vdash (\lambda x . x\, y)(\mathtt{id}\, z) \Downarrow z\, y}$$

Evaluation Types for Call-by-Name

Results

Let

- $\mathscr{D}$ be the set of **typing derivations**
- $\mathscr{R}$ be the set of **normalisation sequences**

Evaluation Types for Call-by-Name

Results

Let

- $\mathscr{D}$ be the set of **typing derivations**
- $\mathscr{R}$ be the set of **normalisation sequences**

Theorem (One-to-one correspondence between $\mathscr{D}$ and $\mathscr{R}$)

There exist mappings $f : \mathscr{D} \to \mathscr{R}$ and $g : \mathscr{R} \to \mathscr{D}$ such that:

1. *If* $\mathsf{D} \rhd \varnothing ; \varnothing \vdash t \Downarrow \mathtt{a}$ *then* $f(\mathsf{D}) \rhd t \twoheadrightarrow \mathtt{a}$.
2. *If* $\mathsf{R} \rhd t \twoheadrightarrow \mathtt{a}$ *then* $g(\mathsf{R}) \rhd \varnothing ; \varnothing \vdash t \Downarrow \mathtt{a}$.

Furthermore, f and g are mutual inverses.

Evaluation Types for Call-by-Name

Theorem (One-to-one correspondence between $\mathscr{D}$ and $\mathscr{R}$)

There exist mappings $f : \mathscr{D} \to \mathscr{R}$ and $g : \mathscr{R} \to \mathscr{D}$ such that:

1. *If* $\mathsf{D} \rhd \varnothing ; \varnothing \vdash t \Downarrow \mathtt{a}$ *then* $f(\mathsf{D}) \rhd t \twoheadrightarrow \mathtt{a}$.
2. *If* $\mathsf{R} \rhd t \twoheadrightarrow \mathtt{a}$ *then* $g(\mathsf{R}) \rhd \varnothing ; \varnothing \vdash t \Downarrow \mathtt{a}$.

Furthermore, f and g are mutual inverses.

Both mappings are obtained by **construction**:

- If $\mathsf{D} \rhd \varnothing ; \varnothing \vdash \mathtt{a} \Downarrow \mathtt{a}$, then $f(\mathsf{D}) := \epsilon$
- Otherwise, t is a β-redex $(\lambda x.s)\,u$, so $f(\mathsf{D}) := (\lambda x.s)\,u\,; f(\mathsf{D}')$, where D' is a typing derivation of $s\{x := u\}$ obtained by Subject Reduction.

Theorem (One-to-one correspondence between $\mathscr{D}$ and $\mathscr{R}$)

There exist mappings $f : \mathscr{D} \to \mathscr{R}$ and $g : \mathscr{R} \to \mathscr{D}$ such that:

1. *If* $\mathsf{D} \rhd \varnothing ; \varnothing \vdash^{n} t \Downarrow \mathsf{a}$ *then* $f(\mathsf{D}) \rhd t \twoheadrightarrow \mathsf{a}$, *with* $|f(\mathsf{D})| = n$.
2. *If* $\mathsf{R} \rhd t \twoheadrightarrow \mathsf{a}$ *with* $|\mathsf{R}| = n$ *then* $g(\mathsf{R}) \rhd \varnothing ; \varnothing \vdash^{n} t \Downarrow \mathsf{a}$.

Furthermore, f and g are mutual inverses.

Both mappings are obtained by **construction**:

- If $\mathsf{D} \rhd \varnothing ; \varnothing \vdash \mathsf{a} \Downarrow \mathsf{a}$, then $f(\mathsf{D}) := \epsilon$
- Otherwise, t is a β-redex $(\lambda x. s)\, u$, so $f(\mathsf{D}) := (\lambda x. s)\, u \,; f(\mathsf{D}')$, where D' is a typing derivation of $s\{x := u\}$ obtained by Subject Reduction.

Conclusions:

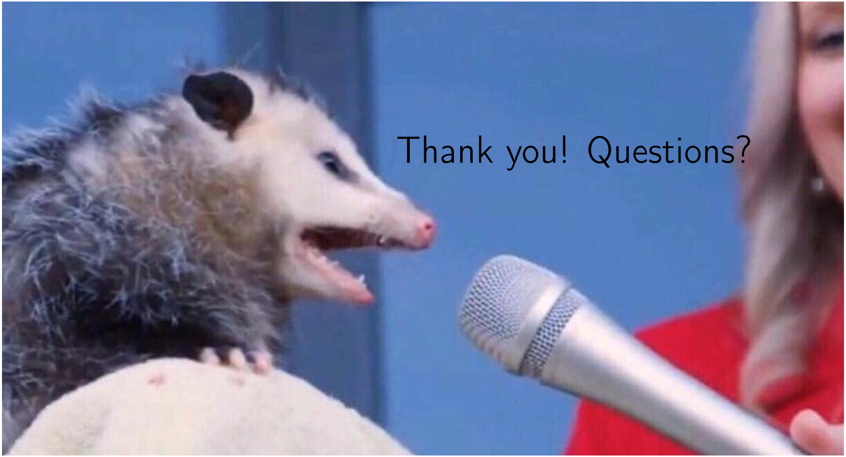
- One-to-one correspondence between typing derivations and normalisation sequences.
- This provides a refined characterisation of termination.

New result I obtained:

- Exact measures in our theorems by adding a counter in the typing judgements.

Future work:

- Propose an evaluation type system for **closed call-by-need**: it is not trivial, as its based on evaluation contexts.
- Relate evaluation type systems to other non-idempotent intersection type systems.


Thank you! Questions?

Evaluation Types for Call-by-Value

Typing rules

$$
\begin{array}{rcll}
A & ::= & \mathtt{v}.M \to \mathtt{w}.N & \\
M & ::= & [A_1, \ldots, A_n] & (n \geq 0)
\end{array}
$$

$$
\frac{x \in \mathrm{dom}(\sigma)}{\sigma \,;\, x : M \vdash x \Downarrow_{\text{value}} \sigma(x).M}
\qquad
\frac{\sigma, x : \mathtt{v} \,;\, \Gamma, x : M \vdash t \Downarrow_{\text{value}} \mathtt{w}.N}{\sigma \,;\, \Gamma \vdash \lambda x.\, t \Downarrow_{\text{value}} \mathtt{v}.M \to \mathtt{w}.N}
$$

$$
\frac{\sigma \,;\, \Gamma \vdash t \Downarrow_{\text{value}} \mathtt{v}.[\mathtt{w}.M \to \mathtt{u}.N] \qquad \sigma \,;\, \Delta \vdash s \Downarrow_{\text{value}} \mathtt{w}.M}{\sigma \,;\, \Gamma + \Delta \vdash t\, s \Downarrow_{\text{value}} \mathtt{u}.N}
$$

$$
\frac{(\sigma \,;\, \Gamma_i \vdash \lambda x.\, t \Downarrow_{\text{value}} A_i)_{i=1}^{n}}{\sigma \,;\, +_{i=1}^{n} \Gamma_i \vdash \lambda x.\, t \Downarrow_{\text{value}} (\lambda x.\, t^{\sigma}).[A_1, \ldots, A_n]}
$$