A Quantitative Interpretation for Useful Call-by-Value

Mariana Milicich

joint work with Pablo Barenbaum and Delia Kesner

University of Tartu

Estonian-Latvian Computer Science Theory Days
Tartu
24/04/2026

Evaluation Strategies

Let fix x := fix x
and constant x := 42:

Call-by-Name	Call-by-Value
> constant (fix 5)	> constant (fix 5)
> 42	> Infinite loop!

Evaluation Strategies

Let fix x := fix x
and constant x := 42:

Call-by-Name	**Call-by-Value**
> constant (fix 5) > 42	> constant (fix 5) > Infinite loop!

Evaluation strategies govern the behavior of programs.

- **Call-by-Name**: silly duplication, wise erasure
- **Call-by-Value**: wise duplication, silly erasure

Historical Background of Useful Evaluation

Silly duplication can trigger **size explosion**

$\Rightarrow$

β-reduction is not atomic

Can the λ-calculus be related to
Turing Machines from a complexity p.o.v.?

β−reduction

$(\lambda x.t)\ u \rightarrow_{\beta} t\{x \leftarrow u\}$

Historical Background of Useful Evaluation

Silly duplication can trigger **size explosion**

$\Rightarrow$

β-reduction is not atomic

Can the λ-calculus be related to
Turing Machines from a complexity p.o.v.?

β−reduction

$$(\lambda x.t)\ u \rightarrow_{\beta} t\{x \leftarrow u\}$$

Accattoli and Dal Lago (CSL-LICS 2014)

The λ-calculus can be simulated
with **polynomial overhead in time** by a RAM.

So, what is Useful Evaluation?

- An **optimisation mechanism** for a given evaluation strategy.
  It relies on **explicit substitutions**:

$$t[x \leftarrow u]$$

So, what is Useful Evaluation?

- An **optimisation mechanism** for a given evaluation strategy. It relies on **explicit substitutions**:

$$t[x \leftarrow u]$$

- Avoids **substitution steps** leading to size explosion and those not leading to the creation of function application steps.

Useful 😄	**Non-Useful** ☹
$(x\ y)[x \leftarrow \lambda z.a] \underline{\to} (\lambda z.a)\ y$	$(x\ y)[x \leftarrow z] \underline{\to} z\ y$
$(x\ y)[x \leftarrow a][a \leftarrow \lambda z.z]$ $\underline{\to}$ $(x\ y)[x \leftarrow \lambda z.z] \underline{\to} (\lambda z.z)\ y$	$x[x \leftarrow \lambda y.y] \underline{\to} \lambda y.y$

So, what is Useful Evaluation?

- An **optimisation mechanism** for a given evaluation strategy. It relies on **explicit substitutions**:

$$t[x \leftarrow u]$$

- Avoids **substitution steps** leading to size explosion and those not leading to the creation of function application steps.

Useful 😄	**Non-Useful** ☹
$(x\ y)[x \leftarrow \lambda z.a] \underline{\rightarrow} (\lambda z.a)\ y$	$(x\ y)[x \leftarrow z] \underline{\rightarrow} z\ y$
$(x\ y)[x \leftarrow a][a \leftarrow \lambda z.z] \underline{\rightarrow} (x\ y)[x \leftarrow \lambda z.z] \underline{\rightarrow} (\lambda z.z)\ y$	$x[x \leftarrow \lambda y.y] \underline{\rightarrow} \lambda y.y$

- Defined through **global predicates**, given its context-sensitivity.

Our Goal

Define the first quantitative interpretation of **useful call-by-value** through **non-idempotent intersection types**.

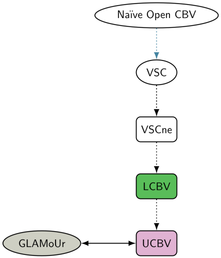
Our Goal

Define the first quantitative interpretation of **useful call-by-value** through **non-idempotent intersection types**.

Previous Key Step

Inductive definition useful call-by-value: UCBV.

(Barenbaum, Kesner, M., CADE 2025).



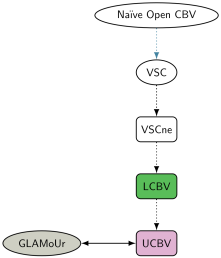
Our Goal

Define the first quantitative interpretation of **useful call-by-value** through **non-idempotent intersection types**.

Previous Key Step

Inductive definition useful call-by-value: UCBV.

(Barenbaum, Kesner, M., CADE 2025).



Results in this Talk

- The type system $\mathscr{U}$ characterises:
  - VSCne-termination (qualitatively)
  - UCBV-termination (quantitatively)
- Both VSC and UCBV are **termination equivalent**.

  ↳ (Accattoli and Paolini, FLOPS 2012)

VSC (and VSCne)

Syntax:

$$t, u ::= x \mid \lambda x.t \mid t\ u \mid t[x \leftarrow u]$$
$$v ::= x \mid \lambda x.t$$
$$\mathtt{L} ::= \Diamond \mid \mathtt{L}[x \leftarrow t]$$

Operational Semantics (VSC)

$$(\lambda x.t)\mathtt{L}\ u \rightarrow_{\mathrm{db}} t[x \leftarrow u]\mathtt{L}$$
$$t[x \leftarrow v\mathtt{L}] \rightarrow_{\mathrm{sv}} t\{x \leftarrow v\}\mathtt{L}$$

Removing *erasing* substitution steps yields the **VSCne** calculus.

UCBV in a Nutshell

$$t \to_{\rho,\mathscr{A},\mathscr{S},\mu} u$$

where:

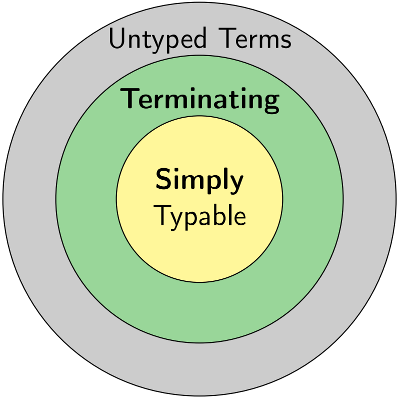
- $\rho \in \{\text{db}, \overbrace{\text{lsv}, \text{sub}(x,v)}^{\text{linear substitution}}\}$: step kinds
- $\mathscr{A}$: set of variables bound by (hereditary) abstractions in t
- $\mathscr{S}$: set of free variables or those bound by structures in t
- μ: t is applied/non-applied to an argument

$$\frac{}{(\lambda x.t)\mathtt{L}\ u \to_{\text{db},\mathscr{A},\mathscr{S},\mu} t[x \leftarrow u]\mathtt{L}} \qquad \frac{}{x \to_{\text{sub}(x,v),\mathscr{A}\cup\{x\},\mathscr{S},@} v}$$

$$\frac{t \to_{\text{sub}(x,v),\mathscr{A}\cup\{x\},\mathscr{S},\mu} t' \qquad x \notin \mathscr{A} \cup \mathscr{S} \qquad v\mathtt{L} \in \text{HAbs}_{\mathscr{A}}}{t[x \leftarrow v\mathtt{L}] \to_{\text{lsv},\mathscr{A},\mathscr{S},\mu} t'[x \leftarrow v]\mathtt{L}}$$

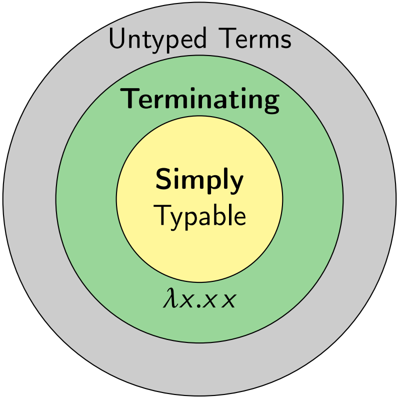
Simple vs. Intersection Types

Simple Types



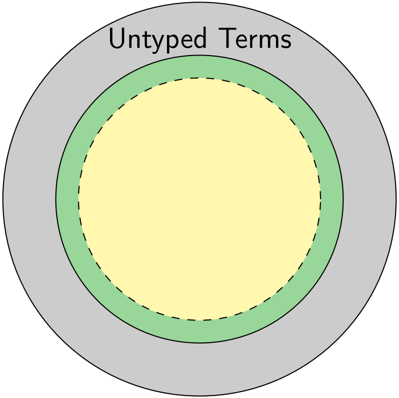
Simple vs. Intersection Types

Simple Types



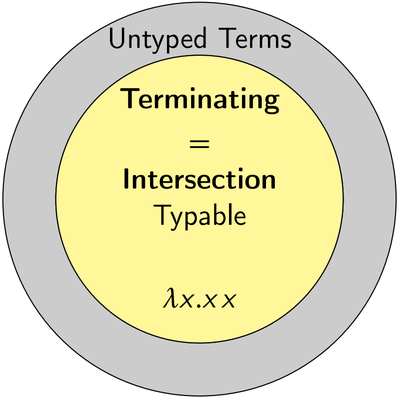
The term $\lambda x.xx$ is not simply typable

Intersection Types



Simple vs. Intersection Types

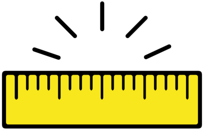
Intersection Types



The term $\lambda x.xx$ is intersection typable

Idempotent and Non-Idempotent Flavours

$$\tau, \sigma ::= \alpha \mid \tau \rightarrow \sigma \mid \tau \cap \sigma$$

Idempotent ($\tau \cap \tau = \tau$)	**Non-Idempotent** ($\tau \cap \tau \neq \tau$)
Coppo & Dezani (1980)	Gardner (1994)
Qualitative properties 	Quantitative properties 
$\vdash P : \texttt{Int}$ $\iff$ P terminates and returns an integer	$\vdash^{n} P : \texttt{Int}$ $\iff$ P terminates after n steps and returns an integer

The Type System $\mathscr{U}$

$$
\begin{array}{rrcll}
\text{Types:} & \mathscr{M}, \mathscr{N} & ::= & \underline{\mathsf{s}} \mid \mathscr{I} & \\
\text{Optional Types:} & \mathscr{M}^{?} & ::= & \mathtt{none} \mid \mathscr{M} & \\
\text{Arrow Types:} & \tau & ::= & \mathscr{M}^{?} \to \mathscr{N} & \\
\text{Multi-Types:} & \mathscr{I} & ::= & [\tau_i]_{i \in I} & I \text{ finite}
\end{array}
$$

$\mathscr{U}$ types:

- Hereditary abstractions with $\mathscr{I}$
- Structures with $\underline{\mathsf{s}}$
- Normal forms with **tight constants** $\mathtt{tt} ::= \underline{\mathsf{s}} \mid [\,]$

The Type System $\mathscr{U}$

$$
\begin{array}{rrcll}
\text{Types:} & \mathscr{M},\mathscr{N} & ::= & \underline{\mathsf{s}} \mid \mathscr{I} & \\
\text{Optional Types:} & \mathscr{M}^{?} & ::= & \mathtt{none} \mid \mathscr{M} & \\
\text{Arrow Types:} & \tau & ::= & \mathscr{M}^{?} \to \mathscr{N} & \\
\text{Multi-Types:} & \mathscr{I} & ::= & [\tau_i]_{i \in I} & I \text{ finite}
\end{array}
$$

$\mathscr{U}$ types:

- Hereditary abstractions with $\mathscr{I}$
- Structures with $\underline{\mathsf{s}}$
- Normal forms with **tight constants** $\mathtt{tt} ::= \underline{\mathsf{s}} \mid [\,]$

$$[\,] \neq \mathtt{none}$$

Typing Rules

Typing judgements: $\Gamma \vdash t : \mathscr{M}$

$$\frac{}{x : \mathscr{M} \vdash x : \mathscr{M}} \qquad \frac{\Gamma \vdash t : \underline{\mathtt{s}} \qquad \Delta \vdash u : \mathtt{tt}}{\Gamma \uplus \Delta \vdash t\ u : \underline{\mathtt{s}}}$$

$$\frac{\left(\Gamma_i ; x : \mathscr{M}_i^? \vdash t : \mathscr{N}_i\right)_{i \in I} \qquad I \text{ finite}}{\biguplus_{i \in I} \Gamma_i \vdash \lambda x.t : [\mathscr{M}_i^? \to \mathscr{N}_i]_{i \in I}}$$

$$\frac{\Gamma \vdash t : [\mathscr{M}^? \to \mathscr{N}] \qquad \mathscr{M}^? \lhd \mathscr{M} \qquad \Delta \vdash u : \mathscr{M}}{\Gamma \uplus \Delta \vdash t\, u : \mathscr{N}}$$

$$\frac{\Gamma ; x : \mathscr{M}^? \vdash t : \mathscr{N} \qquad \mathscr{M}^? \lhd \mathscr{M} \qquad \Delta \vdash u : \mathscr{M}}{\Gamma \uplus \Delta \vdash t[x \leftarrow u] : \mathscr{N}}$$

$\mathscr{U}$-Typability Characterises VSCne-Termination

Theorem

$\rhd_{\mathscr{U}} \Gamma \vdash t : \mathscr{M} \iff t$ *VSCne-terminates*

Proof.

The proof follows from well-known techniques:

⇒) Soundness relies on a <u>quantitative</u> subject reduction lemma and a substitution lemma.

⇐) Completeness relies on a <u>quantitative</u> subject expansion lemma and an anti-substitution lemma.

(Some) Rules of System $\mathscr{U}$, with Counters

Typing judgements: $\Gamma \vdash^{(m,e)} t : \mathscr{M}$

Counters (m, e) represent the number of reduction steps:

- m: distant beta
- e: linear substitution

$$\frac{e = \mathtt{ta}(\mathscr{M})}{x : \mathscr{M} \vdash^{(0, e)} x : \mathscr{M}}$$

$$\frac{\Gamma \vdash^{(m,e)} t : [\mathscr{M}^{?} \to \mathscr{N}] \quad \mathscr{M}^{?} \lhd \mathscr{M} \quad \Delta \vdash^{(m',e')} u : \mathscr{M}}{\Gamma \uplus \Delta \vdash^{(1+m+m', e+e')} t\ u : \mathscr{N}}$$

Quantitative Characterisation of UCBV-Termination

Theorem (Exact Measures)

$\triangleright \Gamma \vdash^{(m,e)} t : \mathtt{tt} \quad \iff \quad$ *t terminates in UCBV in* ***exactly*** *m* db-*steps and e* lsv-*steps*

↑ tightness

$\mathscr{V}$-Typability Characterises VSCne-Termination

Theorem (BKRV, FLOPS 2020)

$\triangleright_{\mathscr{V}} \Gamma \vdash t : \mathscr{M} \iff t$ *VSC-terminates*

$\mathscr{V}$-Typability Characterises VSCne-Termination

Theorem (BKRV, FLOPS 2020)

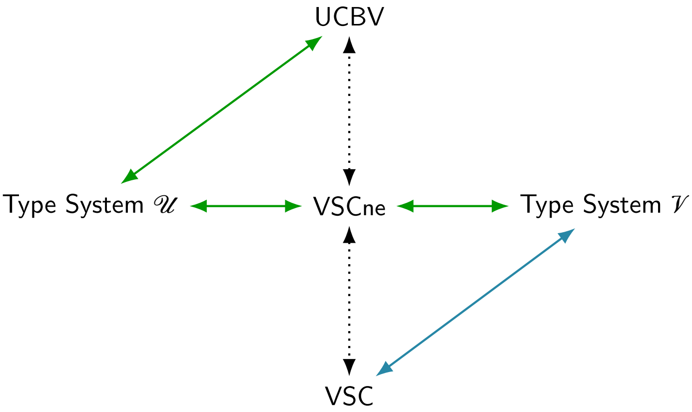
$\triangleright_{\mathscr{V}} \Gamma \vdash t : \mathscr{M} \iff t$ *VSC-terminates*

We reuse this Thm. to relate UCBV and VSC through the type systems $\mathscr{U}$ and $\mathscr{V}$:

Theorem

$\triangleright_{\mathscr{V}} \Gamma \vdash t : \mathscr{M} \iff t$ *VSCne-terminates*

Termination Equivalence Result



Conclusions

System $\mathscr{U}$ characterises termination on both VSCne and UCBV.

- $\triangleright_{\mathscr{U}} \Gamma \vdash^{(m,e)} t : \mathtt{tt} \iff$ t reaches its UCBV-normal form in exactly m db-steps and e lsv-steps.
- Both UCBV and VSCne are termination equivalent.

System $\mathscr{V}$ characterises termination on both VSC and VSCne.

- UCBV, VSCne, and VSC are termination equivalent.
- UCBV (non-erasing) and VSC (erasing) type the same terms.

Future Work

- Quantitatively interpret Strong CBV.
- Relate $\mathscr{U}$ to other definitions of useful (open) CBV.


Thank you! Questions?

VSCne

Operational Semantics (VSCne)

$$(\lambda x.t)\mathtt{L}\ u \rightarrow_{\mathsf{db}} t[x \leftarrow u]\mathtt{L}$$

$$t[x \leftarrow v\mathtt{L}] \rightarrow_{\mathsf{svne}} t\{x \leftarrow v\}\ [x \leftarrow v]\ \mathtt{L} \qquad \text{if } x \in \mathsf{fv}(t)$$

(back to the main document)

Hereditary Abstractions and Structures

Hereditary abstractions, $\mathsf{HAbs}_{\mathscr{A}}$, are given by:

$$\frac{x \in \mathscr{A}}{x \in \mathsf{HAbs}_{\mathscr{A}}} \qquad \frac{}{\lambda x.t \in \mathsf{HAbs}_{\mathscr{A}}} \qquad \frac{t \in \mathsf{HAbs}_{\mathscr{A}} \qquad x \notin \mathscr{A}}{t[x \leftarrow u] \in \mathsf{HAbs}_{\mathscr{A}}}$$

$$\frac{t \in \mathsf{HAbs}_{\mathscr{A} \cup \{x\}} \qquad x \notin \mathscr{A} \qquad u \in \mathsf{HAbs}_{\mathscr{A}}}{t[x \leftarrow u] \in \mathsf{HAbs}_{\mathscr{A}}}$$

Structures, $\mathsf{St}_{\mathscr{S}}$, are given by:

$$\frac{x \in \mathscr{S}}{x \in \mathsf{St}_{\mathscr{S}}} \qquad \frac{t \in \mathsf{St}_{\mathscr{S}}}{t\ u \in \mathsf{St}_{\mathscr{S}}}$$

$$\frac{t \in \mathsf{St}_{\mathscr{S}} \qquad x \notin \mathscr{S}}{t[x \leftarrow u] \in \mathsf{St}_{\mathscr{S}}} \qquad \frac{t \in \mathsf{St}_{\mathscr{S} \cup \{x\}} \qquad x \notin \mathscr{S} \qquad u \in \mathsf{St}_{\mathscr{S}}}{t[x \leftarrow u] \in \mathsf{St}_{\mathscr{S}}}$$

(back to the main document)

Operational Semantics of UCBV

Reduction rules:

$$\frac{}{(\lambda x.t)\mathtt{L}\,u \rightarrow_{\mathsf{db},\mathscr{A},\mathscr{S},\mu} t[x \leftarrow u]\mathtt{L}} \qquad \frac{}{x \rightarrow_{\mathsf{sub}(x,v),\mathscr{A}\cup\{x\},\mathscr{S},@} v}$$

$$\frac{t \rightarrow_{\mathsf{sub}(x,v),\mathscr{A}\cup\{x\},\mathscr{S},\mu} t' \qquad x \notin \mathscr{A} \cup \mathscr{S} \qquad v\mathtt{L} \in \mathsf{HAbs}_{\mathscr{A}}}{t[x \leftarrow v\mathtt{L}] \rightarrow_{\mathsf{lsv},\mathscr{A},\mathscr{S},\mu} t'[x \leftarrow v]\mathtt{L}}$$

Congruence rules:

$$\frac{t \rightarrow_{\rho,\mathscr{A},\mathscr{S},@} t'}{t\,u \rightarrow_{\rho,\mathscr{A},\mathscr{S},\mu} t'\,u} \qquad \frac{t \in \mathsf{St}_{\mathscr{S}} \qquad u \rightarrow_{\rho,\mathscr{A},\mathscr{S},\not@} u'}{t\,u \rightarrow_{\rho,\mathscr{A},\mathscr{S},\mu} t\,u'} \qquad \frac{u \rightarrow_{\rho,\mathscr{A},\mathscr{S},\not@} u'}{t[x \leftarrow u] \rightarrow_{\rho,\mathscr{A},\mathscr{S},\mu} t[x \leftarrow u']}$$

$$\frac{t \rightarrow_{\rho,\mathscr{A}\cup\{x\},\mathscr{S},\mu} t' \qquad u \in \mathsf{HAbs}_{\mathscr{A}} \qquad x \notin \mathscr{A} \cup \mathscr{S} \qquad x \notin \mathsf{fv}(\rho)}{t[x \leftarrow u] \rightarrow_{\rho,\mathscr{A},\mathscr{S},\mu} t'[x \leftarrow u]}$$

$$\frac{t \rightarrow_{\rho,\mathscr{A},\mathscr{S}\cup\{x\},\mu} t' \qquad u \in \mathsf{St}_{\mathscr{S}} \qquad x \notin \mathscr{A} \cup \mathscr{S} \qquad x \notin \mathsf{fv}(\rho)}{t[x \leftarrow u] \rightarrow_{\rho,\mathscr{A},\mathscr{S},\mu} t'[x \leftarrow u]}$$

(back to the main document)

System $\mathscr{U}$: Typing Rules with Counters

$$\frac{e = \mathtt{ta}(\mathscr{M})}{x : \mathscr{M} \vdash^{(0,e)} x : \mathscr{M}} \qquad \frac{\Gamma \vdash^{(m,e)} t : \underline{\mathtt{s}} \qquad \Delta \vdash^{(m',e')} u : \mathtt{tt}}{\Gamma \Cup \Delta \vdash^{(m+m',e+e')} t\ u : \underline{\mathtt{s}}}$$

$$\frac{\left(\Gamma_i; x : \mathscr{M}_i^? \vdash^{(m_i,e_i)} t : \mathscr{N}_i\right)_{i \in I} \qquad I \text{ finite}}{\Cup_{i \in I} \Gamma_i \vdash^{(\Sigma_{i \in I} m_i, \Sigma_{i \in I} e_i)} \lambda x.t : [\mathscr{M}_i^? \to \mathscr{N}_i]_{i \in I}}$$

$$\frac{\Gamma \vdash^{(m,e)} t : [\mathscr{M}^? \to \mathscr{N}] \qquad \mathscr{M}^? \lhd \mathscr{M} \qquad \Delta \vdash^{(m',e')} u : \mathscr{M}}{\Gamma \Cup \Delta \vdash^{(1+m+m',e+e')} t\, u : \mathscr{N}}$$

$$\frac{\Gamma; x : \mathscr{M}^? \vdash^{(m,e)} t : \mathscr{N} \qquad \mathscr{M}^? \lhd \mathscr{M} \qquad \Delta \vdash^{(m',e')} u : \mathscr{M}}{\Gamma \Cup \Delta \vdash^{(m+m',e+e')} t[x \leftarrow u] : \mathscr{N}}$$

(back to the main document)

System $\mathscr{V}$

Grammar of Types

$$\begin{array}{rccl}
\text{Types:} & \sigma & ::= & \alpha \mid \mathscr{M} \mid \mathscr{M} \to \sigma \\
\text{Multi-Types:} & \mathscr{M} & ::= & [\sigma_i]_{i \in I} \qquad\qquad I \text{ finite}
\end{array}$$

Typing Rules

$$\frac{}{x : \mathscr{M} \vdash x : \mathscr{M}} \qquad \frac{(\Gamma_i ; x : \mathscr{M}_i \vdash t : \sigma_i)_{i \in I}}{\Uplus_{i \in I} \Gamma_i \vdash \lambda x.t : [\mathscr{M}_i \to \sigma_i]_{i \in I}}$$

$$\frac{\Gamma \vdash t : [\mathscr{M} \to \sigma] \qquad \Delta \vdash u : \mathscr{M}}{\Gamma \Uplus \Delta \vdash t\,u : \sigma} \qquad \frac{\Gamma ; x : \mathscr{M} \vdash t : \sigma \qquad \Delta \vdash u : \mathscr{M}}{\Gamma \Uplus \Delta \vdash t[x \leftarrow u] : \sigma}$$

(back to the main document)