Useful Call-by-Value, Quantitatively

Mariana Milicich[1]*

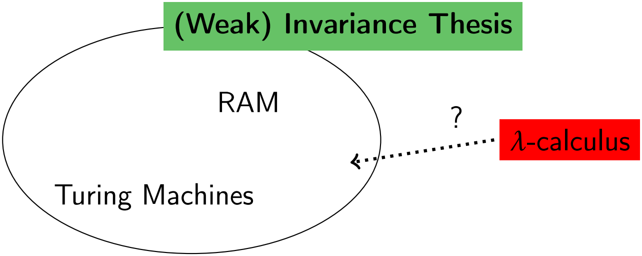
[1]Université Paris-Cité, CNRS, IRIF, France

QCOMICAL Kickoff Meeting
30th January 2025

* *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*

Motivation

- How to measure the **(time) complexity** of algorithms in functional programming languages?
- The λ-calculus and Turing Machines compute the same expressions.
- Is there a way to relate them from a complexity p.o.v.?



The λ-calculus is invariant
(B. Accattoli and U. Dal Lago, 2016)

The technique

1. The λ-calculus is implemented by a low-level language called the Linear Substitution Calculus (LSC).

2. A notion of *useful* evaluation defined on LSC is shown to be **invariant**.

The λ-calculus is invariant

(B. Accattoli and U. Dal Lago, 2016)

The technique

1. The λ-calculus is implemented by a low-level language called the Linear Substitution Calculus (LSC).

2. A notion of *useful* evaluation defined on LSC is shown to be **invariant**.

Calculi with **explicit substitutions**: $t[x \leftarrow s]$.

$$(\lambda x.t)\mathtt{L}\, s \rightarrow_{\mathsf{db}} t[x \leftarrow s]\mathtt{L} \qquad \textit{reduction at distance}$$

Principles of useful evaluation

1. **Sharing structures**:
   - $(x\,x)[x \leftarrow y\,\mathtt{id}]$ is already a normal form
2. **Substituting by an abstraction:** iff it contributes to the creation of a distant beta redex.

$$\begin{array}{rcll} x[x \leftarrow \mathtt{id}]\,y & \to & \mathtt{id}[x \leftarrow \mathtt{id}]\,y & ✓ \\ x[x \leftarrow \mathtt{id}] & \to & \mathtt{id}[x \leftarrow \mathtt{id}] & ✗ \end{array}$$

Our contributions

- **Inductive definition** of useful call-by-value (CBV) for open weak CBV (ucbv$^\bullet$).
- **Semantic interpretation** of useful CBV through quantitative types (System $\mathscr{U}$).

Towards ucbv$^{\bullet}$

1. Starting point: the Value Substitution Calculus (vsc) (B. Accattoli and L. Paolini, 2012).

Towards ucbv$^\bullet$

1. Starting point: the Value Substitution Calculus (vsc) (B. Accattoli and L. Paolini, 2012).
   - Problem: substitutes all the occurrences of variables.
     Ex.: $(x\,(y\,x))[x \leftarrow \mathtt{id}] \rightarrow (\mathtt{id}\,(y\,\mathtt{id}))[x \leftarrow \mathtt{id}]$

Towards ucbv$^{\bullet}$

1. Starting point: the Value Substitution Calculus (vsc) (B. Accattoli and L. Paolini, 2012).
   - Problem: substitutes all the occurrences of variables.
     Ex.: $(x\,(y\,x))[x \leftarrow \mathtt{id}] \rightarrow (\mathtt{id}\,(y\,\mathtt{id}))[x \leftarrow \mathtt{id}]$
2. Linear CBV: vsc with linear substitution.

Towards ucbv$^\bullet$

1. Starting point: the Value Substitution Calculus (vsc) (B. Accattoli and L. Paolini, 2012).
   - Problem: substitutes all the occurrences of variables.
     Ex.: $(x\,(y\,x))[x \leftarrow \mathtt{id}] \rightarrow (\mathtt{id}\,(y\,\mathtt{id}))[x \leftarrow \mathtt{id}]$
2. Linear CBV: vsc with linear substitution.
   - Substitutes one occurrence at a time.
     Ex.: $(x\,(y\,x))[x \leftarrow \mathtt{id}] \rightarrow (\mathtt{id}\,(y\,x))[x \leftarrow \mathtt{id}]$

Towards ucbv$^\bullet$

1. Starting point: the Value Substitution Calculus (vsc) (B. Accattoli and L. Paolini, 2012).
   - Problem: substitutes all the occurrences of variables.
     Ex.: $(x\,(y\,x))[x \leftarrow \mathtt{id}] \rightarrow (\mathtt{id}\,(y\,\mathtt{id}))[x \leftarrow \mathtt{id}]$
2. Linear CBV: vsc with linear substitution.
   - Substitutes one occurrence at a time.
     Ex.: $(x\,(y\,x))[x \leftarrow \mathtt{id}] \rightarrow (\mathtt{id}\,(y\,x))[x \leftarrow \mathtt{id}]$
   - Problem: substituting abstractions indiscriminately.
     Ex.: $\ldots \rightarrow (\mathtt{id}\,(y\,x))[x \leftarrow \mathtt{id}] \rightarrow (\mathtt{id}\,(y\,\mathtt{id}))[x \leftarrow \mathtt{id}]$

The calculus ucbv$^\bullet$

Operational semantics

- Inspired by *A Strong call-by-need calculus*, T. Balabonski, A. Lanco, G. Melquiond.
- Information on the evaluation context is stored in $\xrightarrow{\bullet}$:
  - $\{\mathsf{db}, \mathsf{lsv}, \mathsf{sub}_{(x,v)}\}$: names
  - $\mathscr{A}$: variables bound by (indirect) abstractions
  - $\mathscr{S}$: variables bound by structures
  - μ: term is in applied/non applied position

$$\frac{}{(\lambda x.t)\mathrm{L}\, s \xrightarrow{\bullet}_{\mathsf{db},\mathscr{A},\mathscr{S},\mu} t[x \leftarrow s]\mathrm{L}} \qquad \frac{}{x \xrightarrow{\bullet}_{\mathsf{sub}_{(x,v)},\mathscr{A}\cup\{x\},\mathscr{S},@} v}$$

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

ucbv$^\bullet$

Operational property

Diamond property

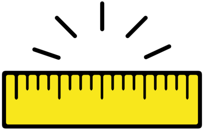
$$\begin{tikzcd}[column sep=small]
 & t \arrow[dl] \arrow[dr] & \\
t_1 \arrow[dr, dotted] & & t_2 \arrow[dl, dotted] \\
 & t' &
\end{tikzcd}$$

Corollaries

1. (All) sequences to normal form **have the same length**.
2. The relation $\stackrel{\bullet}{\to}_{\rho,\emptyset,\mathscr{S},\not@}$ is **confluent**.

Intersection types

$$\tau ::= \alpha \mid \tau \to \tau \mid \tau \cap \tau$$

Idempotent $(\tau \cap \tau = \tau)$	**Non-Idempotent** $(\tau \cap \tau \neq \tau)$
Coppo & Dezani (1978)	Gardner (1994)
Qualitative properties 	Quantitative properties 
$\vdash P : \texttt{Int}$ $\iff$ P normalises and returns an integer	$\vdash^{n} P : \texttt{Int}$ $\iff$ P normalises after n steps and returns an integer

Type System $\mathscr{U}$

Grammar of types:

$$
\begin{array}{lcl@{\qquad}lcl}
\mathscr{M} & ::= & \mathsf{n} \mid \mathscr{I} & \tau & ::= & \mathscr{M}^{?} \to \mathscr{M} \\
\mathscr{I} & ::= & [\tau_k]_{k \in K} & \mathscr{M}^{?} & ::= & \mathtt{none} \mid \mathscr{M}
\end{array}
$$

- Hereditary abstractions are typed with $\mathscr{I}$.
- Structures are typed with n.
- Normal forms are typed with **tight constants** $\mathsf{tt} ::= \mathsf{n} \mid [\,]$.
- Controlled form of *weakening* $\mathscr{M}_1^{?} \lhd \mathscr{M}_2$:

$$\mathtt{none} \lhd \mathsf{tt} \qquad \mathscr{M} \lhd \mathscr{M}$$

Type System $\mathscr{U}$

Grammar of types:

$$
\begin{array}{lcl@{\qquad}lcl}
\mathscr{M} & ::= & \mathsf{n} \mid \mathscr{I} & \tau & ::= & \mathscr{M}^{?} \to \mathscr{M} \\
\mathscr{I} & ::= & [\tau_k]_{k \in K} & \mathscr{M}^{?} & ::= & \mathtt{none} \mid \mathscr{M}
\end{array}
$$

- Hereditary abstractions are typed with $\mathscr{I}$.
- Structures are typed with n.
- Normal forms are typed with **tight constants** $\mathtt{tt} ::= \mathsf{n} \mid [\,]$.
- Controlled form of *weakening* $\mathscr{M}_1^{?} \triangleleft \mathscr{M}_2$:

$$\mathtt{none} \triangleleft \mathtt{tt} \qquad \mathscr{M} \triangleleft \mathscr{M}$$

The meaning of $[\,]$ is not the same as the one of $\mathtt{none}$.

Type System $\mathscr{U}$

Typing rules

$$\frac{n = \mathtt{ta}(\mathscr{M})}{x : \mathscr{M} \vdash^{(0,n)} x : \mathscr{M}} \qquad \frac{(\Gamma_i; x : \mathscr{M}_i^? \vdash^{(m_i,e_i)} t : \mathscr{N}_i)_{i \in I}}{+_{i\in I}\Gamma_i \vdash^{(+_{i\in I} m_i, +_{i\in I} e_i)} \lambda x.t : [\mathscr{M}_i^? \to \mathscr{N}_i]_{i\in I}}$$

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

Type System $\mathscr{U}$

Typing rules

$$\frac{n = \mathtt{ta}(\mathscr{M})}{x : \mathscr{M} \vdash^{(0,n)} x : \mathscr{M}} \qquad \frac{(\Gamma_i; x : \mathscr{M}_i^? \vdash^{(m_i,e_i)} t : \mathscr{N}_i)_{i \in I}}{+_{i \in I}\Gamma_i \vdash^{(+_{i\in I} m_i, +_{i \in I} e_i)} \lambda x.t : [\mathscr{M}_i^? \to \mathscr{N}_i]_{i \in I}}$$

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

Tightness (Accattoli, Graham-Lengrand, and Kesner)

- **Exact** information from **minimal** typing derivations.
- $y : \mathsf{n}; z : \mathsf{n} \vdash^{(1,0)} (\lambda x.y) z : \mathsf{n}$ ✓
- $y : \mathsf{n} \vdash^{(0,0)} \lambda x.y : [\mathtt{none} \to \mathsf{n}]$ ✗

Type System $\mathscr{U}$

Typing rules

$$\frac{n = \mathtt{ta}(\mathcal{M})}{x : \mathcal{M} \vdash^{(0,n)} x : \mathcal{M}} \qquad \frac{(\Gamma_i; x : \mathcal{M}_i^{?} \vdash^{(m_i,e_i)} t : \mathcal{N}_i)_{i\in I}}{+_{i\in I}\Gamma_i \vdash^{(+_{i\in I} m_i, +_{i\in I} e_i)} \lambda x.t : [\mathcal{M}_i^{?} \to \mathcal{N}_i]_{i\in I}}$$

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

Tightness (Accattoli, Graham-Lengrand, and Kesner)

- **Exact** information from **minimal** typing derivations.
- $y : \mathrm{n}; z : \mathrm{n} \vdash^{(1,0)} (\lambda x.y) z : \mathrm{n}$ ✓
- $y : \mathrm{n} \vdash^{(0,0)} \lambda x.y : [\mathtt{none} \to \mathrm{n}]$ ✗

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

- m: number of distant beta steps
- e: number of substitution steps

System $\mathscr{U}$: example

System $\mathscr{U}$ is **resource aware**: all hypothesis must be used.

$$\frac{\dfrac{\dfrac{\text{Problem here!}}{y:\mathsf{n}; x:\mathsf{n} \vdash^{(0,0)} y:\mathsf{n}}}{y:\mathsf{n} \vdash^{(0,0)} \lambda x.y : [\mathsf{n} \to \mathsf{n}]} \qquad \mathsf{n} \lhd \mathsf{n} \qquad \dfrac{}{z:\mathsf{n} \vdash^{(0,0)} z:\mathsf{n}}}{y:\mathsf{n}; z:\mathsf{n} \vdash^{(1,0)} (\lambda x.y)\, z : \mathsf{n}}$$

System $\mathscr{U}$: example

System $\mathscr{U}$ is **resource aware**: all hypothesis must be used.

$$\dfrac{\dfrac{\dfrac{}{y : \mathsf{n}; x : \texttt{none} \vdash^{(0,0)} y : \mathsf{n}}}{y : \mathsf{n} \vdash^{(0,0)} \lambda x.y : [\texttt{none} \rightarrow \mathsf{n}]} \qquad \texttt{none} \lhd \mathsf{n} \qquad \dfrac{}{z : \mathsf{n} \vdash^{(0,0)} z : \mathsf{n}}}{y : \mathsf{n}; z : \mathsf{n} \vdash^{(1,0)} (\lambda x.y)\, z : \mathsf{n}}$$

Theorem (Exact characterisation of normalisation)

t is tightly typable with counters (m, e) in system $\mathscr{U}$

$$\Updownarrow$$

t normalises in ucbv$^\bullet$ *after m* db*-steps and e* lsv*-steps*

Theorem (Exact characterisation of normalisation)

$$t \textit{ is tightly typable with counters } ({\color{red}m}, {\color{green}e}) \textit{ in system } \mathscr{U}$$
$$\Updownarrow$$
$$t \textit{ normalises in } \mathsf{ucbv}^{\bullet} \textit{ after } {\color{red}m} \ \mathsf{db}\textit{-steps and } {\color{green}e} \ \mathsf{lsv}\textit{-steps}$$

Moreover, system $\mathscr{U}$ characterises **strong normalisation**:

- System $\mathscr{U}$ characterises (weak) normalisation
- ucbv$^{\bullet}$ enjoys the diamond property

Conclusions:

- The inductive nature of $\text{ucbv}^{\bullet}$ allows us to define the first semantic interpretation of useful CBV, system $\mathscr{U}$.
- System $\mathscr{U}$ provides:
  - Qualitative props.: characterises termination of $\text{ucbv}^{\bullet}$
  - Quantitative props.: provides exact measures for reduction sequences to normal form

> Simpler than syntactic presentations of useful evaluation.

Future work:

- Design type systems for more complex settings of useful eval.
- Extend our inductive approach to useful strong CBV.
- Relate $\text{ucbv}^{\bullet}$ to other presentations of useful CBV.

Thank you! Questions?

Linear Open CBV (lcbv°)

Semantics

$$\frac{}{(\lambda x.t)\mathtt{L}\, s \overset{\circ}{\to}_{\mathsf{db}} t[x \leftarrow s]\mathtt{L}} \qquad \frac{}{x \overset{\circ}{\to}_{\mathsf{sub}_{(x,v)}} v} \qquad \frac{t \overset{\circ}{\to}_{\mathsf{sub}_{(x,v)}} t'}{t[x \leftarrow v\mathtt{L}] \overset{\circ}{\to}_{\mathsf{lsv}} t'[x \leftarrow v]\mathtt{L}}$$

Congruence rules:

$$\frac{t \overset{\circ}{\to}_{\rho} t'}{t\, s \overset{\circ}{\to}_{\rho} t'\, s} \qquad \frac{s \overset{\circ}{\to}_{\rho} s'}{t\, s \overset{\circ}{\to}_{\rho} t\, s'}$$

$$\frac{t \overset{\circ}{\to}_{\rho} t' \quad x \notin \mathsf{fv}(\rho)}{t[x \leftarrow s] \overset{\circ}{\to}_{\rho} t'[x \leftarrow s]} \qquad \frac{s \overset{\circ}{\to}_{\rho} s'}{t[x \leftarrow s] \overset{\circ}{\to}_{\rho} t[x \leftarrow s']}$$

ucbv$^\bullet$

Semantics

Reduction rules:

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

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

Congruence rules:

$$\frac{t \overset{\bullet}{\to}_{\rho,\mathscr{A},\mathscr{S},@} t'}{t\, s \overset{\bullet}{\to}_{\rho,\mathscr{A},\mathscr{S},\mu} t'\, s} \qquad \frac{t \in \mathsf{St}_{\mathscr{S}} \quad s \overset{\bullet}{\to}_{\rho,\mathscr{A},\mathscr{S},\not{@}} s'}{t\, s \overset{\bullet}{\to}_{\rho,\mathscr{A},\mathscr{S},\mu} t\, s'} \qquad \frac{s \overset{\bullet}{\to}_{\rho,\mathscr{A},\mathscr{S},\not{@}} s'}{t[x \leftarrow s] \overset{\bullet}{\to}_{\rho,\mathscr{A},\mathscr{S},\mu} t[x \leftarrow s']}$$

$$\frac{t \overset{\bullet}{\to}_{\rho,\mathscr{A}\cup\{x\},\mathscr{S},\mu} t' \quad s \in \mathsf{HAbs}_{\mathscr{A}} \quad x \notin \mathscr{A} \cup \mathscr{S} \quad x \notin \mathsf{fv}(\rho)}{t[x \leftarrow s] \overset{\bullet}{\to}_{\rho,\mathscr{A},\mathscr{S},\mu} t'[x \leftarrow s]}$$

$$\frac{t \overset{\bullet}{\to}_{\rho,\mathscr{A},\mathscr{S}\cup\{x\},\mu} t' \quad s \in \mathsf{St}_{\mathscr{S}} \quad x \notin \mathscr{A} \cup \mathscr{S} \quad x \notin \mathsf{fv}(\rho)}{t[x \leftarrow s] \overset{\bullet}{\to}_{\rho,\mathscr{A},\mathscr{S},\mu} t'[x \leftarrow s]}$$

Useful Open CBV

Normal forms

$$\frac{x \in \mathscr{A} \implies \mu = \not@}{x \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S},\mu}} \qquad \frac{}{\lambda x.t \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S},\not@}}$$

$$\frac{t \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S},@} \quad s \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S},\not@}}{t\,s \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S},\mu}}$$

$$\frac{t \in \mathsf{NF}^{\bullet}_{\mathscr{A}\cup\{x\},\mathscr{S},\mu} \quad s \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S},\not@} \quad s \in \mathsf{HAbs}_{\mathscr{A}}}{t[x \leftarrow s] \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S},\mu}}$$

$$\frac{t \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S}\cup\{x\},\mu} \quad s \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S},\not@} \quad s \in \mathsf{St}_{\mathscr{S}}}{t[x \leftarrow s] \in \mathsf{NF}^{\bullet}_{\mathscr{A},\mathscr{S},\mu}}$$

System $\mathscr{U}$

Typing rules

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

$$\frac{(\Gamma_i ; x : \mathcal{M}_i^{?} \vdash^{(m_i,e_i)} t : \mathcal{N}_i)_{i \in I}}{+_{i \in I} \Gamma_i \vdash^{(+_{i \in I} m_i, +_{i \in I} e_i)} \lambda x.t : [\mathcal{M}_i^{?} \to \mathcal{N}_i]_{i \in I}}\,\mathtt{abs}$$

$$\frac{\Gamma \vdash^{(m,e)} t : [\mathcal{M}^{?} \to \mathcal{N}] \qquad \mathcal{M}^{?} \lhd \mathcal{M} \qquad \Delta \vdash^{(m',e')} s : \mathcal{M}}{\Gamma + \Delta \vdash^{(\underline{1}+m+m',e+e')} t\,s : \mathcal{N}}\,\mathtt{appC}$$

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

Counters represent **exactly** the number of reduction steps:

- m: function application steps
- e: substitution steps