LAMBDA CALCULI THROUGH THE LENS OF LINEAR LOGIC

Lesson 4: **A Subsuming Framework Inspired from Linear Logic**

Delia KESNER

Email : kesner@irif.fr

URL : www.irif.fr/~kesner

Université Paris Cité and CNRS

What is this talk about?

Reason about fundamental properties of different models of computation homogeneously

What is this talk about?

Reason about fundamental **properties** of different models of computation homogeneously

Static

Dynamic

Reason about fundamental properties of
different models of computation
homogeneously

Reason about fundamental properties of

different models of computation

homogeneously

NAME Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

VALUE argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Reason about fundamental properties of **different models of computation** homogeneously

$\boxed{\text{NAME}}$ Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

Example: $(\lambda x.\mathtt{fst}(x, x))\ 3^2$

$\boxed{\text{VALUE}}$ argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Reason about fundamental properties of **different models of computation** homogeneously

NAME Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

Example: $(\lambda x.\mathtt{fst}(x,x))\,3^2 \twoheadrightarrow \mathtt{fst}(3^2, 3^2)$

VALUE argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Reason about fundamental properties of **different models of computation** homogeneously

NAME Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

Example: $(\lambda x.\texttt{fst}(x,x))\,3^2 \twoheadrightarrow \texttt{fst}(3^2,3^2) \twoheadrightarrow 3^2 \twoheadrightarrow 9$

VALUE argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Reason about fundamental properties of **different models of computation** homogeneously

NAME Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

Example: $(\lambda x.\mathtt{fst}(x,x))\,3^2 \twoheadrightarrow \mathtt{fst}(3^2,3^2) \twoheadrightarrow 3^2 \twoheadrightarrow 9$

VALUE argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Example: $(\lambda x.\mathtt{fst}(x,x))\,3^2$

Reason about fundamental properties of **different models of computation** homogeneously

NAME Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

Example: $(\lambda x.\mathtt{fst}(x,x))\,3^2 \twoheadrightarrow \mathtt{fst}(3^2,3^2) \twoheadrightarrow 3^2 \twoheadrightarrow 9$

VALUE argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Example: $(\lambda x.\mathtt{fst}(x,x))\,3^2 \twoheadrightarrow (\lambda x.\mathtt{fst}(x,x))\,9$

Reason about fundamental properties of **different models of computation** homogeneously

$\boxed{\textbf{NAME}}$ Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

Example: $(\lambda x.\mathtt{fst}(x,x))\, 3^2 \twoheadrightarrow \mathtt{fst}(3^2, 3^2) \twoheadrightarrow 3^2 \twoheadrightarrow 9$

$\boxed{\textbf{VALUE}}$ argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Example: $(\lambda x.\mathtt{fst}(x,x))\, 3^2 \twoheadrightarrow (\lambda x.\mathtt{fst}(x,x))\, 9 \twoheadrightarrow \mathtt{fst}(9,9)$

Reason about fundamental properties of **different models of computation** homogeneously

NAME Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

Example: $(\lambda x.\mathtt{fst}(x,x))\,3^2 \twoheadrightarrow \mathtt{fst}(3^2,3^2) \twoheadrightarrow 3^2 \twoheadrightarrow 9$

VALUE argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Example: $(\lambda x.\mathtt{fst}(x,x))\,3^2 \twoheadrightarrow (\lambda x.\mathtt{fst}(x,x))\,9 \twoheadrightarrow \mathtt{fst}(9,9) \twoheadrightarrow 9$

Reason about fundamental properties of **different models of computation** homogeneously

$\boxed{\text{NAME}}$ Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

In particular: $(\lambda x.9)\,\Omega \twoheadrightarrow 9$, where Ω is a non-terminating expression

$\boxed{\text{VALUE}}$ argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

In particular: $(\lambda x.9)\,\Omega \twoheadrightarrow \ldots \twoheadrightarrow \ldots \twoheadrightarrow \infty$

Reason about fundamental properties of **different models of computation** homogeneously

NAME Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

Well understood in the theory of programming, but not very used in practice

VALUE argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Reason about fundamental properties of **different models of computation** homogeneously

NAME Argument expressions are consumed without any prior evaluation
”**any expression** can be passed as an **actual argument**”

Well understood in the theory of programming, but not very used in practice

VALUE argument expressions are evaluated before being consumed
”**only values** can be passed as **actual arguments**”

Not very well understood in the theory of programming, but very used in practice

Reason about fundamental properties of **different models of computation** homogeneously

NAME Argument expressions are consumed without any prior evaluation
"**any expression** can be passed as an **actual argument**"

Well understood in the theory of programming, but not very used in practice

VALUE argument expressions are evaluated before being consumed
"**only values** can be passed as **actual arguments**"

Not very well understood in the theory of programming, but very used in practice

Moreover, it is not easy to understand one from the other.

Reason about fundamental properties of different models of computation

homogeneously

Reason about fundamental properties of different models of computation **homogeneously**

By means of a **subsuming** framework inspired from Linear Logic: the **(Distant) Bang Calculus** $\boxed{\textbf{BANG}}$

(Bucciarelli-K.-Ríos-Viso'20)

Take Away Message

Take Away Message

Capture fundamental properties
of **NAME**/ **VALUE**/ **OTHERS**
from the corresponding properties
of **BANG**


WHY
WHY?

- **Simplified and unified understanding**

- **Simplified and unified understanding**
- **Intuitive transfer of results**

- Simplified and unified understanding
- Intuitive transfer of results
- Reveals invariant properties

- Simplified and unified understanding
- Intuitive transfer of results
- Reveals invariant properties
- Facilitates formalization and proof techniques

Agenda

Levy's Call-by-Push-Value

Girard's Linear Logic

The Foundations of the Subsuming Framework

Levy's Call-by-Push-Value

values
versus
computations

Girard's Linear Logic

linear resources
versus
erasable/duplicable resources

The Foundations of the Subsuming Framework

Levy's Call-by-Push-Value

values
versus
computations

Girard's Linear Logic

linear resources
versus
erasable/duplicable resources

both subsume

Call-by-Name $\boxed{\text{NAME}}$

Call-by-Value $\boxed{\text{VALUE}}$

Levy's Call-by-Push-Value

The Distant

BANG-Calculus

Girard's Linear Logic

values
versus
computations

linear resources
versus
erasable/duplicable resources

both subsume

Call-by-Name $\boxed{\text{NAME}}$

Call-by-Value $\boxed{\text{VALUE}}$

A Subsuming Framework Inspired from Linear Logic

The Distant **BANG**-Calculus (Historical Development)

- A first bang calculus **(Ehrhard'16)**:

The Distant **BANG**-Calculus (Historical Development)

- A first bang calculus **(Ehrhard'16)**:
  - bridges the gap between **CBPV** and Linear Logic
  - **incomplete** (some terms denoting non-terminating programs can be artificially blocked)

The Distant **BANG**-Calculus (Historical Development)

- A first bang calculus **(Ehrhard'16)**:
  - ☺ bridges the gap between **CBPV** and Linear Logic
  - ☹ **incomplete** (some terms denoting non-terminating programs can be artificially blocked)
- A second bang calculus with commuting conversions **(Ehrhard-Guerrieri'16)**:
  - ☺ recovers completeness
  - ☹ **breaks confluence**

The Distant **BANG**-Calculus (Historical Development)

- A first bang calculus **(Ehrhard'16)**:
  - bridges the gap between **CBPV** and Linear Logic
  - **incomplete** (some terms denoting non-terminating programs can be artificially blocked)
- A second bang calculus with commuting conversions **(Ehrhard-Guerrieri'16)**:
  - recovers completeness
  - **breaks confluence**
- The (Distant) **BANG**-Calculus **(Bucciarelli-K.-Ríos-Viso'20)**:
  - recovers **completeness**
  - recovers **confluence**

value/bang

dereliction

explicit substitution

Terms: $t, u ::= x \mid tu \mid \lambda x.t \mid\ !\,t \mid \mathrm{der}\,t \mid t[x \backslash u]$

The (Distant) **BANG**-Calculus

value/bang

dereliction

explicit substitution

Terms: $t, u ::= x \mid tu \mid \lambda x.t \mid\ !\,t \mid \mathrm{der}\, t \mid t[x\backslash u]$

Operational Semantics: reduction relation $\rightarrow_{\mathbf{BANG}}$.

The (Distant) **BANG**-Calculus

value/bang

dereliction

explicit substitution

Terms: $t, u ::= x \mid tu \mid \lambda x.t \mid\ !\,t \mid \mathrm{der}\,t \mid t[x\backslash u]$

Operational Semantics: reduction relation $\rightarrow_{\mathbf{BANG}}$.

Beta:

$$(\lambda x.t)\,u \quad\rightarrow\quad t[x\backslash u]$$

Substitution:

$$t[x\backslash !\,u] \quad\rightarrow\quad t\{x\backslash u\}$$

Value/Computation:

$$\mathrm{der}\,(!\,t) \quad\rightarrow\quad t$$

The (Distant) **BANG**-Calculus

Distant calculi are inspired from **linear logic** (they avoid blocking situations).

Distant **Operational Semantics**: reduction relation $\rightarrow_{\mathbf{BANG}}$.

$$\begin{array}{rcl} \textbf{Beta:} & & \\ (\lambda x.t)\, u & \rightarrow & t[x\backslash u] \\ \textbf{Substitution:} & & \\ t[x\backslash !\, u] & \rightarrow & t\{x\backslash u\} \\ \textbf{Value/Computation:} & & \\ \mathrm{der}\,(!\, t) & \rightarrow & t \end{array}$$

Distant calculi are inspired from **linear logic** (they avoid blocking situations).

$t\,\mathtt{L}$ abbreviates $t\,[x_1\backslash u_1]\ldots[x_n\backslash u_n]$ $(n \geq 0)$

Distant **Operational Semantics**: reduction relation $\rightarrow_{\mathbf{BANG}}$.

Beta:

$$(\lambda x.t)\,\mathtt{L}\;u \quad \rightarrow \quad t[x\backslash u]\,\mathtt{L}$$

Substitution:

$$t[x\backslash(!\,u)\,\mathtt{L}\,] \quad \rightarrow \quad t\{x\backslash u\}\,\mathtt{L}$$

Value/Computation:

$$\mathrm{der}\,((!\,t)\,\mathtt{L}\,) \quad \rightarrow \quad t\,\mathtt{L}$$

The (Distant) **BANG**-Calculus

Let $\mathtt{I} := \lambda x.!\, x$. Then,

$$(\lambda y.x)\ [x \backslash !\, \mathtt{I}]\ (!\, \mathtt{I})$$

Distant **Operational Semantics**: reduction relation $\rightarrow_{\mathbf{BANG}}$.

$$\begin{array}{rcl}
\textbf{Beta:} & & \\
(\lambda x.t)\ \mathtt{L}\ u & \rightarrow & t[x \backslash u]\ \mathtt{L} \\
\textbf{Substitution:} & & \\
t[x \backslash (!\, u)\ \mathtt{L}\ ] & \rightarrow & t\{x \backslash u\}\ \mathtt{L} \\
\textbf{Value/Computation:} & & \\
\mathrm{der}\,((!\, t)\ \mathtt{L}\ ) & \rightarrow & t\ \mathtt{L}
\end{array}$$

Let $\mathtt{I} := \lambda x.!\,x$. Then,

$$(\lambda y.x)\,[x\backslash !\,\mathtt{I}]\,(!\,\mathtt{I}) \quad\rightarrow\quad x[y\backslash !\,\mathtt{I}][x\backslash !\,\mathtt{I}]$$

Distant **Operational Semantics**: reduction relation $\rightarrow_{\textbf{BANG}}$.

Beta:

$$(\lambda x.t)\,\mathtt{L}\,u \quad\rightarrow\quad t[x\backslash u]\,\mathtt{L}$$

Substitution:

$$t[x\backslash(!\,u)\,\mathtt{L}\,] \quad\rightarrow\quad t\{x\backslash u\}\,\mathtt{L}$$

Value/Computation:

$$\mathrm{der}\,((!\,t)\,\mathtt{L}\,) \quad\rightarrow\quad t\,\mathtt{L}$$

The (Distant) **BANG**-Calculus

Let $\mathtt{I} := \lambda x.!\,x$. Then,

$$(\lambda y.x)\,[x\backslash !\,\mathtt{I}]\,(!\,\mathtt{I}) \quad\rightarrow\quad x[y\backslash !\,\mathtt{I}][x\backslash !\,\mathtt{I}] \quad\rightarrow\quad x[x\backslash !\,\mathtt{I}]$$

Distant Operational Semantics: reduction relation $\rightarrow_{\mathbf{BANG}}$.

Beta:

$$(\lambda x.t)\,\mathtt{L}\,u \quad\rightarrow\quad t[x\backslash u]\,\mathtt{L}$$

Substitution:

$$t[x\backslash(!\,u)\,\mathtt{L}\,] \quad\rightarrow\quad t\{x\backslash u\}\,\mathtt{L}$$

Value/Computation:

$$\mathrm{der}\,((!\,t)\,\mathtt{L}\,) \quad\rightarrow\quad t\,\mathtt{L}$$

Let $\mathtt{I} := \lambda x.!\, x$. Then,

$$(\lambda y.x)\,[x\backslash !\,\mathtt{I}]\,(!\,\mathtt{I}) \quad\to\quad x[y\backslash !\,\mathtt{I}][x\backslash !\,\mathtt{I}] \quad\to\quad x[x\backslash !\,\mathtt{I}] \quad\to\quad \mathtt{I}$$

Distant Operational Semantics: reduction relation $\to_{\mathbf{BANG}}$.

Beta:

$$(\lambda x.t)\,\mathtt{L}\; u \quad\to\quad t[x\backslash u]\,\mathtt{L}$$

Substitution:

$$t[x\backslash (!\,u)\,\mathtt{L}\,] \quad\to\quad t\{x\backslash u\}\,\mathtt{L}$$

Value/Computation:

$$\mathrm{der}\,((!\,t)\,\mathtt{L}\,) \quad\to\quad t\;\mathtt{L}$$

The (Distant) **BANG**-Calculus

Let $\mathtt{I} := \lambda x.!\,x$. Then,

$$(\lambda y.x)\,[x\backslash !\,\mathtt{I}]\,(!\,\mathtt{I}) \quad\to\quad x[y\backslash !\,\mathtt{I}][x\backslash !\,\mathtt{I}] \quad\to\quad x[x\backslash !\,\mathtt{I}] \quad\to\quad \mathtt{I}$$

Remark:

Reduction does not proceed inside values/bangs

Value/Computation:

$$\mathrm{der}\,((!\,t)\,\mathtt{L}\,) \quad\to\quad t\,\mathtt{L}$$

The (Distant) **BANG**-Calculus

Let $\mathtt{I} := \lambda x.!\,x$. Then,

$$(\lambda y.x)\,[x\backslash !\,\mathtt{I}]\,(!\,\mathtt{I}) \quad\to\quad x[y\backslash !\,\mathtt{I}][x\backslash !\,\mathtt{I}] \quad\to\quad x[x\backslash !\,\mathtt{I}] \quad\to\quad \mathtt{I}$$

Remark:

Calculus $\neq$ Strategy

The calculus is **non-deterministic** (as the original Church's λ-calculus)

Value/Computation:

$$\mathrm{der}\,((!\,t)\,\mathtt{L}\,) \quad\to\quad t\,\mathtt{L}$$

Agenda

Encoding Untyped Call-by-Name and Call-by-Value

$t^{\mathtt{N}} : \boxed{\textsf{NAME}} \mapsto \boxed{\textsf{BANG}}$

$t^{\mathtt{V}} : \boxed{\textsf{VALUE}} \mapsto \boxed{\textsf{BANG}}$

Encoding Untyped Call-by-Name and Call-by-Value

$t^{\mathtt{N}}$: **NAME** $\mapsto$ **BANG**

$t^{\mathtt{V}}$: **VALUE** $\mapsto$ **BANG**

General Form of Expected Results

- t verifies property **P** in **NAME** **if and only if** $t^{\mathtt{N}}$ verifies property **P** in **BANG**.
- t verifies property **P** in **VALUE** **if and only if** $t^{\mathtt{V}}$ verifies property **P** in **BANG**.

Girard's Translations

$\_^{\mathtt{N}} : \boxed{\text{NAME}} \mapsto \boxed{\text{BANG}}$

$$
\begin{aligned}
x^{\mathtt{N}} &\stackrel{\text{def}}{=} x \\
\lambda x.t^{\mathtt{N}} &\stackrel{\text{def}}{=} \lambda x.\, t^{\mathtt{N}} \\
t\,u^{\mathtt{N}} &\stackrel{\text{def}}{=} t^{\mathtt{N}}\, !\, u^{\mathtt{N}} \\
t[x\backslash u]^{\mathtt{N}} &\stackrel{\text{def}}{=} t^{\mathtt{N}}[x\backslash !\, u^{\mathtt{N}}]
\end{aligned}
$$

Girard's Translations

$\boxed{\_}^{\mathtt{N}} : \boxed{\textsf{NAME}} \mapsto \boxed{\textsf{BANG}}$ $\qquad$ $\boxed{\_}^{\mathtt{V}} : \boxed{\textsf{VALUE}} \mapsto \boxed{\textsf{BANG}}$

$$
\begin{array}{rcl@{\qquad}rcl}
\boxed{x}^{\mathtt{N}} & \stackrel{\text{def}}{=} & x & \boxed{x}^{\mathtt{V}} & \stackrel{\text{def}}{=} & !\, x \\
\boxed{\lambda x.t}^{\mathtt{N}} & \stackrel{\text{def}}{=} & \lambda x.\, \boxed{t}^{\mathtt{N}} & \boxed{\lambda x.t}^{\mathtt{V}} & \stackrel{\text{def}}{=} & !\, \lambda x.\, \boxed{t}^{\mathtt{V}} \\
\boxed{t\, u}^{\mathtt{N}} & \stackrel{\text{def}}{=} & \boxed{t}^{\mathtt{N}}\, !\, \boxed{u}^{\mathtt{N}} & \boxed{t\, u}^{\mathtt{V}} & \stackrel{\text{def}}{=} & \mathrm{der}\,(\boxed{t}^{\mathtt{V}})\, \boxed{u}^{\mathtt{V}} \\
\boxed{t[x\backslash u]}^{\mathtt{N}} & \stackrel{\text{def}}{=} & \boxed{t}^{\mathtt{N}}[x\backslash !\, \boxed{u}^{\mathtt{N}}] & \boxed{t[x\backslash u]}^{\mathtt{V}} & \stackrel{\text{def}}{=} & \boxed{t}^{\mathtt{V}}[x\backslash \boxed{u}^{\mathtt{V}}]
\end{array}
$$

Girard's Translations

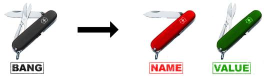
$\_^{\mathtt{N}} : \boxed{\textbf{NAME}} \mapsto \boxed{\textbf{BANG}}$ $\quad\quad$ $\_^{\mathtt{V}} : \boxed{\textbf{VALUE}} \mapsto \boxed{\textbf{BANG}}$

$$
\begin{array}{rcl@{\qquad}rcl}
x^{\mathtt{N}} & \stackrel{\text{def}}{=} & x & x^{\mathtt{V}} & \stackrel{\text{def}}{=} & !\,x \\
\lambda x.t^{\mathtt{N}} & \stackrel{\text{def}}{=} & \lambda x.\,t^{\mathtt{N}} & \lambda x.t^{\mathtt{V}} & \stackrel{\text{def}}{=} & !\,\lambda x.\,t^{\mathtt{V}} \\
t\,u^{\mathtt{N}} & \stackrel{\text{def}}{=} & t^{\mathtt{N}}\,!\,u^{\mathtt{N}} & t\,u^{\mathtt{V}} & \stackrel{\text{def}}{=} & \mathrm{der}\,(t^{\mathtt{V}})\,u^{\mathtt{V}} \\
t[x\backslash u]^{\mathtt{N}} & \stackrel{\text{def}}{=} & t^{\mathtt{N}}[x\backslash !\,u^{\mathtt{N}}] & t[x\backslash u]^{\mathtt{V}} & \stackrel{\text{def}}{=} & t^{\mathtt{V}}[x\backslash u^{\mathtt{V}}]
\end{array}
$$

There are **several variations** of Girard's translations, by different authors, each with different properties: we keep all of them in our **toolbox** for what's next.

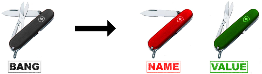
Deriving Different Properties

Derive property **P** for **NAME** and **VALUE** from property **P** for **BANG**.



Deriving Different Properties

Derive property **P** for **NAME** and **VALUE** from property **P** for **BANG**.



P := **confluence**

P := **factorization**

P := **typability with intersection types**

P := **(qualitative/quantitative) termination**

P := **inhabitation**

P := **meaningfulness**

P := **genericity**

Deriving Confluence

Take $\mathbf{P}$:= confluence

Deriving Confluence

Take P := confluence

Definition

A reduction relation enjoys **confluence** if for every terms t, u, v such that $u \twoheadleftarrow t \twoheadrightarrow v$, there is a term s such that $u \twoheadrightarrow s \twoheadleftarrow v$. Graphically,

$$\begin{tikzcd} t \arrow[r, two heads] \arrow[d, two heads] & v \arrow[d, two heads] \\ u \arrow[r, two heads] & s \end{tikzcd}$$

Deriving Confluence

Take P := confluence

Definition

A reduction relation enjoys **confluence** if for every terms t, u, v such that $u \twoheadleftarrow t \twoheadrightarrow v$, there is a term s such that $u \twoheadrightarrow s \twoheadleftarrow v$. Graphically,

$$\begin{array}{ccc} t & \twoheadrightarrow & v \\ \downarrow\!\!\downarrow & & \downarrow\!\!\downarrow \\ u & \twoheadrightarrow & s \end{array}$$

Theorem (Bucciarelli-K.-Ríos-Viso'20)

$\boxed{\textbf{BANG}}$ is confluent.

Deriving Confluence

Take P := confluence

Definition

A reduction relation enjoys **confluence** if for every terms t, u, v such that $u \twoheadleftarrow t \twoheadrightarrow v$, there is a term s such that $u \twoheadrightarrow s \twoheadleftarrow v$. Graphically,

$$\begin{array}{ccc} t & \twoheadrightarrow & v \\ \downarrow\!\!\downarrow & & \downarrow\!\!\downarrow \\ u & \twoheadrightarrow & s \end{array}$$

Theorem (Bucciarelli-K.-Ríos-Viso'20)

BANG is confluent.

Preservation Theorem (Arrial-Guerrieri-K.'24)

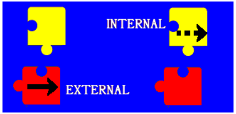
NAME and **VALUE** confluence can be derived from **BANG** confluence.

Deriving Factorization

Take $\mathbf{P}$:= **factorization**

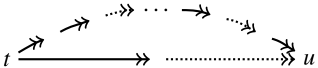
Deriving Factorization

Take P := **factorization**



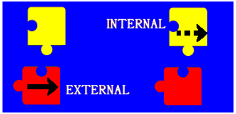
(Informal) Definition

A reduction relation enjoys **factorization** if every sequence can be rearranged so that (relevant) **EXTERNAL** steps are performed before (irrelevant) **INTERNAL** steps.



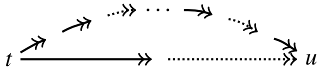
Deriving Factorization

Take **P** := **factorization**



(Informal) Definition

A reduction relation enjoys **factorization** if every sequence can be rearranged so that (relevant) **EXTERNAL** steps are performed before (irrelevant) **INTERNAL** steps.

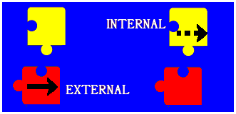


Theorem

BANG admits factorization.

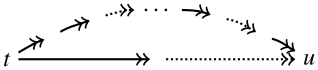
Deriving Factorization

Take P := factorization



(Informal) Definition

A reduction relation enjoys **factorization** if every sequence can be rearranged so that (relevant) **EXTERNAL** steps are performed before (irrelevant) **INTERNAL** steps.

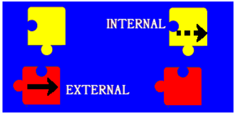


Theorem

BANG admits factorization.

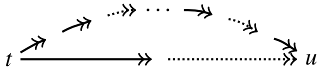
Deriving Factorization

Take $\mathbf{P}$:= **factorization**



(Informal) Definition

A reduction relation enjoys **factorization** if every sequence can be rearranged so that (relevant) **EXTERNAL** steps are performed before (irrelevant) **INTERNAL** steps.



Theorem

BANG admits factorization.

Preservation Theorem **(Arrial-Guerrieri-K.'24)**

NAME and **VALUE** factorization can be derived from **BANG** factorization.

Agenda

Non-Idempotent Intersection Systems for $\boxed{\text{NAME}}$, $\boxed{\text{VALUE}}$, $\boxed{\text{BANG}}$

- An intersection type system/model $\mathcal{N}$ for $\boxed{\text{NAME}}$ **(Gardner'94).**
- An intersection type system/model $\mathcal{V}$ for $\boxed{\text{VALUE}}$ **(Ehrhard'12, Bucciarelli-K.-Ríos-Viso'20).**
- An intersection type system/model $\mathcal{B}$ for $\boxed{\text{BANG}}$ **(Bucciarelli-K.-Ríos-Viso'20).**

Non-Idempotent Intersection Systems for $\boxed{\text{NAME}}$, $\boxed{\text{VALUE}}$, $\boxed{\text{BANG}}$

- An intersection type system/model $\mathcal{N}$ for $\boxed{\text{NAME}}$ **(Gardner'94).**
- An intersection type system/model $\mathcal{V}$ for $\boxed{\text{VALUE}}$ **(Ehrhard'12, Bucciarelli-K.-Ríos-Viso'20).**
- An intersection type system/model $\mathcal{B}$ for $\boxed{\text{BANG}}$ **(Bucciarelli-K.-Ríos-Viso'20).**

Qualitative termination

$$t \text{ is typable in } \mathcal{N}/\mathcal{V}/\mathcal{B}$$

$$\iff$$

$$t \text{ terminates in } \boxed{\text{NAME}}/\boxed{\text{VALUE}}/\boxed{\text{BANG}}$$

About Type Systems $\mathcal{N}$, $\mathcal{V}$ and $\mathcal{B}$

- Type System $\mathcal{N}$ = Gardner's system $\mathcal{H}$ in Lesson 3.
- Type System $\mathcal{V}$ (tomorrow)
- Type System $\mathcal{B}$ (next slide)

Type System $\mathcal{B}$

Grammar:

$$
\begin{array}{lrcl}
\textbf{(Types)} & \mathtt{A} & ::= & \iota \mid \mathtt{M} \mid \mathtt{M} \rightarrow \mathtt{A} \\
\textbf{(Multi-Types)} & \mathtt{M} & ::= & [\mathtt{A}_i]_{i \in I}
\end{array}
$$

Type System $\mathcal{B}$

Grammar:

$$\begin{array}{llll} \textbf{(Types)} & \mathtt{A} & ::= & \iota \mid \mathtt{M} \mid \mathtt{M} \rightarrow \mathtt{A} \\ \textbf{(Multi-Types)} & \mathtt{M} & ::= & [\mathtt{A}_i]_{i \in I} \end{array}$$

Typing Rules:

$$\frac{}{x : [\mathtt{A}] \vdash x : \mathtt{A}} \qquad \frac{\Gamma \vdash t : \mathtt{A}}{\Gamma \setminus x \vdash \lambda x.t : \Gamma(x) \rightarrow \mathtt{A}} \qquad \frac{\Gamma \vdash t : \mathtt{M} \rightarrow \mathtt{A} \qquad \Delta \vdash u : \mathtt{M}}{\Gamma \sqcup \Delta \vdash tu : \mathtt{A}}$$

Type System $\mathcal{B}$

Grammar:

$$
\begin{array}{llll}
\textbf{(Types)} & \mathtt{A} & ::= & \iota \mid \mathtt{M} \mid \mathtt{M} \to \mathtt{A} \\
\textbf{(Multi-Types)} & \mathtt{M} & ::= & [\mathtt{A}_i]_{i \in I}
\end{array}
$$

Typing Rules:

$$
\frac{}{x : [\mathtt{A}] \vdash x : \mathtt{A}}
\qquad
\frac{\Gamma \vdash t : \mathtt{A}}{\Gamma \setminus x \vdash \lambda x.t : \Gamma(x) \to \mathtt{A}}
\qquad
\frac{\Gamma \vdash t : \mathtt{M} \to \mathtt{A} \qquad \Delta \vdash u : \mathtt{M}}{\Gamma \sqcup \Delta \vdash tu : \mathtt{A}}
$$

$$
\frac{\Gamma \vdash t : \mathtt{A} \qquad \Delta \vdash u : \Gamma(x)}{(\Gamma \setminus x) \sqcup \Delta \vdash t[x \backslash u] : \mathtt{A}}
\qquad
\frac{(\Gamma_i \vdash t : \mathtt{A}_i)_{i \in I}}{\sqcup_{i \in I} \Gamma_i \vdash !t : [\mathtt{A}_i]_{i \in I}}
\qquad
\frac{\Gamma \vdash t : [\mathtt{A}]}{\Gamma \vdash \mathrm{der}\, t : \mathtt{A}}
$$

Deriving Typability and (Qualitative) Termination for NAME, VALUE

Take P := typability with intersection types

Preservation Theorem (Bucciarelli-K.-Ríos-Viso'20)

- $\triangleright_{\mathcal{N}} \Gamma \vdash t : \mathtt{A}$ in **NAME** **if and only if** $\triangleright_{\mathcal{B}} \Gamma \vdash t^{\mathtt{N}} : \mathtt{A}$ in **BANG**.
- $\triangleright_{\mathcal{V}} \Gamma \vdash t : \mathtt{A}$ in **VALUE** **if and only if** $\triangleright_{\mathcal{B}} \Gamma \vdash t^{\mathtt{V}} : \mathtt{A}$ in **BANG**.

Deriving Typability and (Qualitative) Termination for NAME, VALUE

Take P := typability with intersection types

Preservation Theorem (Bucciarelli-K.-Ríos-Viso'20)

- $\triangleright_{\mathcal{N}} \Gamma \vdash t : \mathtt{A}$ in NAME **if and only if** $\triangleright_{\mathcal{B}} \Gamma \vdash t^{\mathrm{N}} : \mathtt{A}$ in BANG.
- $\triangleright_{\mathcal{V}} \Gamma \vdash t : \mathtt{A}$ in VALUE **if and only if** $\triangleright_{\mathcal{B}} \Gamma \vdash t^{\mathrm{V}} : \mathtt{A}$ in BANG.

Take P := (qualitative) termination

Preservation Corollary

- t is NAME-terminating **if and only if** t^{N} is BANG-terminating.
- t is VALUE-terminating **if and only if** t^{V} is BANG-terminating.

Deriving Typability and (Qualitative) Termination for NAME, VALUE

Take P := typability with intersection types

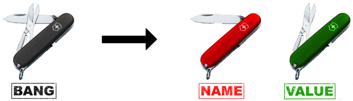
Preservation Theorem (Bucciarelli-K.-Ríos-Viso'20)

- $\triangleright_{\mathcal{N}} \Gamma \vdash t : \mathtt{A}$ in NAME **if and only if** $\triangleright_{\mathcal{B}} \Gamma \vdash t^{\mathrm{N}} : \mathtt{A}$ in BANG.
- $\triangleright_{\mathcal{V}} \Gamma \vdash t : \mathtt{A}$ in VALUE **if and only if** $\triangleright_{\mathcal{B}} \Gamma \vdash t^{\mathrm{V}} : \mathtt{A}$ in BANG.

Take P := (qualitative) termination

Preservation Corollary

- t is NAME-terminating **if and only if** t^{N} is BANG-terminating.
- t is VALUE-terminating **if and only if** t^{V} is BANG-terminating.



Take $\mathbf{P}$:= **(quantitative) termination**

Take P := (quantitative) termination

Can we derive evaluation **length** for **NAME** / **VALUE**
from evaluation **length** for **BANG** ?

Towards Quantitative Termination

Take $\mathbf{P}$:= **(quantitative) termination**

Can we derive evaluation **length** for $\boxed{\text{NAME}}$ / $\boxed{\text{VALUE}}$
from evaluation **length** for $\boxed{\text{BANG}}$?

Can we derive **size** of $\boxed{\text{NAME}}$ / $\boxed{\text{VALUE}}$ normal forms
from **size** of $\boxed{\text{BANG}}$ normal forms?

Towards Quantitative Termination

Take P := (quantitative) termination

Can we derive evaluation **length** for **NAME** / **VALUE**
from evaluation **length** for **BANG** ?

Can we derive **size** of **NAME** / **VALUE** normal forms
from **size** of **BANG** normal forms?

- We appeal to non-idempotent intersection type systems with **counters** **(Accattoli-GrahamLengrand-K.'18)**

Towards Quantitative Termination

Take P := (quantitative) termination

Can we derive evaluation **length** for **NAME** / **VALUE**
from evaluation **length** for **BANG** ?

Can we derive **size** of **NAME** / **VALUE** normal forms
from **size** of **BANG** normal forms?

- We appeal to non-idempotent intersection type systems with **counters (Accattoli-GrahamLengrand-K.'18)**
- Type judgments $\Gamma \vdash t : \sigma$ are decorated with **counters** $\Gamma \vdash^{(C_1,\ldots,C_n)} t : \sigma$

Towards Quantitative Termination

Take P := (quantitative) termination

Can we derive evaluation **length** for **NAME** / **VALUE**
from evaluation **length** for **BANG** ?

Can we derive **size** of **NAME** / **VALUE** normal forms
from **size** of **BANG** normal forms?

- We appeal to non-idempotent intersection type systems with **counters** **(Accattoli-GrahamLengrand-K.'18)**
- Type judgments $\Gamma \vdash t : \sigma$ are decorated with **counters** $\Gamma \vdash^{(C_1,\ldots,C_n)} t : \sigma$
- New intersection type systems $\mathcal{EN}/\mathcal{EV}/\mathcal{EB}$ with two counters (**L**,**S**), capturing respectively **length** and **size**

Deriving Quantitative Termination for $\boxed{\text{NAME}}$ and $\boxed{\text{VALUE}}$

Quantitative Termination with **Counters** for $\boxed{\text{NAME}}$/ $\boxed{\text{VALUE}}$/ $\boxed{\text{BANG}}$

t is (tightly) typable in $\mathcal{EN}/\mathcal{EV}/\mathcal{EB}$: $\quad \Pi \triangleright_{\mathcal{EN}/\mathcal{EV}/\mathcal{EB}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$

$$\Longleftrightarrow$$

t terminates in $\boxed{\text{NAME}}$/ $\boxed{\text{VALUE}}$/ $\boxed{\text{BANG}}$ with **length** $\mathbf{L}$ and returns a result of **size** $\mathbf{S}$.

Deriving Quantitative Termination for NAME and VALUE

Quantitative Termination with Counters for NAME / VALUE / BANG

t is (tightly) typable in $\mathcal{EN}/\mathcal{EV}/\mathcal{EB}$: $\Pi \triangleright_{\mathcal{EN}/\mathcal{EV}/\mathcal{EB}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$

$\Longleftrightarrow$

t terminates in NAME / VALUE / BANG with **length** L and returns a result of **size** S.

Preservation Theorem **(K.-Viso'22)**

- $\triangleright_{\mathcal{EN}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$ in NAME **if and only if** $\triangleright_{\mathcal{EB}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t^{\mathtt{N}} : \mathtt{A}$ in BANG.
- $\triangleright_{\mathcal{EV}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$ in VALUE **if and only if** $\triangleright_{\mathcal{EB}} \Gamma \vdash^{(\mathsf{L'},\mathsf{S})} t^{\mathtt{V}} : \mathtt{A}$ in BANG,
  where L and L' are calculable one from the typing derivation of the other.

Deriving Quantitative Termination for $\boxed{\text{NAME}}$ and $\boxed{\text{VALUE}}$

Quantitative Termination with **Counters** for $\boxed{\text{NAME}}$/ $\boxed{\text{VALUE}}$/ $\boxed{\text{BANG}}$

t is (tightly) typable in $\mathcal{EN}/\mathcal{EV}/\mathcal{EB}$: $\quad \Pi \rhd_{\mathcal{EN}/\mathcal{EV}/\mathcal{EB}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$

$$\Longleftrightarrow$$

t terminates in $\boxed{\text{NAME}}$/ $\boxed{\text{VALUE}}$/ $\boxed{\text{BANG}}$ with **length** L and returns a result of **size** S.

Preservation Theorem **(K.-Viso'22)**

- $\rhd_{\mathcal{EN}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$ in $\boxed{\text{NAME}}$ **if and only if** $\rhd_{\mathcal{EB}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t^{\mathtt{N}} : \mathtt{A}$ in $\boxed{\text{BANG}}$.
- $\rhd_{\mathcal{EV}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$ in $\boxed{\text{VALUE}}$ **if and only if** $\rhd_{\mathcal{EB}} \Gamma \vdash^{(\mathsf{L}',\mathsf{S})} t^{\mathtt{V}} : \mathtt{A}$ in $\boxed{\text{BANG}}$,
  where L and L' are calculable one from the typing derivation of the other.

Preservation Corollary

Length and **Size** for $\boxed{\text{NAME}}$ and $\boxed{\text{VALUE}}$ evaluation relate (back and forth) to **Length** and **Size** for $\boxed{\text{BANG}}$ evaluation.

Deriving Quantitative Termination for NAME and VALUE

Quantitative Termination with Counters for NAME / VALUE / BANG

t is (tightly) typable in $\mathcal{EN}/\mathcal{EV}/\mathcal{EB}$: $\quad \Pi \triangleright_{\mathcal{EN}/\mathcal{EV}/\mathcal{EB}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$

$$\Longleftrightarrow$$

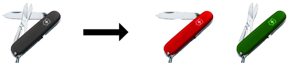
t terminates in NAME / VALUE / BANG with **length** $\mathtt{L}$ and returns a result of **size** $\mathtt{S}$.

Preservation Theorem (K.-Viso'22)

- $\triangleright_{\mathcal{EN}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$ in NAME **if and only if** $\triangleright_{\mathcal{EB}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t^{\mathtt{N}} : \mathtt{A}$ in BANG.
- $\triangleright_{\mathcal{EV}} \Gamma \vdash^{(\mathsf{L},\mathsf{S})} t : \mathtt{A}$ in VALUE **if and only if** $\triangleright_{\mathcal{EB}} \Gamma \vdash^{(\mathsf{L'},\mathsf{S})} t^{\mathtt{V}} : \mathtt{A}$ in BANG,
  where L and L' are calculable one from the typing derivation of the other.

Preservation Corollary

Length and **Size** for NAME and VALUE evaluation relate (back and forth) to
Length and **Size** for BANG evaluation.



Agenda

Typing Problem

$$\Gamma? \vdash t : \mathtt{A}?$$

Typing Problem

$$\Gamma ? \vdash t : A ?$$

Term Language

Duality between Typing and Inhabitation

Typing Problem

$\Gamma? \vdash t : \mathtt{A}?$

Term Language

Typing System

Duality between Typing and Inhabitation

Typing Problem

$\Gamma ? \vdash t : \mathtt{A} ?$

Term Language **Typing System**

Inhabitation Problem

$\Gamma \vdash \mathbf{t} ? : \mathtt{A}$

Duality between Typing and Inhabitation

Typing Problem

$\Gamma? \vdash t : \mathtt{A}?$

Term Language **Typing System**

Inhabitation Problem

$\Gamma \vdash \mathbf{t}? : \mathtt{A}$

Typing System

Duality between Typing and Inhabitation

Typing Problem

$\boldsymbol{\Gamma}? \vdash t : \mathtt{A}?$

Term Language **Typing System**

Inhabitation Problem

$\Gamma \vdash \mathbf{t}? : \mathtt{A}$

Term Language **Typing System**

Duality between Typing and Inhabitation

Typing Problem

$\Gamma? \vdash t : \mathtt{A}?$

Term Language **Typing System**

Inhabitation Problem

$\Gamma \vdash \mathbf{t}? : \mathtt{A}$

Term Language **Typing System**

Proof Search

Program Synthesis

The Inhabitation Problem (IP) for $\boxed{\textbf{BANG}}$

Take $\mathbf{P}$:= **inhabitation**

The Inhabitation Problem (IP) for $\boxed{\textbf{BANG}}$

Take $\mathbf{P}$:= inhabitation

- An inhabitation algorithm for **BANG**

The Inhabitation Problem (IP) for $\boxed{\textbf{BANG}}$

Take P := inhabitation

- An inhabitation algorithm for **BANG**
- Which **terminates**, is **sound** and **complete**

The Inhabitation Problem (IP) for $\boxed{\textbf{BANG}}$

Take $\mathbf{P}$:= **inhabitation**

- An inhabitation algorithm for **BANG**
- Which **terminates**, is **sound** and **complete**
- **Implemented** in OCaml by **(V. Arrial)**

The Inhabitation Problem (IP) for $\boxed{\textbf{BANG}}$

Take P := inhabitation

- An inhabitation algorithm for **BANG**
- Which **terminates**, is **sound** and **complete**
- **Implemented** in OCaml by **(V. Arrial)**

The Inhabitation Problem (IP) for BANG

Take P := inhabitation

- An inhabitation algorithm for **BANG**
- Which **terminates**, is **sound** and **complete**
- **Implemented** in OCaml by **(V. Arrial)**

Preservation Theorem (Arrial-Guerrieri-K.'23)

Decidability of the NAME and VALUE IP can be derived from **decidability** of the BANG IP.

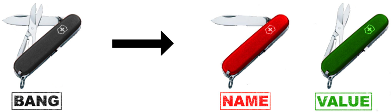
The Inhabitation Problem (IP) for $\boxed{\textbf{BANG}}$

Take **P** := **inhabitation**

- An inhabitation algorithm for **BANG**
- Which **terminates**, is **sound** and **complete**
- **Implemented** in OCaml by **(V. Arrial)**

Preservation Theorem (Arrial-Guerrieri-K.'23)

Decidability of the $\boxed{\text{NAME}}$ and $\boxed{\text{VALUE}}$ IP can be derived from **decidability** of the $\boxed{\text{BANG}}$ IP.



Duality between Typing and Inhabitation

In **Lambda-Calculus**:

	Typing $\Gamma? \vdash t : \mathtt{A}?$	**Inhabitation** $\Gamma \vdash t? : \mathtt{A}$
Simple Types	Decidable	Decidable
Idempotent Types	Undecidable	Undecidable
Non-Idempotent Types	Undecidable	**NAME** Decidable **VALUE** **?**

Duality between Typing and Inhabitation

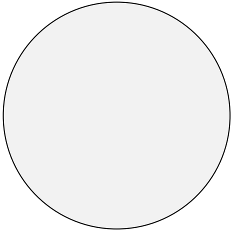
In **Lambda-Calculus**:

	Typing $\Gamma ? \vdash t : \mathtt{A} ?$	**Inhabitation** $\Gamma \vdash t ? : \mathtt{A}$
Simple Types	Decidable	Decidable
Idempotent Types	Undecidable	Undecidable
Non-Idempotent Types	Undecidable	NAME Decidable VALUE **Decidable**

Agenda

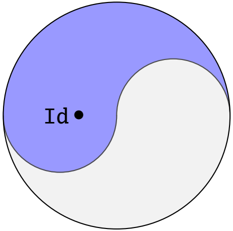
Intuitions

Terms

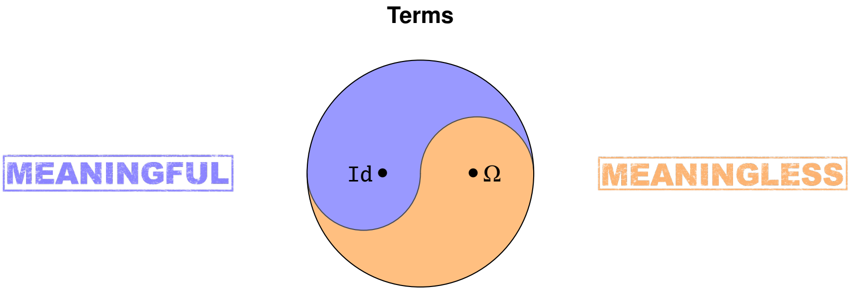


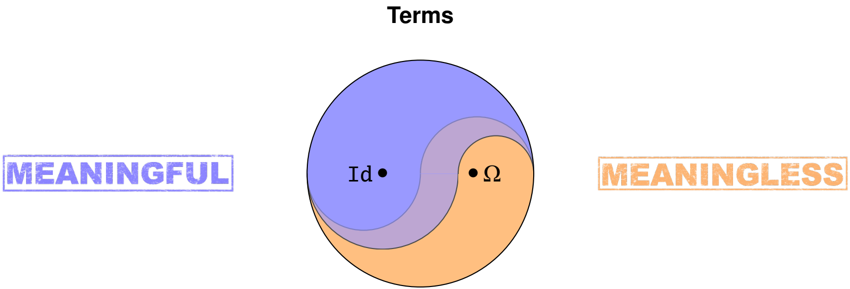
Intuitions

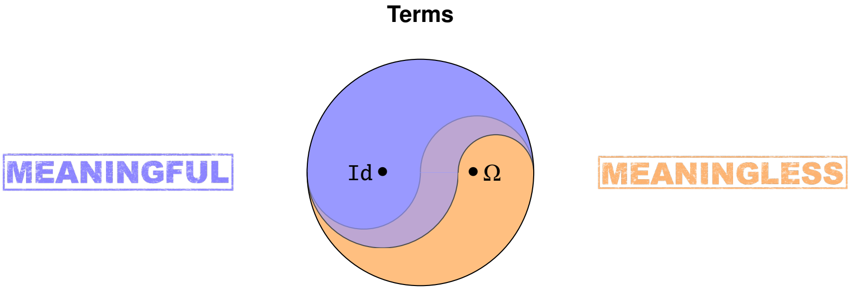
Terms



Intuitions




Terms
MEANINGFUL
Id
Ω
MEANINGLESS



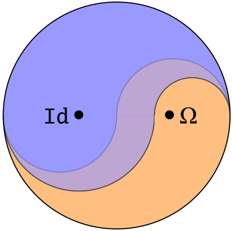
Clashes (reducing to ill-formed terms) should be MEANINGLESS

Example: $(!\,t)\,u$ and $t\,(\lambda x.u)$

Intuitions

Terms

MEANINGFUL



MEANINGLESS

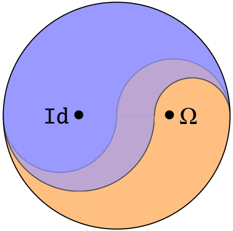
Non-terminating terms "producing" **observable terms** should be MEANINGFUL

Example: $Y := \lambda f.\Delta_f \,!\, \Delta_f$, where $\Delta_f := \lambda x.f \,!\, (x \,!\, x)$

Intuitions

Terms

MEANINGFUL



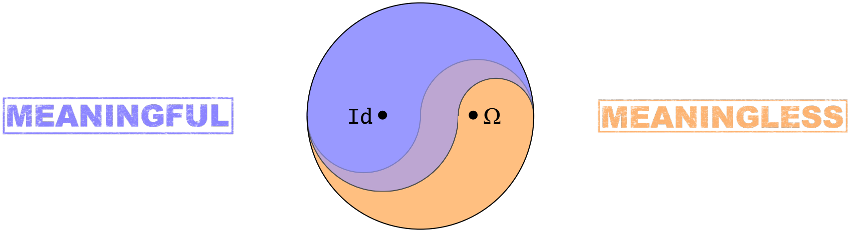
MEANINGLESS

Some **premature-like** normal forms should be MEANINGLESS

Example: $(\lambda x.\Delta)\,(y\,!\,\mathtt{I})\,!\,\Delta \cong \Delta\,!\,\Delta = \Omega$, where Ω is a non-terminating term

Intuitions

Terms



There are different sanity checks for MEANINGFUL / MEANINGLESS, e.g. :

- Logical Characterization
- Genericity Property
- Consistency when equating all meaningless terms

Formal Definition of Meaningfulness

Needed key notions:

- **Testing contexts** being able to **feed** a given term with arguments: $\mathtt{T} \ ::= \ \Diamond \mid \mathtt{T}\, s \mid (\lambda x.\mathtt{T})\, s$
- **Observable** terms (*i.e.* **values**) providing at least some minimal information.

Formal Definition of Meaningfulness

Needed key notions:

- **Testing contexts** being able to **feed** a given term with arguments: $\mathtt{T} ::= \Diamond \mid \mathtt{T}\, s \mid (\lambda x.\mathtt{T})\, s$
- **Observable** terms (*i.e.* **values**) providing at least some minimal information.

Definition

A term t is MEANINGFUL in $\boxed{\textbf{BANG}}$ if
there is a **testing context** $\mathtt{T}$ and an **observable** term $!\,u$ such that $\mathtt{T}\langle t\rangle$ reduces to $!\,u$.

Formal Definition of Meaningfulness

Needed key notions:

- **Testing contexts** being able to **feed** a given term with arguments: $\mathtt{T} ::= \Diamond \mid \mathtt{T}\, s \mid (\lambda x.\mathtt{T})\, s$
- **Observable** terms (*i.e.* **values**) providing at least some minimal information.

Definition

A term t is MEANINGFUL in **BANG** if
there is a **testing context** $\mathtt{T}$ and an **observable** term $!\,u$ such that $\mathtt{T}\langle t\rangle$ reduces to $!\,u$.

Examples:

$\lambda x.x$ MEANINGFUL

Formal Definition of Meaningfulness

Needed key notions:

- **Testing contexts** being able to **feed** a given term with arguments: $\mathtt{T} \ ::= \ \diamond \mid \mathtt{T}\, s \mid (\lambda x.\mathtt{T})\, s$
- **Observable** terms (*i.e.* **values**) providing at least some minimal information.

Definition

A term t is MEANINGFUL in **BANG** if
there is a **testing context** $\mathtt{T}$ and an **observable** term $!\,u$ such that $\mathtt{T}\langle t\rangle$ reduces to $!\,u$.

Examples:

$\lambda x.x$ MEANINGFUL $\quad \Omega, x\Omega$ MEANINGLESS

Formal Definition of Meaningfulness

Needed key notions:

- **Testing contexts** being able to **feed** a given term with arguments: $\mathtt{T} \ ::= \ \diamond \mid \mathtt{T}\, s \mid (\lambda x.\mathtt{T})\, s$
- **Observable** terms (*i.e.* **values**) providing at least some minimal information.

Definition

A term t is MEANINGFUL in **BANG** if
there is a **testing context** $\mathtt{T}$ and an **observable** term $!\,u$ such that $\mathtt{T}\langle t\rangle$ reduces to $!\,u$.

Examples:

$\lambda x.x$ MEANINGFUL $\quad$ Ω, $x\Omega$ MEANINGLESS $\quad$ $x\,x$ MEANINGLESS

Towards a Logical Characterization of Meaningfulness

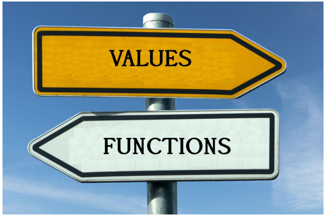
- The term $\mathsf{t} := x\,x$ is MEANINGLESS

Towards a Logical Characterization of Meaningfulness

- The term $\mathsf{t} := x\,x$ is MEANINGLESS
- Indeed, t **cannot** produce a value within an appropriate testing context `T`:

Towards a Logical Characterization of Meaningfulness

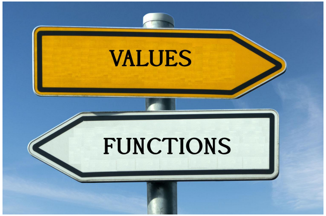
- The term t:= $x\,x$ is MEANINGLESS
- Indeed, t **cannot** produce a value within an appropriate testing context `T`:



The **rightmost** x in t should be substituted by a **VALUE**.

The **leftmost** x in t should be substituted by a **FUNCTION**.

- The term t:= $x\,x$ is MEANINGLESS
- Indeed, t **cannot** produce a value within an appropriate testing context `T`:



The **rightmost** x in t should be substituted by a **VALUE**.

The **leftmost** x in t should be substituted by a **FUNCTION**.

- Similar phenomenon in other calculi having **incompatible** data structures
  (Bucciarelli-K.-RonchiDellaRocca'15)

Meaningfulness for $\boxed{\text{NAME}}$ and $\boxed{\text{VALUE}}$

Meaningfulness for $\boxed{\text{NAME}}$ and $\boxed{\text{VALUE}}$

- $\boxed{\text{NAME}}$-meaningfulness := solvability **(Wadsworth'71)**.
- $\boxed{\text{VALUE}}$-meaningfulness := potential valuability **(Paolini-RonchiDellaRocca'99)**.

Meaningfulness for $\boxed{\text{NAME}}$ and $\boxed{\text{VALUE}}$

- $\boxed{\text{NAME}}$-meaningfulness := solvability **(Wadsworth'71)**.
  A term t is **solvable** if there is a testing context $\mathtt{T}$ s.t. $\mathtt{T}\langle t\rangle$ reduces to $\mathtt{I}$.
- $\boxed{\text{VALUE}}$-meaningfulness := potential valuability **(Paolini-RonchiDellaRocca'99)**.
  A term t is **potentially valuable** if there is a testing context $\mathtt{T}$ s.t. $\mathtt{T}\langle t\rangle$ reduces to a value.

Meaningfulness for NAME and VALUE

- NAME-meaningfulness := solvability **(Wadsworth'71)**.
  A term t is **solvable** if there is a testing context $\mathtt{T}$ s.t. $\mathtt{T}\langle t\rangle$ reduces to $\mathtt{I}$.
- VALUE-meaningfulness := potential valuability **(Paolini-RonchiDellaRocca'99)**.
  A term t is **potentially valuable** if there is a testing context $\mathtt{T}$ s.t. $\mathtt{T}\langle t\rangle$ reduces to a value.

Example: **P** := **meaningfulness**

Preservation Theorem (Arrial-Guerrieri-K.'24)

- t is NAME-meaningful **if and only if** $t^{\mathtt{N}}$ is BANG-meaningful.
- t is VALUE-meaningful **if and only if** $t^{\mathtt{V}}$ is BANG-meaningful.

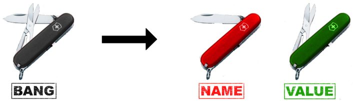
Meaningfulness for NAME and VALUE

- NAME-meaningfulness := solvability **(Wadsworth'71)**.
  A term t is **solvable** if there is a testing context $\mathtt{T}$ s.t. $\mathtt{T}\langle t\rangle$ reduces to $\mathtt{I}$.
- VALUE-meaningfulness := potential valuability **(Paolini-RonchiDellaRocca'99)**.
  A term t is **potentially valuable** if there is a testing context $\mathtt{T}$ s.t. $\mathtt{T}\langle t\rangle$ reduces to a value.

Example: **P** := **meaningfulness**

Preservation Theorem (Arrial-Guerrieri-K.'24)

- t is NAME-meaningful **if and only if** $t^{\mathtt{N}}$ is BANG-meaningful.
- t is VALUE-meaningful **if and only if** $t^{\mathtt{V}}$ is BANG-meaningful.



Agenda

Conclusion

- **BANG** is a **robust** subsuming **untyped** framework capturing:
  - Confluence, Factorization
  - Meaningfulness, Genericity

Conclusion

- **BANG** is a **robust** subsuming **untyped** framework capturing:
  - Confluence, Factorization
  - Meaningfulness, Genericity
- **BANG** is a **robust** subsuming **typed** framework capturing:
  - Upper Bounds for **Length** + **Size**.
  - Exact Measures for both **Length** and **Size**.
  - Decidable Inhabitation Problem (and related algorithms).

Conclusion

- **BANG** is a **robust** subsuming **untyped** framework capturing:
  - Confluence, Factorization
  - Meaningfulness, Genericity
- **BANG** is a **robust** subsuming **typed** framework capturing:
  - Upper Bounds for **Length** + **Size**.
  - Exact Measures for both **Length** and **Size**.
  - Decidable Inhabitation Problem (and related algorithms).

Some further related questions:

Conclusion

- **BANG** is a **robust** subsuming **untyped** framework capturing:
  - Confluence, Factorization
  - Meaningfulness, Genericity
- **BANG** is a **robust** subsuming **typed** framework capturing:
  - Upper Bounds for **Length** + **Size**.
  - Exact Measures for both **Length** and **Size**.
  - Decidable Inhabitation Problem (and related algorithms).

Some further related questions:

- Subsuming other properties of $\boxed{\text{NAME}}$ and $\boxed{\text{VALUE}}$, e.g. the **theory of approximation**

Conclusion

- **BANG** is a **robust** subsuming **untyped** framework capturing:
  - Confluence, Factorization
  - Meaningfulness, Genericity
- **BANG** is a **robust** subsuming **typed** framework capturing:
  - Upper Bounds for **Length** + **Size**.
  - Exact Measures for both **Length** and **Size**.
  - Decidable Inhabitation Problem (and related algorithms).

Some further related questions:

- Subsuming other properties of **NAME** and **VALUE**, e.g. the **theory of approximation**
- Extending **BANG** to more **expressive features**: effectful computation (global state, exceptions, continuations, non-determinism, etc)

Conclusion

- **BANG** is a **robust** subsuming **untyped** framework capturing:
  - Confluence, Factorization
  - Meaningfulness, Genericity
- **BANG** is a **robust** subsuming **typed** framework capturing:
  - Upper Bounds for **Length** + **Size**.
  - Exact Measures for both **Length** and **Size**.
  - Decidable Inhabitation Problem (and related algorithms).

Some further related questions:

- Subsuming other properties of **NAME** and **VALUE**, e.g. the **theory of approximation**
- Extending **BANG** to more **expressive features**: effectful computation (global state, exceptions, continuations, non-determinism, etc)
- Alternative subsuming frameworks capturing also **NEED**


THANK
YOU
FOR
YOUR
ATTENTION


Time for
Questions