# (Sharing in Linear Logic) and Call-by-Need

or

# Sharing in (Linear Logic and Call-by-Need)

Work **in progress** with Eduardo Bonelli

December 7, 2023

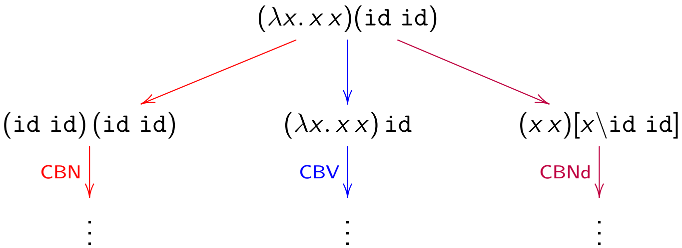
Pablo Barenbaum

Universidad Nacional de Quilmes (CONICET), Argentina

Universidad de Buenos Aires, Argentina

# Outline

Call-by-Name Call-by-Value Call-by-Need



Call-by-Name — Call-by-Value — Call-by-Need

$$(\lambda x.\, x\, x)(\mathtt{id}\ \mathtt{id})$$

$$(\mathtt{id}\ \mathtt{id})\,(\mathtt{id}\ \mathtt{id}) \xrightarrow{\text{CBN}} \vdots \qquad (\lambda x.\, x\, x)\,\mathtt{id} \xrightarrow{\text{CBV}} \vdots \qquad (x\, x)[x \backslash \mathtt{id}\ \mathtt{id}] \xrightarrow{\text{CBNd}} \vdots$$

| | **Duplicates** | **Erases** |
|---|---|---|
| CBN | arbitrary terms | arbitrary terms |
| CBV | values | values |
| CBNd | values | arbitrary terms |

# Unifying frameworks

Languages that subsume other languages as special cases.

# Unifying frameworks

Languages that subsume other languages as special cases.

## Examples

- The parametric $\lambda$-calculus — Della Rocca & Paolini
- Call-by-push-value — Levy
- The Bang Calculus — Ehrhard & Guerrieri

# Unifying frameworks

Languages that subsume other languages as special cases.

## Examples

- The parametric $\lambda$-calculus — `Della Rocca & Paolini`
- Call-by-push-value — `Levy`
- The Bang Calculus — `Ehrhard & Guerrieri`

## Motivations

- Explanation of operational mechanisms through simpler primitives.

# Unifying frameworks

Languages that subsume other languages as special cases.

## Examples

- The parametric $\lambda$-calculus — Della Rocca & Paolini
- Call-by-push-value — Levy
- The Bang Calculus — Ehrhard & Guerrieri

## Motivations

- Explanation of operational mechanisms through simpler primitives.
- Logical justification for operational mechanisms.

# Unifying frameworks

Languages that subsume other languages as special cases.

## Examples

- The parametric $\lambda$-calculus — Della Rocca & Paolini
- Call-by-push-value — Levy
- The Bang Calculus — Ehrhard & Guerrieri

## Motivations

- Explanation of operational mechanisms through simpler primitives.
- Logical justification for operational mechanisms.
- Tackle questions in a uniform and methodical way.

*What is the right notion of strong CBNd?*
*What is the right notion of classical CBNd?*
*What is the right quantitative type system for strong CBNd?*

# Unifying frameworks

Languages that subsume other languages as special cases.

## Examples

- The parametric $\lambda$-calculus — Della Rocca & Paolini
- Call-by-push-value — Levy
- The Bang Calculus — Ehrhard & Guerrieri

## Motivations

- Explanation of operational mechanisms through simpler primitives.
- Logical justification for operational mechanisms.
- Tackle questions in a uniform and methodical way.

  *What is the right notion of strong CBNd?*

  *What is the right notion of classical CBNd?*

  *What is the right quantitative type system for strong CBNd?*
- Generalize the metatheory to prove theorems only once.

# Unifying frameworks

Languages that subsume other languages as special cases.

## Examples

- The parametric $\lambda$-calculus — Della Rocca & Paolini
- Call-by-push-value — Levy
- The Bang Calculus — Ehrhard & Guerrieri

## Motivations

- Explanation of operational mechanisms through simpler primitives.
- Logical justification for operational mechanisms.
- Tackle questions in a uniform and methodical way.

  *What is the right notion of strong CBNd?*
  *What is the right notion of classical CBNd?*
  *What is the right quantitative type system for strong CBNd?*
- Generalize the metatheory to prove theorems only once.

The frameworks above subsume CBN and CBV but not CBNd.

# Linear logic and reduction strategies

Embeddings of **intuitionistic** into **linear** logic correspond to notions of reduction.

`Folklore; made precise by Mackie (1994), Wadler` *et al.* `(1995)`

# Linear logic and reduction strategies

Embeddings of **intuitionistic** into **linear** logic correspond to notions of reduction.

Folklore; made precise by Mackie (1994), Wadler *et al.* (1995)

## Girard's "standard" translation

$$(A \to B)^{\mathsf{N}} := (!A^{\mathsf{N}}) \multimap B^{\mathsf{N}}$$

# Linear logic and reduction strategies

Embeddings of **intuitionistic** into **linear** logic correspond to notions of reduction.

Folklore; made precise by Mackie (1994), Wadler *et al.* (1995)

## Girard's "standard" translation

$$(A \to B)^{\mathsf{N}} := (!A^{\mathsf{N}}) \multimap B^{\mathsf{N}}$$

Sound and complete for CBN.

# Linear logic and reduction strategies

Embeddings of **intuitionistic** into **linear** logic correspond to notions of reduction.

`Folklore; made precise by Mackie (1994), Wadler` *`et al.`* `(1995)`

## Girard's "standard" translation

$$(A \to B)^{\mathsf{N}} := (!A^{\mathsf{N}}) \multimap B^{\mathsf{N}}$$

Sound and complete for CBN. $\quad t \to_{\mathsf{N}}^{*} s \quad \text{iff} \quad t^{\mathsf{N}} \to_{\mathtt{Lin}}^{*} s^{\mathsf{N}}$

# Linear logic and reduction strategies

Embeddings of **intuitionistic** into **linear** logic correspond to notions of reduction.

`Folklore; made precise by Mackie (1994), Wadler et al. (1995)`

## Girard's "standard" translation

$$(A \to B)^{\mathsf{N}} \;:=\; (!A^{\mathsf{N}}) \multimap B^{\mathsf{N}}$$

Sound and complete for CBN. $\quad t \to^{*}_{\mathsf{N}} s \quad \text{iff} \quad t^{\mathsf{N}} \to^{*}_{\mathtt{Lin}} s^{\mathsf{N}}$

## Girard's "boring" translation

$$(A \to B)^{\mathsf{V}} \;:=\; !(A^{\mathsf{V}} \multimap B^{\mathsf{V}})$$

Sound (but not complete) for CBV.

# Linear logic and reduction strategies

Embeddings of **intuitionistic** into **linear** logic correspond to notions of reduction.

`Folklore; made precise by Mackie (1994), Wadler *et al.* (1995)`

## Girard's "standard" translation

$$(A \to B)^{\mathsf{N}} := (!A^{\mathsf{N}}) \multimap B^{\mathsf{N}}$$

Sound and complete for CBN. $t \to_{\mathsf{N}}^{*} s$ iff $t^{\mathsf{N}} \to_{\mathtt{Lin}}^{*} s^{\mathsf{N}}$

## Girard's "boring" translation

$$(A \to B)^{\mathsf{V}} := !(A^{\mathsf{V}} \multimap B^{\mathsf{V}})$$

Sound (but not complete) for CBV. $t \to_{\mathsf{V}}^{*} s$ implies $t^{\mathsf{V}} \to_{\mathtt{Lin}}^{*} s^{\mathsf{V}}$

# Linear logic and reduction strategies

Embeddings of **intuitionistic** into **linear** logic correspond to notions of reduction.

Folklore; made precise by Mackie (1994), Wadler *et al.* (1995)

## Girard's "standard" translation

$$(A \to B)^{\mathsf{N}} \;:=\; (!A^{\mathsf{N}}) \multimap B^{\mathsf{N}}$$

Sound and complete for CBN. $\quad t \to_{\mathsf{N}}^{*} s \quad$ iff $\quad t^{\mathsf{N}} \to_{\mathtt{Lin}}^{*} s^{\mathsf{N}}$

## Girard's "boring" translation

$$(A \to B)^{\mathsf{V}} \;:=\; !(A^{\mathsf{V}} \multimap B^{\mathsf{V}})$$

Sound (but not complete) for CBV. $\quad t \to_{\mathsf{V}}^{*} s \quad$ implies $\quad t^{\mathsf{V}} \to_{\mathtt{Lin}}^{*} s^{\mathsf{V}}$

Completeness fails: $(\mathtt{id}\, t\, s)^{\mathsf{V}} \to_{\mathtt{Lin}}^{*} (t\, s)^{\mathsf{V}}$.

### A call-by-need translation

Wadler *et al.* also studied a CBNd translation:

$$(A \to B)^{\mathsf{Nd}} := !(A^{\mathsf{Nd}} \multimap B^{\mathsf{Nd}})$$

# Linear logic and reduction strategies

## A call-by-need translation

Wadler *et al.* also studied a CBNd translation:

$$(A \to B)^{\mathsf{Nd}} := !(A^{\mathsf{Nd}} \multimap B^{\mathsf{Nd}})$$

**Same as the CBV translation**

## A call-by-need translation

Wadler *et al.* also studied a CBNd translation:

$$(A \to B)^{\mathsf{Nd}} \; := \; !(A^{\mathsf{Nd}} \multimap B^{\mathsf{Nd}})$$

**Same as the CBV translation**

But the target language is an *affine* rather than a *linear* $\lambda$-calculus.

## A call-by-need translation

Wadler *et al.* also studied a CBNd translation:

$$(A \to B)^{\mathsf{Nd}} \; := \, !(A^{\mathsf{Nd}} \multimap B^{\mathsf{Nd}})$$

**Same as the CBV translation**

But the target language is an *affine* rather than a *linear* $\lambda$-calculus.

Arbitrary terms may be erased, even if they are not "values" yet.

## A call-by-need translation

Wadler *et al.* also studied a CBNd translation:

$$(A \to B)^{\mathsf{Nd}} := !(A^{\mathsf{Nd}} \multimap B^{\mathsf{Nd}})$$

**Same as the CBV translation**

But the target language is an *affine* rather than a *linear* $\lambda$-calculus.

Arbitrary terms may be erased, even if they are not "values" yet.

Sound (but not complete) for CBNd.

## A call-by-need translation

Wadler *et al.* also studied a CBNd translation:

$$(A \to B)^{\mathsf{Nd}} := !(A^{\mathsf{Nd}} \multimap B^{\mathsf{Nd}})$$

**Same as the CBV translation**

But the target language is an *affine* rather than a *linear* $\lambda$-calculus.

Arbitrary terms may be erased, even if they are not "values" yet.

Sound (but not complete) for CBNd.

$$t \to^*_{\mathsf{Nd}} s \quad \text{implies} \quad t^{\mathsf{Nd}} \to^*_{\mathtt{Aff}} s^{\mathsf{Nd}}$$

# Goal of this work

## Diagnosis

The exponential modalities confuse two different notions:

1. Ability to make **copies** of shared subterms.
2. Ability to duplicate and erase **references** to shared subterms.

# Goal of this work

## Diagnosis

The exponential modalities confuse two different notions:

1. Ability to make **copies** of shared subterms.
2. Ability to duplicate and erase **references** to shared subterms.

duplicating a reference $\neq$ copying

# Goal of this work

## Diagnosis

The exponential modalities confuse two different notions:

1. Ability to make **copies** of shared subterms.
2. Ability to duplicate and erase **references** to shared subterms.

duplicating a reference $\neq$ copying

## Goal

1. Refine Linear Logic to distinguish between these two notions.
2. Derive a term calculus unifying CBN, CBV, and CBNd.

# Outline

# MELL

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \parr A \mid !A \mid ?A$$

# MELL

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \parr A \mid\ !A \mid\ ?A$$

$$\frac{}{\vdash A, A^\perp}\ \mathtt{ax} \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta}\ \mathtt{cut}$$

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\ \otimes \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ \parr \qquad \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\ !$$

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ \mathtt{w} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ \mathtt{c} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\ \mathtt{d}$$

# MELL

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \parr A \mid\ !A \mid\ ?A$$

$$\frac{}{\vdash A, A^\perp}\,\texttt{ax} \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta}\,\texttt{cut}$$

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\,\otimes \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\,\parr \qquad \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\,!$$

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A}\,\texttt{w} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\,\texttt{c} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\,\texttt{d}$$

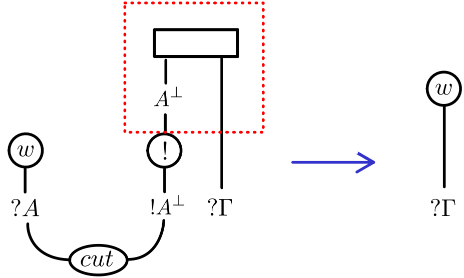
Structural rules have **two** possible computational interpretations.

These interpretations are **not** equivalent if one is interested in sharing.

# Computational interpretation of structural rules

**The cloning interpretation**

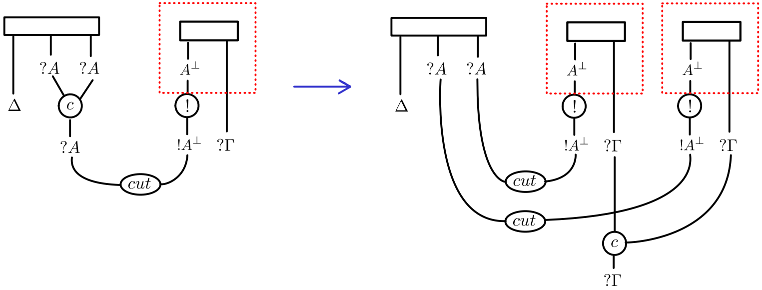
Weakening: erase box



# Computational interpretation of structural rules

**The cloning interpretation**

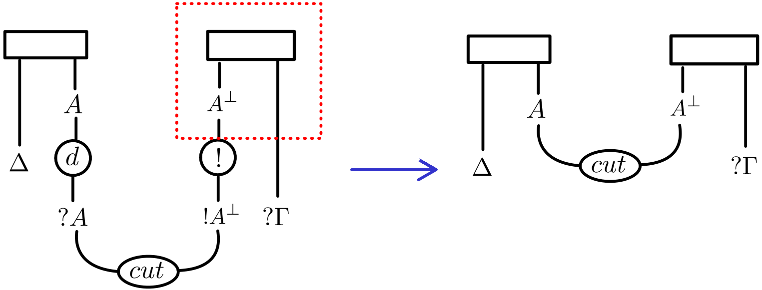
Contraction: duplicate box



# Computational interpretation of structural rules

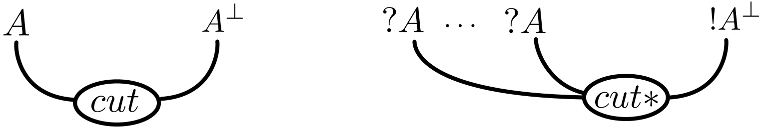
**The cloning interpretation**

Dereliction: unbox



**The sharing interpretation**

Consider a generalized notion of cut:

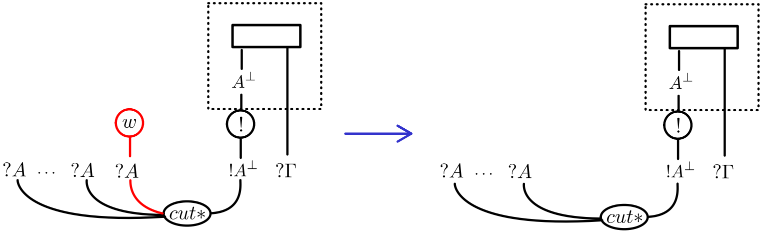



A *cut*$*$ node connects $n \geq 0$ proofs of $?A$ and a **shared** proof of $!A^{\perp}$.

# Computational interpretation of structural rules

**The sharing interpretation**

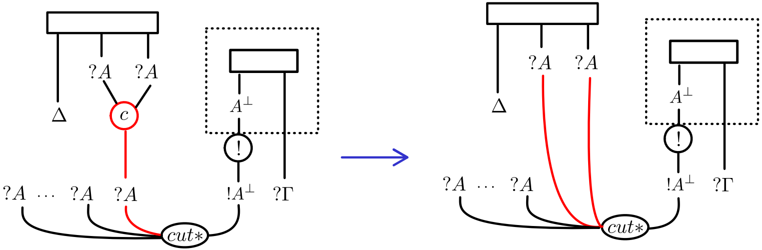
Weakening: erase a reference to a shared box



# Computational interpretation of structural rules

**The sharing interpretation**

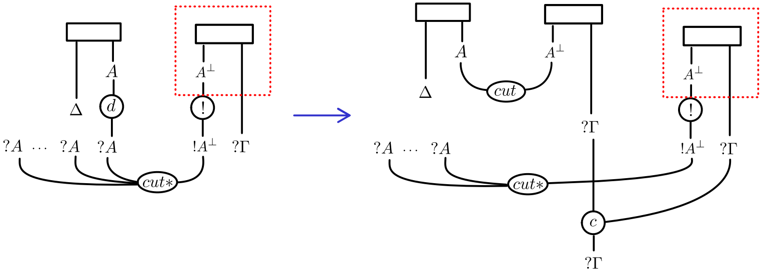
Contraction: duplicate a reference to a shared box



# Computational interpretation of structural rules

**The sharing interpretation**

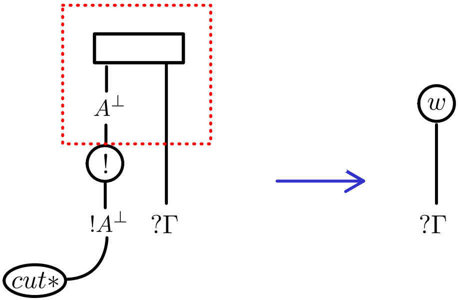
Dereliction: copy box (duplicate & unbox)



# Computational interpretation of structural rules

**The sharing interpretation**

Garbage collection: erase box



# Computational interpretation of structural rules

## Summary

| | Cloning | Sharing |
|---|---|---|
| Weakening | erase box | erase reference |
| Contraction | duplicate box | duplicate reference |
| Dereliction | unbox | copy box (duplicate & unbox) |
| (Garbage collection) | | erase box |

# Computational interpretation of structural rules

## Summary

| | Cloning | Sharing |
|---|---|---|
| Weakening | erase box | erase reference |
| Contraction | duplicate box | duplicate reference |
| Dereliction | unbox | copy box (duplicate & unbox) |
| (Garbage collection) | | erase box |

CBNd cannot copy arbitrary shared subterms.

To understand CBNd:

- We adopt the point of view of sharing.
- To restrict duplication of boxes:
  we consider a variant of MELL with restricted dereliction.

# Outline

# $\text{MELL}_\bullet$

Formulae are extended with two operators $\bullet$ and $\circ$.

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \mathbin{\rotatebox[origin=c]{180}{\&}} A \mid !A \mid ?A \mid \bullet A \mid \circ A$$

$$(\bullet A)^\perp = \circ A^\perp \qquad (\circ A)^\perp = \bullet A^\perp$$

# MELL•

Formulae are extended with two operators • and ∘.

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \parr A \mid !A \mid ?A \mid \bullet A \mid \circ A$$

$$(\bullet A)^{\perp} = \circ A^{\perp} \qquad (\circ A)^{\perp} = \bullet A^{\perp}$$

Two new rules:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \bullet A}\bullet \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, \circ A}\circ$$

# MELL$_\bullet$

Formulae are extended with two operators $\bullet$ and $\circ$.

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \parr A \mid !A \mid ?A \mid \bullet A \mid \circ A$$

$$(\bullet A)^\perp = \circ A^\perp \qquad (\circ A)^\perp = \bullet A^\perp$$

Two new rules:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \bullet A}\bullet \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, \circ A}\circ$$

The dereliction rule is replaced by:

$$\frac{\vdash \Gamma, \circ A}{\vdash \Gamma, ?\circ A}\,\mathtt{d}\circ$$

# MELL$_\bullet$

Formulae are extended with two operators $\bullet$ and $\circ$.

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \parr A \mid !A \mid ?A \mid \bullet A \mid \circ A$$

$$(\bullet A)^\perp = \circ A^\perp \qquad (\circ A)^\perp = \bullet A^\perp$$

Two new rules:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \bullet A}\,\bullet \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, \circ A}\,\circ$$

The dereliction rule is replaced by:

$$\frac{\vdash \Gamma, \circ A}{\vdash \Gamma, ?\circ A}\,\mathtt{d}\circ$$

## Example

$$\vdash A, \circ A^\perp \qquad \vdash A, ?\circ A^\perp \qquad \vdash !A, ?\circ A^\perp$$

# MELL$_\bullet$

Formulae are extended with two operators $\bullet$ and $\circ$.

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \parr A \mid !A \mid ?A \mid \bullet A \mid \circ A$$

$$(\bullet A)^\perp = \circ A^\perp \qquad (\circ A)^\perp = \bullet A^\perp$$

Two new rules:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \bullet A}\bullet \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, \circ A}\circ$$

The dereliction rule is replaced by:

$$\frac{\vdash \Gamma, \circ A}{\vdash \Gamma, ?\circ A}\,\mathtt{d}\circ$$

## Example

$$\vdash A, \circ A^\perp \qquad \vdash A, ?\circ A^\perp \qquad \vdash !A, ?\circ A^\perp \qquad \nvdash A, ?A^\perp \qquad \nvdash !A, ?A^\perp$$

# MELL$_\bullet$

Formulae are extended with two operators $\bullet$ and $\circ$.

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \parr A \mid !A \mid ?A \mid \bullet A \mid \circ A$$

$$(\bullet A)^\perp = \circ A^\perp \qquad (\circ A)^\perp = \bullet A^\perp$$

Two new rules:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \bullet A}\,\bullet \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, \circ A}\,\circ$$

The dereliction rule is replaced by:

$$\frac{\vdash \Gamma, \circ A}{\vdash \Gamma, ?\circ A}\,\mathtt{d}\circ$$

## Example

$$\vdash A, \circ A^\perp \qquad \vdash A, ?\circ A^\perp \qquad \vdash !A, ?\circ A^\perp \qquad \nvdash A, ?A^\perp \qquad \nvdash !A, ?A^\perp$$

$$A \multimapboth \bullet A \qquad \text{but} \qquad !A \not\multimapboth !\bullet A$$

# MELL$_\bullet$

Formulae are extended with two operators $\bullet$ and $\circ$.

$$A ::= \alpha \mid \overline{\alpha} \mid A \otimes A \mid A \parr A \mid\ !A \mid\ ?A \mid \bullet A \mid \circ A$$

$$(\bullet A)^\perp = \circ A^\perp \qquad (\circ A)^\perp = \bullet A^\perp$$

Two new rules:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \bullet A}\,\bullet \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, \circ A}\,\circ$$

The dereliction rule is replaced by:

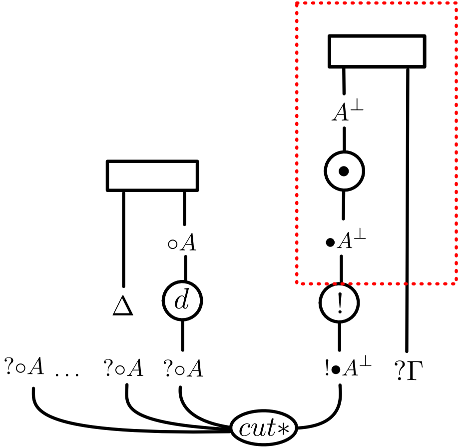
$$\frac{\vdash \Gamma, \circ A}{\vdash \Gamma, ?\circ A}\,\mathtt{d}\circ$$

## Example

$$\vdash A, \circ A^\perp \qquad \vdash A, ?\circ A^\perp \qquad \vdash\ !A, ?\circ A^\perp \qquad \nvdash A, ?A^\perp \qquad \nvdash\ !A, ?A^\perp$$

$A \multimapboth \bullet A$ but $!A \not\multimapboth\ !\bullet A$ **! and ? are not monotonic**

**Intuition:** a box may be copied only if there is a • node immediately next to the promotion node.



**Theorem (Cut elimination)**

For any MELL$_\bullet$ proof there is a cut-free proof of the same conclusion.

# $\mathsf{MELL}_\bullet$

### Theorem (Cut elimination)

For any $\mathsf{MELL}_\bullet$ proof there is a cut-free proof of the same conclusion.

*Proof.*

1. Extend $\mathsf{MELL}_\bullet$ with the (admissible) `cut*` rule.
2. Prove `cut` + `cut*` elimination.

# $\mathsf{MELL}_\bullet$

**Theorem (Conservative extension)**

$\vdash \Gamma$ holds in MELL if and only if $\vdash \Gamma^\bullet$ holds in $\mathsf{MELL}_\bullet$

where $\Gamma^\bullet := \Gamma\{! \mapsto !\bullet, ? \mapsto ?\circ\}$.

# MELL$_\bullet$

## Theorem (Conservative extension)

$\vdash \Gamma$ holds in MELL if and only if $\vdash \Gamma^\bullet$ holds in MELL$_\bullet$

where $\Gamma^\bullet := \Gamma\{! \mapsto !\bullet, ? \mapsto ?\circ\}$.

*Proof.*

($\Rightarrow$) The following rules are admissible in MELL$_\bullet$:

$$\frac{\vdash ?\circ\Gamma, A}{\vdash ?\circ\Gamma, !\bullet A}\,!\bullet \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?\circ A}\,?\circ\mathtt{w} \qquad \frac{\vdash \Gamma, ?\circ A, ?\circ A}{\vdash \Gamma, ?\circ A}\,?\circ\mathtt{c} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?\circ A}\,?\circ\mathtt{d}$$

($\Leftarrow$) Any MELL$_\bullet$ proof is valid in MELL erasing $\bullet$ and $\circ$.

# $\mathsf{MELL}_\bullet$

### Theorem (Conservative extension)

$\vdash \Gamma$ holds in MELL if and only if $\vdash \Gamma^\bullet$ holds in $\mathsf{MELL}_\bullet$

where $\Gamma^\bullet := \Gamma\{! \mapsto !\bullet, ? \mapsto ?\circ\}$.

*Proof.*

($\Rightarrow$) The following rules are admissible in $\mathsf{MELL}_\bullet$:

$$\frac{\vdash ?\circ\Gamma, A}{\vdash ?\circ\Gamma, !\bullet A}\,!\bullet \quad \frac{\vdash \Gamma}{\vdash \Gamma, ?\circ A}\,?\circ\mathtt{w} \quad \frac{\vdash \Gamma, ?\circ A, ?\circ A}{\vdash \Gamma, ?\circ A}\,?\circ\mathtt{c} \quad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?\circ A}\,?\circ\mathtt{d}$$

($\Leftarrow$) Any $\mathsf{MELL}_\bullet$ proof is valid in MELL erasing $\bullet$ and $\circ$.

$\mathsf{MELL}_\bullet$ refines MELL.

# Outline

# The linear sharing $\lambda$-calculus

$\mathsf{MELL}_\bullet$ suggests the definition of a term calculus, inspired also by:

1. Many linear $\lambda$-calculi — `Lafont, Wadler, Pfenning, ...`
2. The Linear Substitution Calculus — `Accattoli & Kesner`
3. The Bang Calculus — `Ehrhard & Guerrieri`

# The linear sharing $\lambda$-calculus

$MELL_\bullet$ suggests the definition of a term calculus, inspired also by:

1. Many linear $\lambda$-calculi — Lafont, Wadler, Pfenning, ...
2. The Linear Substitution Calculus — Accattoli & Kesner
3. The Bang Calculus — Ehrhard & Guerrieri

## Syntax of terms

$$
\begin{array}{rcllll}
t & ::= & a & \mid & x & \text{(linear vs. unrestricted variables)} \\
 & \mid & \lambda a.\, t & \mid & t\, s & \text{(abstractions bind linear variables)} \\
 & \mid & \bullet t & \mid & \circ(t) & \\
 & \mid & !t & \mid & t[x \backslash s] & \text{(ESs bind unrestricted variables)}
\end{array}
$$

# The linear sharing $\lambda$-calculus — Type system

## Types and typing judgments

$$A ::= \alpha \mid A \multimap B \mid \bullet A \mid !A \qquad \boxed{\Delta; \Gamma \vdash t : A}$$

**(1)** Types in $\Delta$ are implicitly prefixed by $!\bullet$. **(2)** $\Gamma$ is treated linearly.

# The linear sharing $\lambda$-calculus — Type system

## Types and typing judgments

$$A ::= \alpha \mid A \multimap B \mid \bullet A \mid !A \qquad \boxed{\Delta; \Gamma \vdash t : A}$$

**(1)** Types in $\Delta$ are implicitly prefixed by $!\bullet$. **(2)** $\Gamma$ is treated linearly.

## Typing rules

$$\frac{}{\Delta; a : A \vdash a : A}\ \texttt{ax} \qquad \frac{}{\Delta, x : A; \cdot \vdash x : \bullet A}\ \texttt{der}$$

$$\frac{\Delta; \Gamma, a : A \vdash t : B}{\Delta; \Gamma \vdash \lambda a.\, t : A \multimap B}\ {\multimap}\texttt{i} \qquad \frac{\Delta; \Gamma_1 \vdash t : A \multimap B \qquad \Delta; \Gamma_2 \vdash s : B}{\Delta; \Gamma_1, \Gamma_2 \vdash t\, s : B}\ {\multimap}\texttt{e}$$

$$\frac{\Delta; \Gamma \vdash t : A}{\Delta; \Gamma \vdash \bullet t : \bullet A}\ \bullet\texttt{i} \qquad \frac{\Delta; \Gamma \vdash t : \bullet A}{\Delta; \Gamma \vdash \circ(t) : A}\ \bullet\texttt{e}$$

$$\frac{\Delta; \cdot \vdash t : A}{\Delta; \cdot \vdash !t : !A}\ \texttt{!i} \qquad \frac{\Delta, x : A; \Gamma_1 \vdash t : B \qquad \Delta; \Gamma_2 \vdash s : !\bullet A}{\Delta; \Gamma_1, \Gamma_2 \vdash t[x \backslash s] : B}\ \texttt{!}\bullet\texttt{e}$$

# The linear sharing $\lambda$-calculus — Type system

## Types and typing judgments

$$A ::= \alpha \mid A \multimap B \mid \bullet A \mid !A \qquad \boxed{\Delta; \Gamma \vdash t : A}$$

**(1)** Types in $\Delta$ are implicitly prefixed by $!\bullet$. **(2)** $\Gamma$ is treated linearly.

## Typing rules

$$\frac{}{\Delta; a : A \vdash a : A}\ \texttt{ax} \qquad \frac{}{\Delta, x : A; \cdot \vdash x : \bullet A}\ \texttt{der}$$

$$\frac{\Delta; \Gamma, a : A \vdash t : B}{\Delta; \Gamma \vdash \lambda a.\, t : A \multimap B}\ \multimap\texttt{i} \qquad \frac{\Delta; \Gamma_1 \vdash t : A \multimap B \qquad \Delta; \Gamma_2 \vdash s : B}{\Delta; \Gamma_1, \Gamma_2 \vdash t\, s : B}\ \multimap\texttt{e}$$

$$\frac{\Delta; \Gamma \vdash t : A}{\Delta; \Gamma \vdash \bullet t : \bullet A}\ \bullet\texttt{i} \qquad \frac{\Delta; \Gamma \vdash t : \bullet A}{\Delta; \Gamma \vdash \circ(t) : A}\ \bullet\texttt{e}$$

$$\frac{\Delta; \cdot \vdash t : A}{\Delta; \cdot \vdash !t : !A}\ \texttt{!i} \qquad \frac{\Delta, x : A; \Gamma_1 \vdash t : B \qquad \Delta; \Gamma_2 \vdash s : !\bullet A}{\Delta; \Gamma_1, \Gamma_2 \vdash t[x \backslash s] : B}\ \texttt{!}\bullet\texttt{e}$$

# The linear sharing $\lambda$-calculus — Reduction

## Substitution contexts

$$\mathtt{L} ::= \Box \mid \mathtt{L}[x \backslash t] \qquad t\mathtt{L} \text{ plugs } t \text{ in } \mathtt{L}$$

# The linear sharing $\lambda$-calculus — Reduction

## Substitution contexts

$$\mathtt{L} ::= \Box \mid \mathtt{L}[x\backslash t] \qquad t\mathtt{L} \text{ plugs } t \text{ in } \mathtt{L}$$

## Reduction rules (without "L" contexts)

$$
\begin{array}{rll}
(\lambda a.\, t)\, s & \to_{\bullet\mathsf{db}} & t\{a\backslash s\} \\
\mathtt{o}(\bullet t) & \to_{\bullet\mathsf{open}} & t \\
\mathtt{C}\langle x\rangle[x\backslash !\bullet t] & \to_{\bullet\mathsf{ls}} & \mathtt{C}\langle \bullet t\rangle[x\backslash !\bullet t] \\
t[x\backslash !s] & \to_{\bullet\mathsf{gc}} & t \qquad \text{if } x \notin \mathsf{fv}(t)
\end{array}
$$

# The linear sharing $\lambda$-calculus — Reduction

## Substitution contexts

$$\mathtt{L} ::= \Box \mid \mathtt{L}[x \backslash t] \qquad t\mathtt{L} \text{ plugs } t \text{ in } \mathtt{L}$$

## Reduction rules

$$\begin{array}{rll} (\lambda a.\, t)\mathtt{L}\, s & \to_{\bullet\mathsf{db}} & t\{a \backslash s\}\mathtt{L} \\ \mathtt{o}((\bullet t)\mathtt{L}) & \to_{\bullet\mathsf{open}} & t\mathtt{L} \\ \mathtt{C}\langle x \rangle[x \backslash (!(\bullet t)\mathtt{L}_1)\mathtt{L}_2] & \to_{\bullet\mathsf{ls}} & \mathtt{C}\langle(\bullet t)\mathtt{L}_1\rangle[x \backslash !(\bullet t)\mathtt{L}_1]\mathtt{L}_2 \\ t[x \backslash (!s)\mathtt{L}] & \to_{\bullet\mathsf{gc}} & t\mathtt{L} \qquad \text{if } x \notin \mathsf{fv}(t) \end{array}$$

# The linear sharing $\lambda$-calculus — Reduction

## Substitution contexts

$$\mathtt{L} ::= \Box \mid \mathtt{L}[x \backslash t] \qquad t\mathtt{L} \text{ plugs } t \text{ in } \mathtt{L}$$

## Reduction rules

$$\begin{array}{rll}
(\lambda a.\, t)\mathtt{L}\, s & \to_{\bullet \mathsf{db}} & t\{a \backslash s\}\mathtt{L} \\
\mathsf{o}((\bullet t)\mathtt{L}) & \to_{\bullet \mathsf{open}} & t\mathtt{L} \\
\mathtt{C}\langle x \rangle [x \backslash (!(\bullet t)\mathtt{L}_1)\mathtt{L}_2] & \to_{\bullet \mathsf{ls}} & \mathtt{C}\langle (\bullet t)\mathtt{L}_1 \rangle [x \backslash !(\bullet t)\mathtt{L}_1]\mathtt{L}_2 \\
t[x \backslash (!s)\mathtt{L}] & \to_{\bullet \mathsf{gc}} & t\mathtt{L} \qquad \text{if } x \notin \mathsf{fv}(t)
\end{array}$$

## Example

$$\mathsf{o}(x[x \backslash !\bullet y]) \to_{\bullet \mathsf{ls}} \mathsf{o}((\bullet y)[x \backslash !\bullet y]) \to_{\bullet \mathsf{open}} y[x \backslash !\bullet y] \to_{\bullet \mathsf{gc}} y$$

# The linear sharing $\lambda$-calculus — Reduction

## Substitution contexts

$$\mathtt{L} ::= \Box \mid \mathtt{L}[x \backslash t] \qquad t\mathtt{L} \text{ plugs } t \text{ in } \mathtt{L}$$

## Reduction rules

$$
\begin{array}{rcll}
(\lambda a.\, t)\mathtt{L}\, s & \rightarrow_{\bullet\mathsf{db}} & t\{a \backslash s\}\mathtt{L} & \\
\mathtt{o}((\bullet t)\mathtt{L}) & \rightarrow_{\bullet\mathsf{open}} & t\mathtt{L} & \\
\mathtt{C}\langle x \rangle[x \backslash (!(\bullet t)\mathtt{L}_1)\mathtt{L}_2] & \rightarrow_{\bullet\mathsf{ls}} & \mathtt{C}\langle (\bullet t)\mathtt{L}_1 \rangle[x \backslash !(\bullet t)\mathtt{L}_1]\mathtt{L}_2 & \\
t[x \backslash (!s)\mathtt{L}] & \rightarrow_{\bullet\mathsf{gc}} & t\mathtt{L} & \text{if } x \notin \mathsf{fv}(t)
\end{array}
$$

## Example

$$\mathtt{o}(x[x \backslash !\bullet y]) \rightarrow_{\bullet\mathsf{ls}} \mathtt{o}((\bullet y)[x \backslash !\bullet y]) \rightarrow_{\bullet\mathsf{open}} y[x \backslash !\bullet y] \rightarrow_{\bullet\mathsf{gc}} y$$

$z[x \backslash y]$ and $x[x \backslash !y]$ are normal forms

# Basic properties

**Proposition (Soundness wrt $\mathsf{MELL}_\bullet$)**

If $\Delta; \Gamma \vdash t : A$ then $\vdash ?\circ\Delta^\perp, \Gamma^\perp, A$ in $\mathsf{MELL}_\bullet$.

# Basic properties

### Proposition (Soundness wrt MELL$_\bullet$)

If $\Delta; \Gamma \vdash t : A$ then $\vdash ?\circ\Delta^\perp, \Gamma^\perp, A$ in MELL$_\bullet$.

### Proposition (Subject reduction)

If $\Delta; \Gamma \vdash t : A$ and $t \to s$ then $\Delta; \Gamma \vdash s : A$.

# Confluence

Harder than we expected.

# Confluence

Harder than we expected.

## First attempt

Apply Tait–Martin-Löf's technique. $(\to \subseteq \Rightarrow \subseteq \to^*) + \Diamond(\Rightarrow)$
Defining an inductive notion of simulatenous reduction $\Rightarrow$ is very difficult.

# Confluence

Harder than we expected.

## First attempt

Apply Tait–Martin-Löf's technique. $(\rightarrow \subseteq \Rightarrow \subseteq \rightarrow^*) + \Diamond(\Rightarrow)$

Defining an inductive notion of simulatenous reduction $\Rightarrow$ is very difficult.

## Second attempt

Use techniques based on residuals (Lévy, Huet, Melliès). FD + PERM

PERM fails:

$$\begin{array}{ccc}
z[x\backslash !y][y\backslash (!\bullet t)\mathtt{L}] & \xrightarrow{\bullet\mathsf{ls}} & z[x\backslash !\bullet t][y\backslash !\bullet t]\mathtt{L} \\
{\scriptstyle \bullet\mathsf{gc}}\downarrow & & {\scriptstyle \bullet\mathsf{gc}}\downarrow \\
z[y\backslash (!\bullet t)\mathtt{L}] & & z[y\backslash !\bullet t]\mathtt{L}
\end{array}$$

# Confluence

## Definition (Structural equivalence)

Structural equivalence is the congruence generated by:

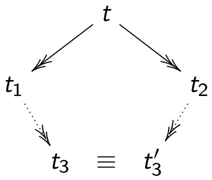
$$t[x\backslash s[y\backslash r]] \equiv t[x\backslash s][y\backslash r]$$

## Lemma (Strong bisimulation)

$(\equiv \rightarrow) \subseteq (\rightarrow \equiv)$

## Theorem (Confluence)

$\lambda^\bullet$ is CR up to $\equiv$: FD + PERM



# Normalization of typable terms

## The Linear Substitution Calculus (LSC)

$$(\lambda x.\, t)\mathtt{L}\, s \rightarrow_{\mathsf{db}} t[x\backslash s]\mathtt{L} \qquad \mathtt{C}\langle x\rangle[x\backslash t] \rightarrow_{\mathsf{ls}} \mathtt{C}\langle t\rangle[x\backslash t]$$

$$t[x\backslash s] \rightarrow_{\mathsf{gc}} t \qquad (x \notin \mathsf{fv}(t))$$

# Normalization of typable terms

## The Linear Substitution Calculus (LSC)

$$(\lambda x.\, t)\mathtt{L}\, s \to_{\mathsf{db}} t[x\backslash s]\mathtt{L} \qquad \mathtt{C}\langle x\rangle[x\backslash t] \to_{\mathsf{ls}} \mathtt{C}\langle t\rangle[x\backslash t]$$

$$t[x\backslash s] \to_{\mathsf{gc}} t \qquad (x \notin \mathsf{fv}(t))$$

**Fact.** Simply typed LSC is SN. (db-expansion + SN of STLC + PSN)

# Normalization of typable terms

## The Linear Substitution Calculus (LSC)

$$(\lambda x.\, t)\mathtt{L}\, s \to_{\mathsf{db}} t[x\backslash s]\mathtt{L} \qquad \mathtt{C}\langle x\rangle[x\backslash t] \to_{\mathsf{ls}} \mathtt{C}\langle t\rangle[x\backslash t]$$

$$t[x\backslash s] \to_{\mathsf{gc}} t \qquad (x \notin \mathsf{fv}(t))$$

**Fact.** Simply typed LSC is SN. (db-expansion + SN of STLC + PSN)
**Idea.** From $t_1 \to_\bullet t_2 \to_\bullet \ldots$ obtain $[\![t_1]\!] \to_{\mathsf{lsc}} [\![t_2]\!] \to_{\mathsf{lsc}} \ldots$.

# Normalization of typable terms

## The Linear Substitution Calculus (LSC)

$$(\lambda x.\, t)\mathtt{L}\, s \rightarrow_{\mathsf{db}} t[x\backslash s]\mathtt{L} \qquad \mathtt{C}\langle x\rangle[x\backslash t] \rightarrow_{\mathsf{ls}} \mathtt{C}\langle t\rangle[x\backslash t]$$

$$t[x\backslash s] \rightarrow_{\mathsf{gc}} t \qquad (x \notin \mathsf{fv}(t))$$

**Fact.** Simply typed LSC is SN. (db-expansion + SN of STLC + PSN)
**Idea.** From $t_1 \rightarrow_\bullet t_2 \rightarrow_\bullet \ldots$ obtain $[\![t_1]\!] \rightarrow_{\mathsf{lsc}} [\![t_2]\!] \rightarrow_{\mathsf{lsc}} \ldots$.

Problem: mismatch between $\rightarrow_\bullet$ and $\rightarrow_{\mathsf{lsc}}$ reduction

$$\begin{array}{lll} x[x\backslash(!\bullet t)\mathtt{L}] & \rightarrow_\bullet & (\bullet t)[x\backslash !\bullet t]\mathtt{L} \\ x[x\backslash t\mathtt{L}] & \rightarrow_{\mathsf{lsc}} & (t\mathtt{L})[x\backslash t\mathtt{L}] \end{array}$$

# Normalization of typable terms

## The Linear Substitution Calculus (LSC)

$$(\lambda x.\, t)\mathtt{L}\, s \rightarrow_{\mathsf{db}} t[x\backslash s]\mathtt{L} \qquad \mathtt{C}\langle x\rangle[x\backslash t] \rightarrow_{\mathsf{ls}} \mathtt{C}\langle t\rangle[x\backslash t]$$

$$t[x\backslash s] \rightarrow_{\mathsf{gc}} t \qquad (x \notin \mathsf{fv}(t))$$

**Fact.** Simply typed LSC is SN. (db-expansion + SN of STLC + PSN)
**Idea.** From $t_1 \rightarrow_\bullet t_2 \rightarrow_\bullet \ldots$ obtain $[\![t_1]\!] \rightarrow_{\mathsf{lsc}} [\![t_2]\!] \rightarrow_{\mathsf{lsc}} \ldots$.

Problem: mismatch between $\rightarrow_\bullet$ and $\rightarrow_{\mathsf{lsc}}$ reduction

$$\begin{array}{lll} x[x\backslash(!\bullet t)\mathtt{L}] & \rightarrow_\bullet & (\bullet t)[x\backslash !\bullet t]\mathtt{L} \\ x[x\backslash t\mathtt{L}] & \rightarrow_{\mathsf{lsc}} & (t\mathtt{L})[x\backslash t\mathtt{L}] \quad \Rrightarrow \quad t[x\backslash t]\mathtt{L} \end{array}$$

# Normalization of typable terms

## The Linear Substitution Calculus (LSC)

$$(\lambda x.\, t)\mathtt{L}\, s \to_{\mathsf{db}} t[x\backslash s]\mathtt{L} \qquad \mathtt{C}\langle x\rangle[x\backslash t] \to_{\mathsf{ls}} \mathtt{C}\langle t\rangle[x\backslash t]$$

$$t[x\backslash s] \to_{\mathsf{gc}} t \qquad (x \notin \mathsf{fv}(t))$$

**Fact.** Simply typed LSC is SN. (db-expansion + SN of STLC + PSN)
**Idea.** From $t_1 \to_\bullet t_2 \to_\bullet \ldots$ obtain $[\![t_1]\!] \to_{\mathsf{lsc}} [\![t_2]\!] \to_{\mathsf{lsc}} \ldots$.

Problem: mismatch between $\to_\bullet$ and $\to_{\mathsf{lsc}}$ reduction

$$\begin{array}{lll} x[x\backslash(!\bullet t)\mathtt{L}] & \to_\bullet & (\bullet t)[x\backslash !\bullet t]\mathtt{L} \\ x[x\backslash t\mathtt{L}] & \to_{\mathsf{lsc}} & (t\mathtt{L})[x\backslash t\mathtt{L}] \quad \Rrightarrow \quad t[x\backslash t]\mathtt{L} \end{array}$$

## Definition (Fusion)

$$\begin{array}{rcl} t[y\backslash s][x\backslash s] & \Rrightarrow & t\{y\backslash x\}[x\backslash s] \\ \mathtt{C}\langle t[x\backslash s]\rangle & \Rrightarrow & \mathtt{C}\langle t\rangle[x\backslash s] \end{array}$$

## Lemma (Postponement)

$(\Rrightarrow\to_{\mathsf{lsc}}) \subseteq (\to_{\mathsf{lsc}}\Rrightarrow)$

# Normalization of typable terms

### Theorem (Strong normalization)

The typed $\lambda^{\bullet}$-calculus is SN.

*Proof.*

# Normalization of typable terms

## Theorem (Strong normalization)

The typed $\lambda^\bullet$-calculus is SN.

*Proof.* Translate typable $\lambda^\bullet$ terms to typable LSC terms:

$$[\![A \multimap B]\!] := [\![A]\!] \to [\![B]\!] \qquad [\![\bullet A]\!] := \bigstar \to [\![A]\!] \qquad [\![!A]\!] := [\![A]\!]$$

$$\begin{array}{rcll} [\![a]\!] & := & a & \\ [\![\lambda a.\, t]\!] & := & \lambda a.\, [\![t]\!] & \\ [\![\bullet\, t]\!] & := & \lambda z.\, [\![t]\!] & (z \text{ fresh}) \\ [\![!t]\!] & := & [\![t]\!] & \end{array} \qquad \begin{array}{rcl} [\![x]\!] & := & x \\ [\![t\, s]\!] & := & [\![t]\!]\, [\![s]\!] \\ [\![\mathsf{o}(t)]\!] & := & [\![t]\!] * \\ [\![t[x \backslash s]]\!] & := & [\![t]\!][x \backslash [\![s]\!]] \end{array}$$

1. Map an infinite $t_1 \to_\bullet t_2 \to_\bullet \ldots$ to $[\![t_1]\!] \to_{\mathsf{lsc}} \Rrightarrow [\![t_2]\!] \to_{\mathsf{lsc}} \Rrightarrow \ldots$.
2. Postpone $\Rrightarrow$ to obtain an infinite reduction sequence in LSC.

(Strictly speaking, we also need to postpone $\to_{\bullet \mathsf{gc}}$).

# Outline

# Notions of reduction

## A zoo of rewriting rules

**Values** $\mathtt{v} ::= \lambda x.\, t$ **Lax values** $\mathtt{v}^{+} ::= x \mid \lambda x.\, t$

$$
\begin{array}{rlll}
(\lambda x.\, t)\mathtt{L}\, s & \to_{\mathsf{db}} & t[x\backslash s]\mathtt{L} & \\
\mathtt{C}\langle x\rangle[x\backslash t] & \to_{\mathsf{ls}} & \mathtt{C}\langle t\rangle[x\backslash t] & \\
\mathtt{C}\langle x\rangle[x\backslash \mathtt{vL}] & \to_{\mathsf{lsv}} & \mathtt{C}\langle \mathtt{v}\rangle[x\backslash \mathtt{v}]\mathtt{L} & \\
\mathtt{C}\langle x\rangle[x\backslash \mathtt{v}^{+}\mathtt{L}] & \to_{\mathsf{lsv+}} & \mathtt{C}\langle \mathtt{v}^{+}\rangle[x\backslash \mathtt{v}^{+}]\mathtt{L} & \\
\mathtt{C}\langle x\rangle[x\backslash \mathtt{vL}] & \to_{\mathsf{lsv}\times} & \mathtt{C}\langle \mathtt{vL}\rangle[x\backslash \mathtt{vL}] & \\
t[x\backslash s] & \to_{\mathsf{gc}} & t & \text{if } x \notin \mathsf{fv}(t) \\
t[x\backslash \mathtt{v}^{+}\mathtt{L}] & \to_{\mathsf{gcv+}} & t\mathtt{L} & \text{if } x \notin \mathsf{fv}(t)
\end{array}
$$

# Notions of reduction

## A zoo of rewriting rules

**Values** $\mathtt{v} ::= \lambda x.\, t$ **Lax values** $\mathtt{v}^{+} ::= x \mid \lambda x.\, t$

$$
\begin{array}{rll l}
(\lambda x.\, t)\mathtt{L}\, s & \to_{\mathsf{db}} & t[x \backslash s]\mathtt{L} & \\
\mathtt{C}\langle x\rangle[x \backslash t] & \to_{\mathsf{ls}} & \mathtt{C}\langle t\rangle[x \backslash t] & \\
\mathtt{C}\langle x\rangle[x \backslash \mathtt{vL}] & \to_{\mathsf{lsv}} & \mathtt{C}\langle \mathtt{v}\rangle[x \backslash \mathtt{v}]\mathtt{L} & \\
\mathtt{C}\langle x\rangle[x \backslash \mathtt{v}^{+}\mathtt{L}] & \to_{\mathsf{lsv+}} & \mathtt{C}\langle \mathtt{v}^{+}\rangle[x \backslash \mathtt{v}^{+}]\mathtt{L} & \\
\mathtt{C}\langle x\rangle[x \backslash \mathtt{vL}] & \to_{\mathsf{lsv}\times} & \mathtt{C}\langle \mathtt{vL}\rangle[x \backslash \mathtt{vL}] & \\
t[x \backslash s] & \to_{\mathsf{gc}} & t & \text{if } x \notin \mathsf{fv}(t) \\
t[x \backslash \mathtt{v}^{+}\mathtt{L}] & \to_{\mathsf{gcv+}} & t\mathtt{L} & \text{if } x \notin \mathsf{fv}(t)
\end{array}
$$

## Definition: notions of CBN, CBV, and CBNd

$$\to_{\mathsf{N}} := \to_{\mathsf{db}} \cup \to_{\mathsf{ls}} \cup \to_{\mathsf{gc}}$$

$$\to_{\mathsf{V}} := \to_{\mathsf{db}} \cup \to_{\mathsf{lsv}} \cup \to_{\mathsf{gcv+}} \qquad \to_{\mathsf{V+}} := \to_{\mathsf{db}} \cup \to_{\mathsf{lsv+}} \cup \to_{\mathsf{gcv+}}$$

$$\to_{\mathsf{Nd}} := \to_{\mathsf{db}} \cup \to_{\mathsf{lsv}} \cup \to_{\mathsf{gc}} \qquad \to_{\mathsf{Nd}\times} := \to_{\mathsf{db}} \cup \to_{\mathsf{lsv}\times} \cup \to_{\mathsf{gc}}$$

# Embedding CBN, CBV, CBNd

| | CBN | CBV⁺ | CBV | CBNd$^{\times}$ | CBNd |
|---|---|---|---|---|---|
| $A \to B$ | $!\bullet A \multimap B$ | $!\bullet A \multimap !\bullet B$ | $!\bullet A \multimap !\bullet B$ | $!\bullet A \multimap \bullet B$ | – |

# Embedding CBN, CBV, CBNd

| | CBN | CBV$^+$ | CBV | CBNd$^\times$ | CBNd |
|---|---|---|---|---|---|
| $A \to B$ | $!\bullet A \multimap B$ | $!\bullet A \multimap !\bullet B$ | $!\bullet A \multimap !\bullet B$ | $!\bullet A \multimap \bullet B$ | – |
| $x$ | $\circ(x)$ | $!\bullet\circ(x)$ | $!x$ | $x$ | – |
| $\lambda x.\, t$ | $\lambda a.\, t[x\backslash a]$ | $!\bullet\lambda a.\, t[x\backslash a]$ | $!\bullet\lambda a.\, t[x\backslash a]$ | $\bullet\lambda a.\, t[x\backslash a]$ | – |
| $t\ s$ | $t\,(!\bullet s)$ | $\circ(x)[x\backslash t]\, s$ | $\circ(x)[x\backslash t]\, s$ | $\circ(t)\,(!s)$ | – |
| $t[x\backslash s]$ | $t[x\backslash !\bullet s]$ | $t[x\backslash s]$ | $t[x\backslash s]$ | $t[x\backslash !s]$ | – |

# Embedding CBN, CBV, CBNd

| | CBN | CBV⁺ | CBV | CBNd$^\times$ | CBNd |
|---|---|---|---|---|---|
| $A \to B$ | $!\bullet A \multimap B$ | $!\bullet A \multimap !\bullet B$ | $!\bullet A \multimap !\bullet B$ | $!\bullet A \multimap \bullet B$ | – |
| $x$ | $\circ(x)$ | $!\bullet\circ(x)$ | $!x$ | $x$ | – |
| $\lambda x.\, t$ | $\lambda a.\, t[x\backslash a]$ | $!\bullet\lambda a.\, t[x\backslash a]$ | $!\bullet\lambda a.\, t[x\backslash a]$ | $\bullet\lambda a.\, t[x\backslash a]$ | – |
| $t\ s$ | $t\,(!\bullet s)$ | $\circ(x)[x\backslash t]\ s$ | $\circ(x)[x\backslash t]\ s$ | $\circ(t)\,(!s)$ | – |
| $t[x\backslash s]$ | $t[x\backslash !\bullet s]$ | $t[x\backslash s]$ | $t[x\backslash s]$ | $t[x\backslash !s]$ | – |

## Theorem (Soundness/completeness)

The above embeddings are:

| | |
|---|---|
| CBN | Sound and complete for reduction. |
| CBV⁺ | Sound for reduction but **not** complete. |
| CBV | Sound for reduction and complete for equality. |
| CBNd$^\times$ | Sound and complete for reduction. |
| CBNd | (Does not seem possible) |

# Embedding the Bang calculus

## The Bang Calculus

(Bucciarelli *et al.*'s $\lambda!$ with linear subst.)

$$A ::= \alpha \mid !A \mid !A \to B \qquad t ::= x \mid \lambda x.\, t \mid t\, s \mid !t \mid \mathtt{der}(t) \mid t[x\backslash s]$$

$$\begin{array}{rcll}
(\lambda x.\, t)\mathtt{L}\, s & \to_{\mathsf{db}} & t[x\backslash s]\mathtt{L} & \\
\mathtt{C}\langle x\rangle[x\backslash(!s)\mathtt{L}] & \to_{\mathsf{ls!}} & \mathtt{C}\langle s\rangle[x\backslash !s]\mathtt{L} & \\
t[x\backslash(!s)\mathtt{L}] & \to_{\mathsf{gc!}} & t\mathtt{L} & \text{if } x \notin \mathsf{fv}(t) \\
\mathtt{der}((!t)\mathtt{L}) & \to_{\mathsf{d!}} & t\mathtt{L} &
\end{array}$$

# Embedding the Bang calculus

## The Bang Calculus (Bucciarelli *et al.*'s $\lambda!$ with linear subst.)

$$A ::= \alpha \mid !A \mid !A \to B \qquad t ::= x \mid \lambda x.\, t \mid t\, s \mid !t \mid \mathtt{der}(t) \mid t[x\backslash s]$$

$$\begin{array}{rcll} (\lambda x.\, t)\mathtt{L}\, s & \to_{\mathsf{db}} & t[x\backslash s]\mathtt{L} & \\ \mathtt{C}\langle x\rangle[x\backslash(!s)\mathtt{L}] & \to_{\mathsf{ls!}} & \mathtt{C}\langle s\rangle[x\backslash !s]\mathtt{L} & \\ t[x\backslash(!s)\mathtt{L}] & \to_{\mathsf{gc!}} & t\mathtt{L} & \text{if } x \notin \mathsf{fv}(t) \\ \mathtt{der}((!t)\mathtt{L}) & \to_{\mathsf{d!}} & t\mathtt{L} & \end{array}$$

## Theorem

The following embedding is sound and complete for reduction, up to identifying $\mathtt{der}(t) \equiv x[x\backslash t]$:

$$\begin{array}{ccc} !A & \mapsto & !\bullet A \\ !A \to B & \mapsto & !\bullet A \multimap B \\ x & \mapsto & \mathsf{o}(x) \\ \lambda x.\, t & \mapsto & \lambda a.\, t[x\backslash a] \\ t\, s & \mapsto & t\, s \\ !t & \mapsto & !\bullet t \\ t[x\backslash s] & \mapsto & t[x\backslash s] \\ \mathtt{der}(t) & \mapsto & \mathsf{o}(x)[x\backslash t] \end{array}$$

# Strategies

The embeddings above are for calculi. We are also working on embedding strategies.

# Strategies

The embeddings above are for calculi. We are also working on embedding strategies.

## Idea

| | | |
|---|---|---|
| $t[x\backslash s]$ | $\rightsquigarrow$ | evaluate inside $s$ until the outermost !... appears |
| $t[x\backslash !s]$ | $\rightsquigarrow$ | evaluate $t$ until $x$ is needed |
| $(...x...)[x\backslash !s]$ | $\rightsquigarrow$ | evaluate inside $s$ until the outermost •... appears |
| $(...x...)[x\backslash !\bullet s]$ | $\rightsquigarrow$ | perform the substitution |

# Strategies

The embeddings above are for calculi. We are also working on embedding strategies.

## Idea

| | | |
|---|---|---|
| $t[x\backslash s]$ | $\rightsquigarrow$ | evaluate inside $s$ until the outermost !... appears |
| $t[x\backslash !s]$ | $\rightsquigarrow$ | evaluate $t$ until $x$ is needed |
| $(...x...)[x\backslash !s]$ | $\rightsquigarrow$ | evaluate inside $s$ until the outermost •... appears |
| $(...x...)[x\backslash !\bullet s]$ | $\rightsquigarrow$ | perform the substitution |

| CBN | CBV | CBNd$^{\times}$ |
|---|---|---|
| $t[x\backslash !\bullet s]$ | $t[x\backslash s]$ | $t[x\backslash !s]$ |

# Outline

# Conclusion

## Summary

- $\mathsf{MELL}_\bullet$: a variant of MELL with restricted dereliction.
- $\lambda^\bullet$: a derived linear $\lambda$-calculus with controlled sharing.
- Embeddings of CBN, CBV, CBNd, Bang calculus.

# Conclusion

## Summary

- $\mathsf{MELL}_\bullet$: a variant of MELL with restricted dereliction.
- $\lambda^\bullet$: a derived linear $\lambda$-calculus with controlled sharing.
- Embeddings of CBN, CBV, CBNd, Bang calculus.

## Ongoing/future work

- Embeddings of families of strategies: weak, head, open, strong, etc.
- $\mathsf{MELL}_\bullet$: proof nets.
  In this talk, proof nets were used just for intuition.
  Cut elimination in sequent calculus $\mathsf{MELL}_\bullet$ does not actually share.
- $\mathsf{MELL}_\bullet$: models.
  *E.g.* adapting phase or coherence semantics is not obvious.
- Theory of $\lambda^\bullet$: confluence, standardization, solvability, etc.