Exact measures of evaluation in classical natural deduction

Pierre VIAL
Équipe Gallinette
Inria - LS2N

joint w. with Delia Kesner (IRIF)

March 19, 2019

Where we lie

Non-Idempotent
Gardner 94 - de Carvalho 07

Intersection
Coppo-Dezani 80

Type Theory

Where we lie

Non-Idempotent

Gardner 94 - de Carvalho 07

Intersection

Coppo-Dezani 80

Type Theory $\rightarrow$ Curry-Howard correspondence

Where we lie

Non-Idempotent

Gardner 94 - de Carvalho 07

Intersection

Coppo-Dezani 80

characterizes:

- normalization
- complexity classes
- MSO-sat.

Type Theory

Curry-Howard correspondence

Where we lie

Non-Idempotent

Gardner 94 - de Carvalho 07

Intersection

Coppo-Dezani 80

characterizes:

- normalization

Type Theory

Curry-Howard correspondence

Quantitative info (upper bounds)

Non-Idempotent

Gardner 94 - de Carvalho 07

characterizes:

- normalization

Intersection

Coppo-Dezani 80

Type Theory

Curry-Howard correspondence

Where we lie

Quantitative info
(upper bounds)

Non-Idempotent
Gardner 94 - de Carvalho 07

characterizes:
- normalization

Intersection
Coppo-Dezani 80

Type Theory

Quantitative info
(upper bounds)

Non-Idempotent

Intersection

Type Theory

Where we lie

Non-Idempotent

Intersection

Type Theory

Quantitative info
(upper bounds)

Exact measures

Bernadet-Lengrand 11

Accatoli-Kesner-Lengrand 18

Quantitative info
(upper bounds)

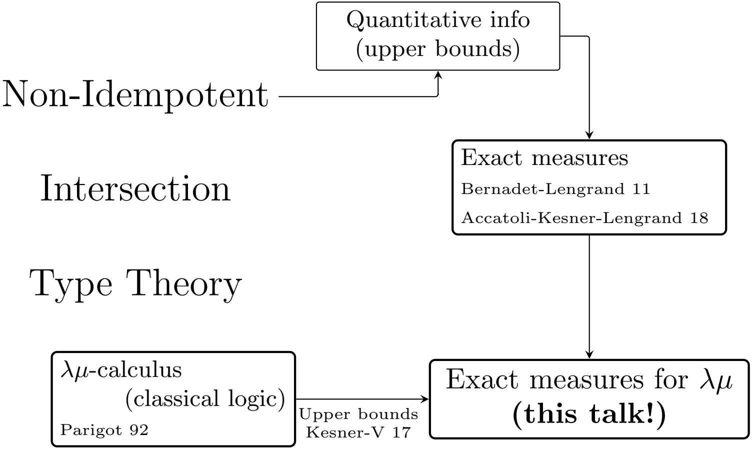
Non-Idempotent

Intersection

Type Theory

Exact measures
Bernadet-Lengrand 11
Accatoli-Kesner-Lengrand 18

$\lambda\mu$-calculus
(classical logic)
Parigot 92


Non-Idempotent
Intersection
Type Theory
Quantitative info
(upper bounds)
Exact measures
Bernadet-Lengrand 11
Accatoli-Kesner-Lengrand 18
$\lambda\mu$-calculus
(classical logic)
Parigot 92
Upper bounds
Kesner-V 17
Exact measures for $\lambda\mu$
(this talk!)

Plan

Intersections types (Coppo, Dezani, 1980)

Goal

Equivalences of the form

"the program t is typable iff it can reach a terminal state"

Idea: **several** certificates to a same subprogram (next slides).

Intersections types (Coppo, Dezani, 1980)

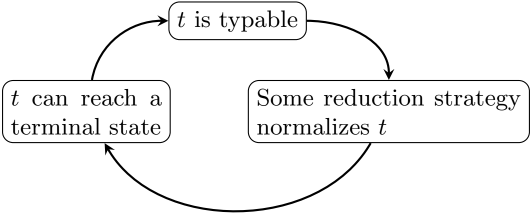
Goal

Equivalences of the form

"the program t is typable iff it can reach a terminal state"

Idea: **several** certificates to a same subprogram (next slides).

Proof: by the "circular" implications:



Intersections types (Coppo, Dezani, 1980)

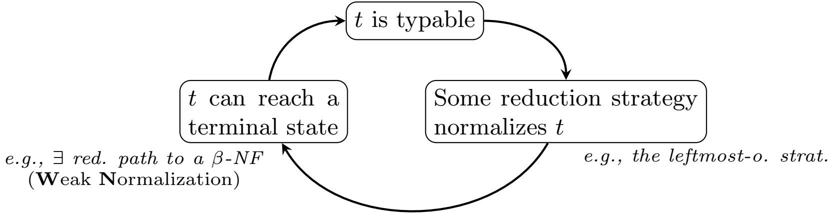
Goal

Equivalences of the form

"the program t is typable iff it can reach a terminal state"

Idea: **several** certificates to a same subprogram (next slides).

Proof: by the "circular" implications:



Intersections types (Coppo, Dezani, 1980)

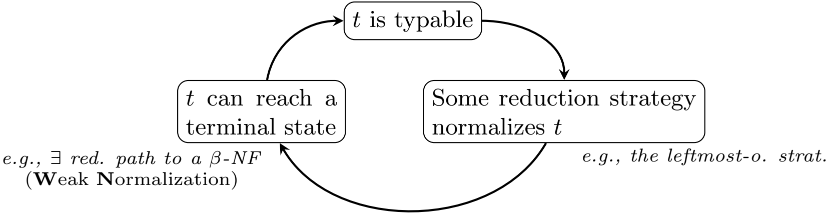
Goal

Equivalences of the form

> *"the program t is typable iff it can reach a terminal state"*

Idea: **several** certificates to a same subprogram (next slides).

Proof: by the "circular" implications:



t is WN iff the leftmost-o. stategy terminates on t

nothing to do with types

Intersections types (Coppo, Dezani, 1980)

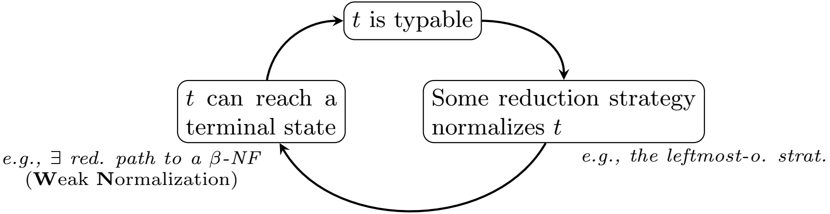
Goal

Equivalences of the form

"the program t is typable iff it can reach a terminal state"

Idea: **several** certificates to a same subprogram (next slides).

Proof: by the "circular" implications:



Intersections types (Coppo, Dezani, 1980)

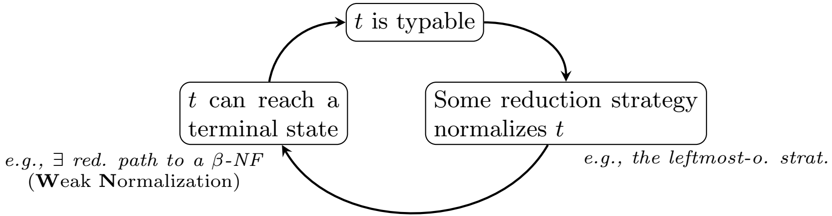
Goal

Equivalences of the form

> *"the program t is typable iff it can reach a terminal state"*

Idea: **several** certificates to a same subprogram (next slides).

Proof: by the "circular" implications:



Intersection types

- Perhaps too expressive...
- ...but certify **reduction strategies!**

- Naively, $A \wedge B$ stands for $A \cap B$:

$$\textit{t is of type } A \wedge B \textit{ if } t \textit{ can be typed with } A \textit{ as well as } B.$$

$$\frac{I : A \to A \qquad I : (A \to B) \to (A \to B)}{I : (A \to A) \wedge ((A \to B) \to (A \to B))} \wedge\!-\mathtt{intro} \qquad (\textit{with } I = \lambda x.x)$$

Intuitions (syntax)

- Naively, $A \wedge B$ stands for $A \cap B$:

$$\textit{t is of type } A \wedge B \textit{ if } t \textit{ can be typed with } A \textit{ as well as } B.$$

$$\frac{I : A \to A \qquad I : (A \to B) \to (A \to B)}{I : (A \to A) \wedge ((A \to B) \to (A \to B))} \wedge-\texttt{intro} \qquad \textit{(with } I = \lambda x.x\textit{)}$$

- Intersection = kind of *finite polymorphism*.

$$(A \to A) \wedge ((A \to B) \to (A \to B)) = \textbf{double} \text{ instance of } \forall X.X \to X$$

$$\textit{(with } X = A \textit{ and } X = A \to B\textit{)}$$

- Naively, $A \wedge B$ stands for $A \cap B$:

$$\textit{t is of type } A \wedge B \textit{ if } t \textit{ can be typed with } A \textit{ as well as } B.$$

$$\frac{I : A \to A \qquad I : (A \to B) \to (A \to B)}{I : (A \to A) \wedge ((A \to B) \to (A \to B))} \wedge -\texttt{intro} \qquad \textit{(with } I = \lambda x.x\textit{)}$$

- Intersection = kind of *finite polymorphism*.

$$(A \to A) \wedge ((A \to B) \to (A \to B)) = \textbf{double} \text{ instance of } \forall X.X \to X$$

$$\textit{(with } X = A \textit{ and } X = A \to B\textit{)}$$

- But *less constrained*:

$$\text{assigning } x : o \wedge (o \to o') \wedge (o \to o) \to o \text{ is legal.}$$

$$\textit{(}\boldsymbol{not}\textit{ an instance of a polymorphic type except } \forall X.X := \texttt{False}\textit{!)}$$

Subject Reduction and Subject Expansion

A good intersection type system should enjoy:

Subject Reduction (SR):
Typing is stable under reduction.

Subject Expansion (SE):
Typing is stable under anti-reduction.

SE is usually not verified by simple or polymorphic type systems

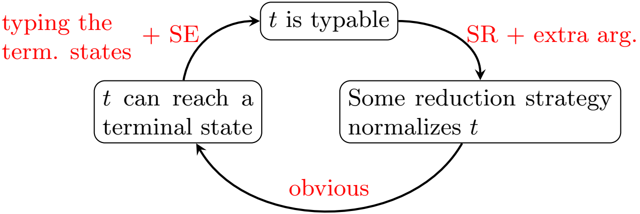
Subject Reduction and Subject Expansion

A good intersection type system should enjoy:

Subject Reduction (SR):
Typing is stable under reduction.

Subject Expansion (SE):
Typing is stable under anti-reduction.

SE is usually not verified by simple or polymorphic type systems

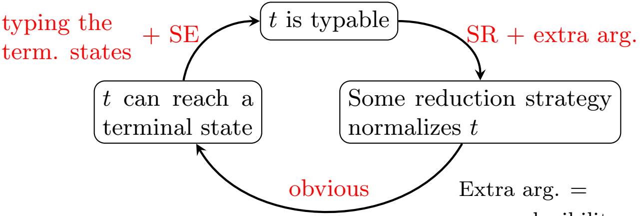


A good intersection type system should enjoy:

Subject Reduction (SR):
Typing is stable under reduction.

Subject Expansion (SE):
Typing is stable under anti-reduction.

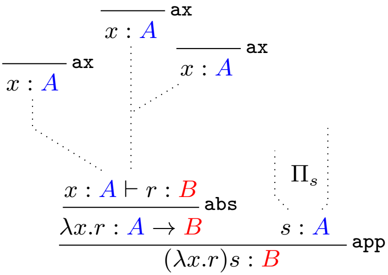
SE is usually not verified by simple or polymorphic type systems



Extra arg. =

- reducibility cand.
- non-trivial well-founded order.
- can it be simpler?

Subject Reduction (SR):
Typing is stable under reduction.



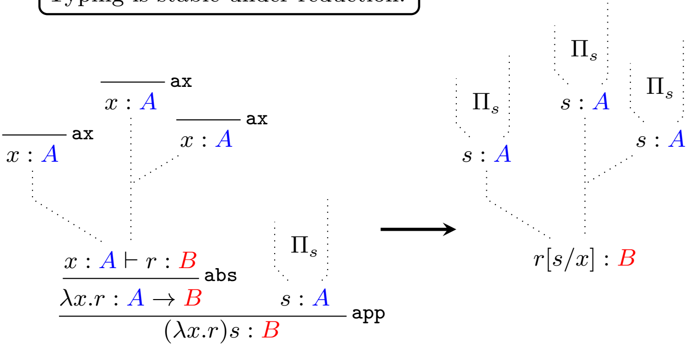
Ensuring Subject Expansion

Subject Reduction (SR):
Typing is stable under reduction.

$$\dfrac{}{x : A}\,\texttt{ax} \qquad \dfrac{}{x : A}\,\texttt{ax} \qquad \dfrac{}{x : A}\,\texttt{ax}$$

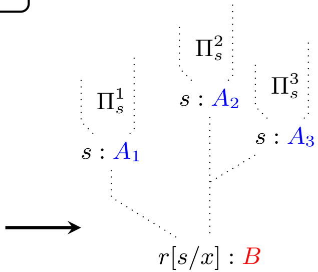
$$\dfrac{\dfrac{x : A \vdash r : B}{\lambda x.r : A \to B}\,\texttt{abs} \qquad \overset{\Pi_s}{s : A}}{(\lambda x.r)s : B}\,\texttt{app} \qquad \longrightarrow$$

Subject Reduction (SR):
Typing is stable under reduction.

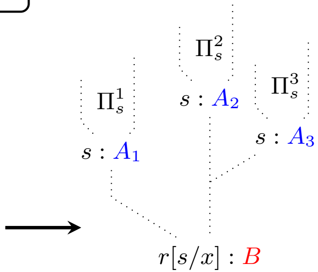


Ensuring Subject Expansion

Subject Expansion (SE):
Typing is stable under anti-reduction.



Subject Expansion (SE):
Typing is stable under anti-reduction.

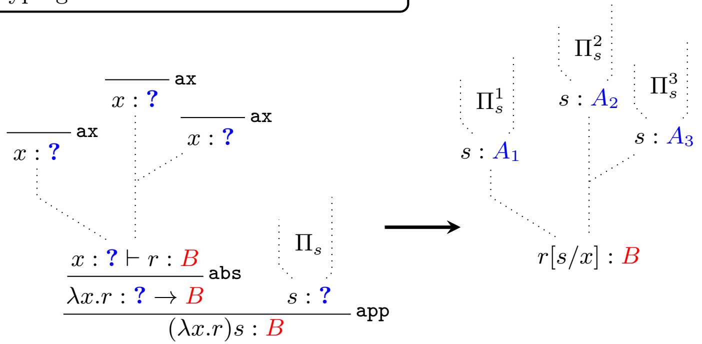


think of $(\lambda x.x\,x)I \to_\beta I\,I$

- Left occ. of I: $(A \to A) \to (A \to A)$
- Right occ. of I: $A \to A$

Ensuring Subject Expansion

Subject Expansion (SE):
Typing is stable under anti-reduction.

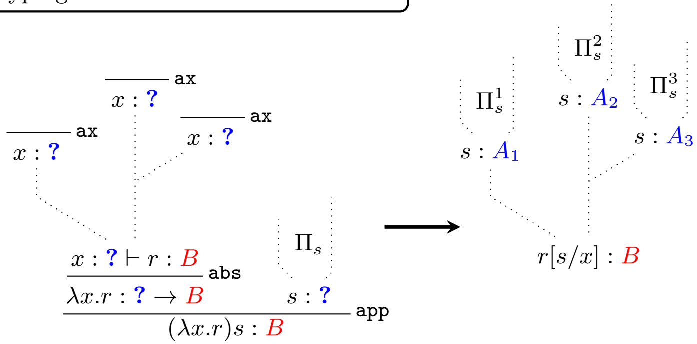


think of $(\lambda x.x\,x)I \to_\beta I\,I$

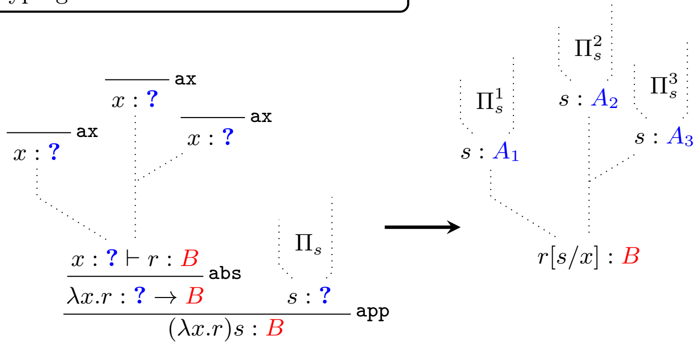
- Left occ. of I: $(A\to A)\to(A\to A)$
- Right occ. of I: $A\to A$

Ensuring Subject Expansion

Subject Expansion (SE):
Typing is stable under anti-reduction.



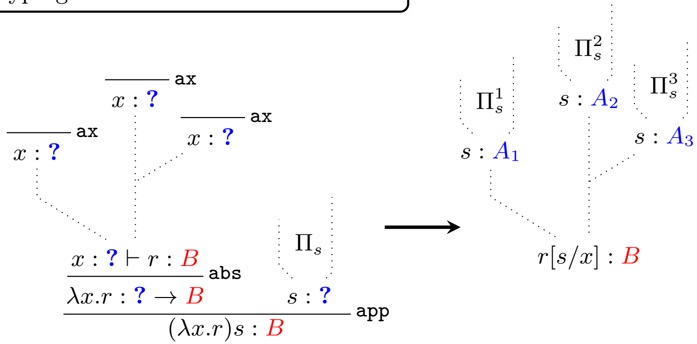
Subject Expansion (SE):
Typing is stable under anti-reduction.



Solution:
- Allow several type assignments for a same variable/subterm

$x : A_1 \wedge A_2 \wedge A_3$

Subject Expansion (SE):
Typing is stable under anti-reduction.



Solution:

- Allow several type assignments for a same variable/subterm

$$x : A_1 \wedge A_2 \wedge A_3 \vdash x : A_i \ (i = 1, 2, 3)$$

Typing every normal form

- Consider $(y(x\,(\lambda z.z)))\,(x\,(\lambda z.z\,c))$

Typing every normal form

- Consider $(y(x\,(\lambda z.z)))\,(x\,(\lambda z.z\,c))$
- We want $x : E \to F$

Typing every normal form

- Consider $(y(x\,(\lambda z.z)))\,(x\,(\lambda z.z\,c))$
- We want $x : E \to F$
- $\lambda z.z : A \to A$ *vs.* $\lambda z.z\,c : (C \to D) \to D$

Typing every normal form

- Consider $(y(x\,(\lambda z.z)))\,(x\,(\lambda z.z\,c))$
- We want $x : E \to F$
- $\lambda z.z : A \to A$ *vs.* $\lambda z.z\,c : (C \to D) \to D$

$$E = A \to A \text{ or } E = (C \to D) \to D?$$

Typing every normal form

- Consider $(y(x\,(\lambda z.z)))\,(x\,(\lambda z.z\,c))$
- We want $x : E \to F$
- $\lambda z.z : A \to A$ *vs.* $\lambda z.z\,c : (C \to D) \to D$

$$E = A \to A \text{ or } E = (C \to D) \to D?$$

Solution:

- Allow several type assignments for a same variable/subterm

Typing every normal form

- Consider $(y(x\,(\lambda z.z)))\,(x\,(\lambda z.z\,c))$
- We want $x : E \to F$
- $\lambda z.z : A \to A$ *vs.* $\lambda z.z\,c : (C \to D) \to D$

$$E = A \to A \text{ or } E = (C \to D) \to D?$$

Solution:
- Allow several type assignments for a same variable/subterm

- Typing normal form: just structural induction (no clash).

Non-idempotency

Computation causes **duplication**.

Computation causes **duplication**.

Non-idempotent intersection types

Disallow duplication for typing certificates.

- $\rightsquigarrow$ Possibly many certificates (subderivations) for a subprogram.
- $\rightsquigarrow$ Size of certificates decreases.

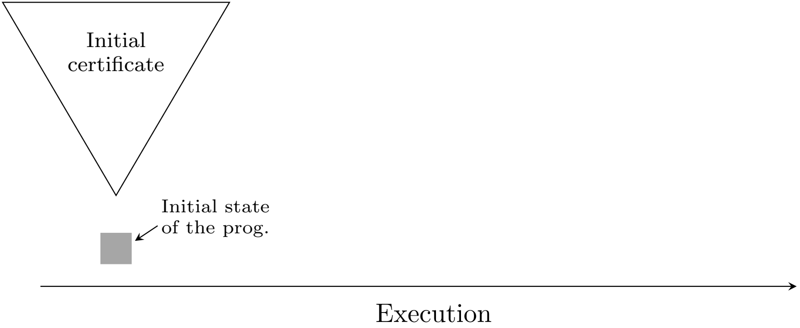
Non-idempotency

Computation causes **duplication**.

Non-idempotent intersection types

Disallow duplication for typing certificates.

- $\rightsquigarrow$ Possibly many certificates (subderivations) for a subprogram.
- $\rightsquigarrow$ Size of certificates decreases.

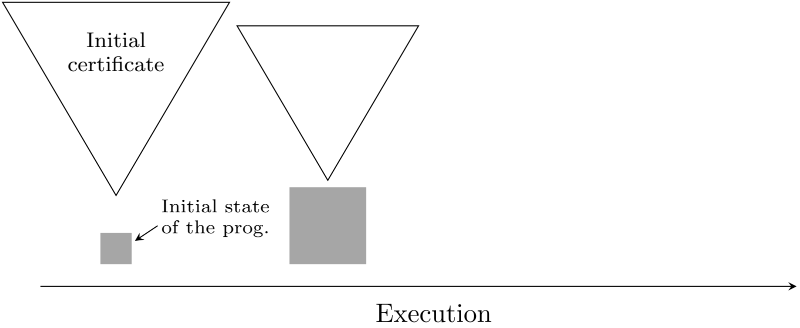


Computation causes **duplication**.

Non-idempotent intersection types

Disallow duplication for typing certificates.

- ⇝ Possibly many certificates (subderivations) for a subprogram.
- ⇝ Size of certificates decreases.

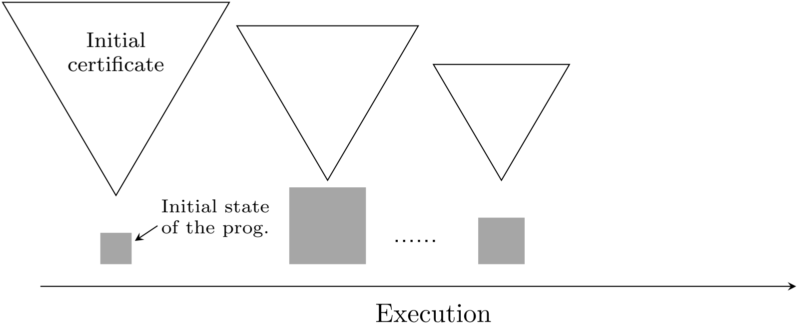


Computation causes **duplication**.

Non-idempotent intersection types

Disallow duplication for typing certificates.

- $\rightsquigarrow$ Possibly many certificates (subderivations) for a subprogram.
- $\rightsquigarrow$ Size of certificates decreases.

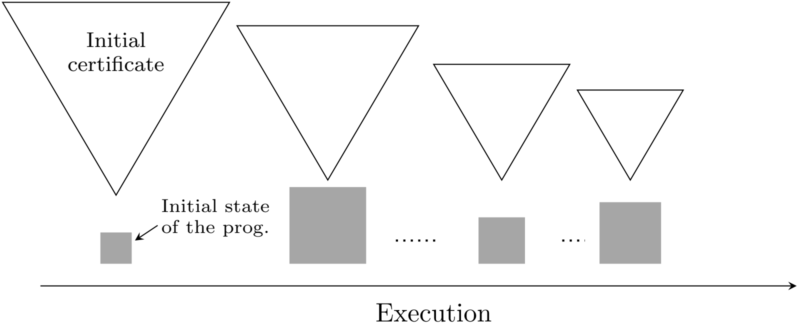


Computation causes **duplication**.

Non-idempotent intersection types

Disallow duplication for typing certificates.

- $\rightsquigarrow$ Possibly many certificates (subderivations) for a subprogram.
- $\rightsquigarrow$ Size of certificates decreases.

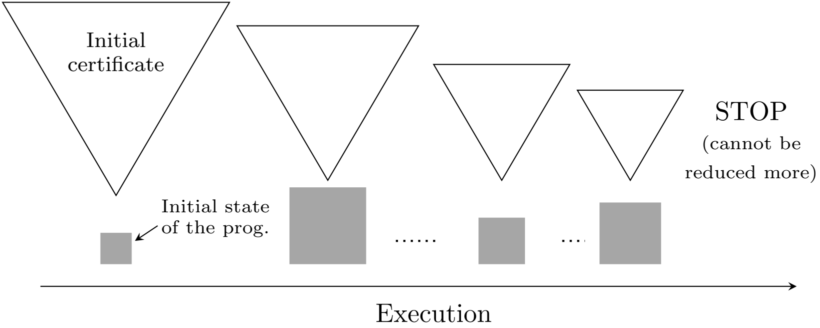


Computation causes **duplication**.

Non-idempotent intersection types

Disallow duplication for typing certificates.

- $\rightsquigarrow$ Possibly many certificates (subderivations) for a subprogram.
- $\rightsquigarrow$ Size of certificates decreases.

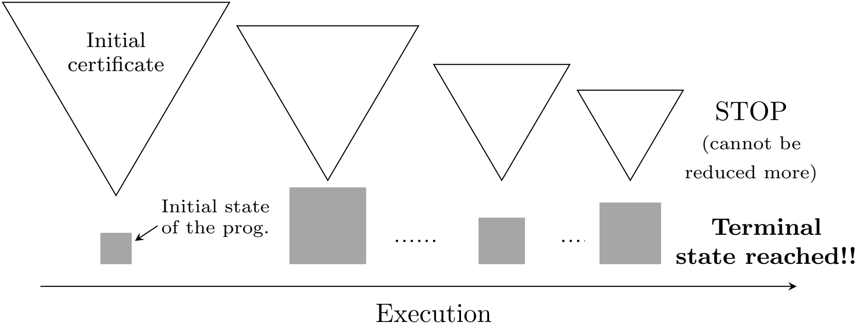


Computation causes **duplication**.

Non-idempotent intersection types

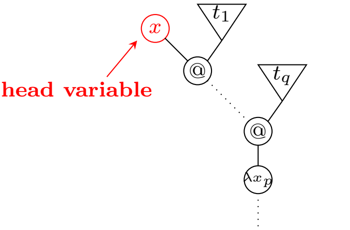
Disallow duplication for typing certificates.

- $\rightsquigarrow$ Possibly many certificates (subderivations) for a subprogram.
- $\rightsquigarrow$ Size of certificates decreases.

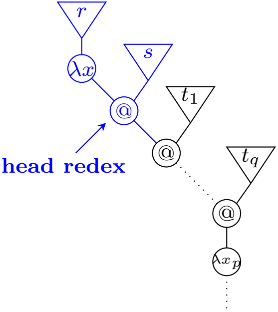


Plan

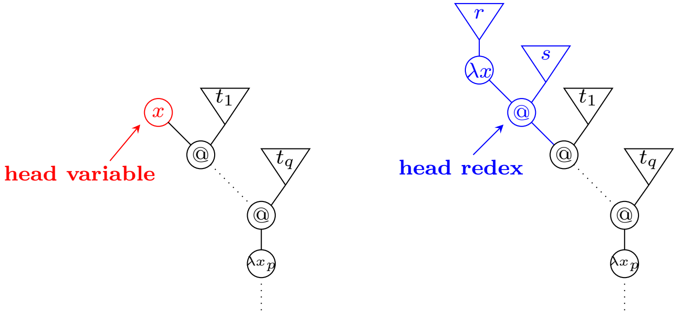
Head Normalization (λ)



Head Normal Form



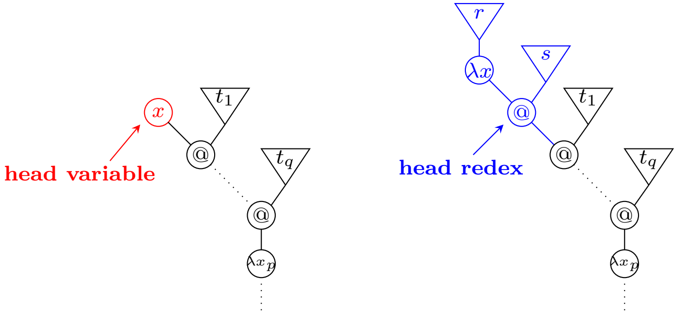
Head Reducible Term



Head Normal Form **Head Reducible Term**

- t is **head normalizing (HN)** if $\exists$ reduction path from t to a HNF.

Head Normalization (λ)



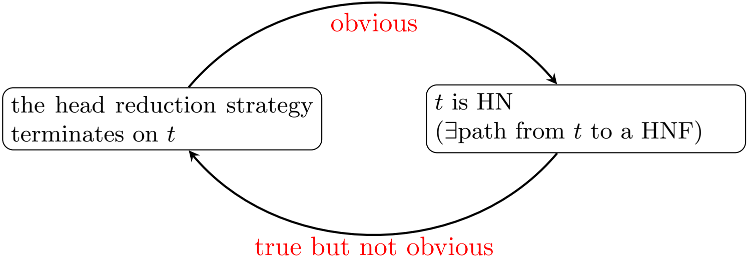
Head Normal Form **Head Reducible Term**

- t is **head normalizing (HN)** if $\exists$ reduction path from t to a HNF.
- The **head reduction strategy**: reducing head redexes while it is possible.

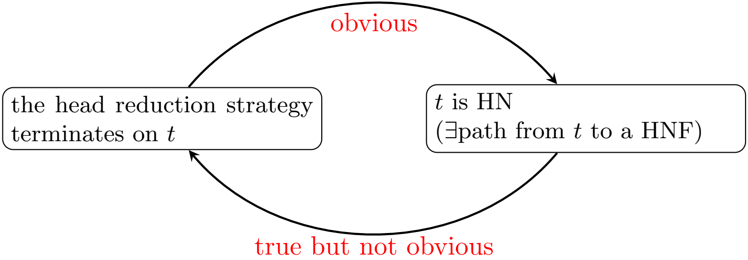
Head Normalization (λ)

- t is **head normalizing (HN)** if $\exists$ reduction path from t to a HNF.
- The **head reduction strategy**: reducing head redexes while it is possible.

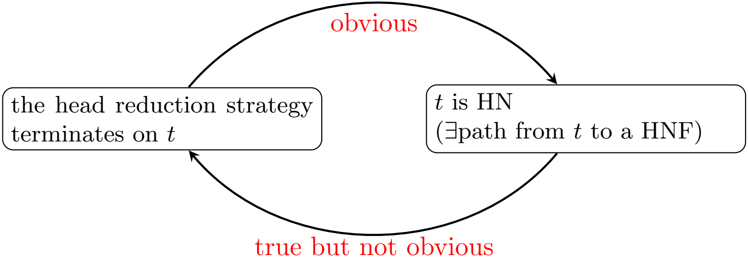
Head Normalization (λ)



- t is **head normalizing (HN)** if $\exists$ reduction path from t to a HNF.
- The **head reduction strategy**: reducing head redexes while it is possible.



- The **head reduction strategy**: reducing head redexes while it is possible.



Intersection types come to help!

- The **head reduction strategy**: reducing head redexes while it is possible.

Intersection types (Coppo-Dezani 80)

- Type constructors: $o \in \mathscr{O}$, $\rightarrow$ and $\wedge$ (intersection).

Intersection types (Coppo-Dezani 80)

- Type constructors: $o \in \mathscr{O}$, $\to$ and $\wedge$ (intersection).

- **Strict types**:
  no inter. on the *right* h.s. of $\to$, *e.g.*, $(A \wedge B) \to A$, not $A \to (B \wedge C)$

  $\rightsquigarrow$ *no intro/elim. rules for* $\wedge$

Intersection types (Coppo-Dezani 80)

- Type constructors: $o \in \mathscr{O}$, $\to$ and $\wedge$ (intersection).

- **Strict types**:
  no inter. on the *right* h.s. of $\to$, *e.g.*, $(A \wedge B) \to A$, not $A \to (B \wedge C)$

  $\rightsquigarrow$ *no intro/elim. rules for* $\wedge$

$\boxed{\textbf{Assoc.: } (A \wedge B) \wedge C \sim A \wedge (B \wedge C)}$ $\boxed{\textbf{Comm.: } A \wedge B \sim B \wedge A}$

Intersection types (Coppo-Dezani 80)

- Type constructors: $o \in \mathscr{O}$, $\to$ and $\wedge$ (intersection).

- **Strict types**:
  no inter. on the *right* h.s. of $\to$, *e.g.*, $(A \wedge B) \to A$, not $A \to (B \wedge C)$

  $\leadsto$ *no intro/elim. rules for* $\wedge$

$\boxed{\textbf{Assoc.: } (A \wedge B) \wedge C \sim A \wedge (B \wedge C)}$ $\boxed{\textbf{Comm.: } A \wedge B \sim B \wedge A}$

$\boxed{\textbf{Idempotency? } A \wedge A \sim A}$

Intersection types (Coppo-Dezani 80)

- Type constructors: $o \in \mathscr{O}$, $\to$ and $\wedge$ (intersection).

- **Strict types**:
  no inter. on the *right* h.s. of $\to$, *e.g.*, $(A \wedge B) \to A$, not $A \to (B \wedge C)$

  $\rightsquigarrow$ *no intro/elim. rules for* $\wedge$

$\boxed{\textbf{Assoc.:}\ (A \wedge B) \wedge C \sim A \wedge (B \wedge C)}$ $\boxed{\textbf{Comm.:}\ A \wedge B \sim B \wedge A}$

$\boxed{\textbf{Idempotency?}\ A \wedge A \sim A}$

Yes — Coppo-Dezani 80 — Typing= *qualitative* info.

No — Gardner 94 - de Carvalho 07 — Typing= *quantitative* info.

Intersection types (Coppo-Dezani 80)

- Type constructors: $o \in \mathscr{O}$, $\rightarrow$ and $\wedge$ (intersection).

- **Strict types**:
  no inter. on the *right* h.s. of $\rightarrow$, *e.g.*, $(A \wedge B) \rightarrow A$, not $A \rightarrow (B \wedge C)$

  $\rightsquigarrow$ *no intro/elim. rules for* $\wedge$

Assoc.: $(A \wedge B) \wedge C \sim A \wedge (B \wedge C)$

Comm.: $A \wedge B \sim B \wedge A$

Idempotency? $A \wedge A \sim A$

Yes

Coppo-Dezani 80

Typing= *qualitative* info.

No

Gardner 94 - de Carvalho 07

Typing= *quantitative* info.

- Collapsing $A \wedge B \wedge C$ into $[A, B, C]$ (**multiset**) $\rightsquigarrow$ no need for perm rules etc.

$$A \wedge B \wedge A := [A, B, A] = [A, A, B] \neq [A, B] \qquad [A, B, A] = [A, B] + [A]$$

Types: $\tau,\ \sigma \quad ::= \quad o \quad | \quad [\sigma_i]_{i \in I} \to \tau$

- **intersection** = **multiset** of types $[\sigma_i]_{i \in I}$
- only on the left-h.s of $\to$ (*strictness*)

System $\mathscr{R}_0$ (Gardner 94-de Carvalho 07)

Types: $\quad \tau,\ \sigma \quad ::= \quad o \quad | \quad [\sigma_i]_{i \in I} \to \tau$

- **intersection** = **multiset** of types $[\sigma_i]_{i \in I}$
- only on the left-h.s of $\to$ (*strictness*)

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma;\ x : [\sigma_i]_{i \in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i \in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i \in I} \to \tau \qquad (\Gamma_i \vdash u : \sigma_i)_{i \in I}}{\Gamma +_{i \in I} \Gamma_i \vdash t\, u : \tau}\ \texttt{app}$$

System $\mathscr{R}_0$ (Gardner 94-de Carvalho 07)

$$\text{Types:} \qquad \tau,\ \sigma \quad ::= \quad o \quad | \quad [\sigma_i]_{i\in I} \to \tau$$

- **intersection** = **multiset** of types $[\sigma_i]_{i\in I}$
- only on the left-h.s of $\to$ (*strictness*)

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma; x : [\sigma_i]_{i\in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i\in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i\in I} \to \tau \qquad (\Gamma_i \vdash u : \sigma_i)_{i\in I}}{\Gamma +_{i\in I} \Gamma_i \vdash t\,u : \tau}\ \texttt{app}$$

Remark

- **Relevant** system (no weakening, *cf.* `ax`-rule)

System $\mathscr{R}_0$ (Gardner 94-de Carvalho 07)

Types: $\tau, \sigma \quad ::= \quad o \quad | \quad [\sigma_i]_{i \in I} \to \tau$

- **intersection** = **multiset** of types $[\sigma_i]_{i \in I}$
- only on the left-h.s of $\to$ (*strictness*)

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma; x : [\sigma_i]_{i \in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i \in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i \in I} \to \tau \qquad (\Gamma_i \vdash u : \sigma_i)_{i \in I}}{\Gamma +_{i \in I} \Gamma_i \vdash t\, u : \tau}\ \texttt{app}$$

Remark

- **Relevant** system (no weakening, *cf.* `ax`-rule)
- **Non-idempotency ($\sigma \wedge \sigma \neq \sigma$):**
  in `app`-rule, pointwise multiset sum *e.g.*,

$$(x : [\sigma]; y : [\tau]) + (x : [\sigma, \tau]) = x : [\sigma, \sigma, \tau]; y : [\tau]$$

$$\text{Types:} \qquad \tau,\ \sigma \quad ::= \quad o \quad | \quad [\sigma_i]_{i\in I} \to \tau$$

- **intersection** = **multiset** of types $[\sigma_i]_{i\in I}$
- only on the left-h.s of $\to$ (*strictness*)

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma; x : [\sigma_i]_{i\in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i\in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i\in I} \to \tau \qquad (\Gamma_i \vdash u : \sigma_i)_{i\in I}}{\Gamma +_{i\in I} \Gamma_i \vdash t\, u : \tau}\ \texttt{app}$$

Example

$$\cfrac{\cfrac{}{f : [o] \to o}\ \texttt{ax} \qquad \cfrac{\cfrac{}{f : [o] \to o}\ \texttt{ax} \qquad \cfrac{}{x : o}\ \texttt{ax}}{f\, x : o}\ \texttt{app}}{f(f\, x) : o}\ \texttt{app}$$

Types: $\tau,\ \sigma \quad ::= \quad o \quad | \quad [\sigma_i]_{i\in I} \to \tau$

- **intersection** = **multiset** of types $[\sigma_i]_{i\in I}$
- only on the left-h.s of $\to$ (*strictness*)

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma;\ x : [\sigma_i]_{i\in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i\in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i\in I} \to \tau \qquad (\Gamma_i \vdash u : \sigma_i)_{i\in I}}{\Gamma +_{i\in I} \Gamma_i \vdash t\, u : \tau}\ \texttt{app}$$

Example

$$\dfrac{\dfrac{}{f : [o] \to o}\ \texttt{ax} \qquad \dfrac{\dfrac{}{f : [o] \to o}\ \texttt{ax} \qquad \dfrac{}{x : o}\ \texttt{ax}}{f\, x : o}\ \texttt{app}}{f : [[o] \to o, [o] \to o], x : [o] \vdash f(f\, x) : o}\ \texttt{app}$$

Types: $\tau,\ \sigma \quad ::= \quad o \quad | \quad [\sigma_i]_{i \in I} \to \tau$

- **intersection** = **multiset** of types $[\sigma_i]_{i \in I}$
- only on the left-h.s of $\to$ (*strictness*)

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma;\, x : [\sigma_i]_{i \in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i \in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i \in I} \to \tau \qquad (\Gamma_i \vdash u : \sigma_i)_{i \in I}}{\Gamma +_{i \in I} \Gamma_i \vdash t\,u : \tau}\ \texttt{app}$$

System $\mathscr{R}_0$ (Gardner 94-de Carvalho 07)

Types: $\quad \tau,\ \sigma \quad ::= \quad o \quad | \quad [\sigma_i]_{i\in I} \to \tau$

- **intersection** = **multiset** of types $[\sigma_i]_{i\in I}$
- only on the left-h.s of $\to$ (*strictness*)

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma; x : [\sigma_i]_{i\in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i\in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i\in I} \to \tau \qquad (\Gamma_i \vdash u : \sigma_i)_{i\in I}}{\Gamma +_{i\in I} \Gamma_i \vdash t\, u : \tau}\ \texttt{app}$$

Head redexes always typed!

System $\mathscr{R}_0$ (Gardner 94-de Carvalho 07)

Types: $\tau,\ \sigma \quad ::= \quad o \quad | \quad [\sigma_i]_{i \in I} \to \tau$

- **intersection** = **multiset** of types $[\sigma_i]_{i \in I}$
- only on the left-h.s of $\to$ (*strictness*)

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma;\, x : [\sigma_i]_{i \in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : [\sigma_i]_{i \in I} \to \tau}\ \texttt{abs}$$

$$\frac{\Gamma \vdash t : [\sigma_i]_{i \in I} \to \tau \qquad (\Gamma_i \vdash u : \sigma_i)_{i \in I}}{\Gamma +_{i \in I} \Gamma_i \vdash t\, u : \tau}\ \texttt{app}$$

Head redexes always typed!

but an arg. may be typed 0 time

Properties ($\mathscr{R}_0$)

- **Weighted Subject Reduction**
  - Reduction preserves types and environments, and...
  - ...*head* reduction strictly decreases the number of nodes of the deriv. tree (`size`).

    (actually, holds for any typed redex)
- **Subject Expansion**
  - Anti-reduction preserves types and environments.

Theorem (de Carvalho)

Let t be a λ-term. Then equivalence between:

1. *t is typable (in $\mathscr{R}_0$)*
2. *t is HN*
3. *the head reduction strategy terminates on t ($\rightsquigarrow$ **certification!**)*

Bonus (quantitative information)

If Π types t, then $\texttt{size}(\Pi)$ bounds the number of **steps** of the head red. strategy on t

Let t be a λ-term.

- **Head normalization (HN):**
  there is a path from t to a head normal form.

- **Weak normalization (WN):**
  there is *at least one path* from t to a β**-Normal Form** (NF)

- **Strong normalization (SN):**
  there is *no infinite path* starting at t.

$$\boxed{\mathrm{SN} \Rightarrow \mathrm{WN} \Rightarrow \mathrm{HN}}$$

Nota Bene: $y\,\Omega$ HNF but not WN $\qquad$ $(\lambda x.y)\Omega$ WN but not SN

Characterizing Weak and Strong Normalization

HN	System $\mathscr{R}_0$ $\boxed{\textit{any}\text{ arg. can be left }\textit{untyped}}$	$\mathtt{sz}(\Pi)$ bounds the number of *head* reduction steps
WN	System $\mathscr{R}_0$ + **unforgetfulness criterion** $\boxed{\textit{non-erasable}\text{ args must be typed}}$	$\mathtt{sz}(\Pi)$ bounds the number of leftmost-outermost red. steps (and more)
SN	Modify system $\mathscr{R}_0$ with **choice operator** $\boxed{\textit{all}\text{ args must be typed}}$	$\mathtt{sz}(\Pi)$ bounds the length of *any* reduction path

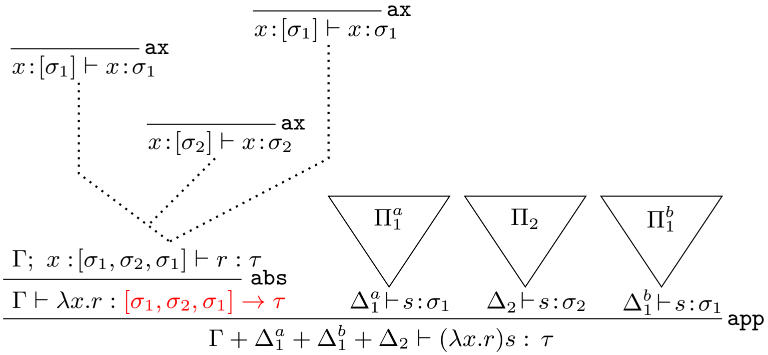
Subject reduction and expansion in $\mathscr{R}_0$

From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$

$$\dfrac{}{x:[\sigma_1] \vdash x:\sigma_1}\,\texttt{ax} \qquad \dfrac{}{x:[\sigma_2] \vdash x:\sigma_2}\,\texttt{ax} \qquad \dfrac{}{x:[\sigma_1] \vdash x:\sigma_1}\,\texttt{ax}$$

$$\dfrac{\dfrac{\Gamma;\ x:[\sigma_1,\sigma_2,\sigma_1] \vdash r:\tau}{\Gamma \vdash \lambda x.r : [\sigma_1,\sigma_2,\sigma_1] \to \tau}\,\texttt{abs} \quad \begin{array}{c}\Pi_1^a \\ \Delta_1^a \vdash s:\sigma_1\end{array} \quad \begin{array}{c}\Pi_2 \\ \Delta_2 \vdash s:\sigma_2\end{array} \quad \begin{array}{c}\Pi_1^b \\ \Delta_1^b \vdash s:\sigma_1\end{array}}{\Gamma + \Delta_1^a + \Delta_1^b + \Delta_2 \vdash (\lambda x.r)s : \tau}\,\texttt{app}$$

From a typing of $(\lambda x.r)s$ ... to a typing of $r[s/x]$

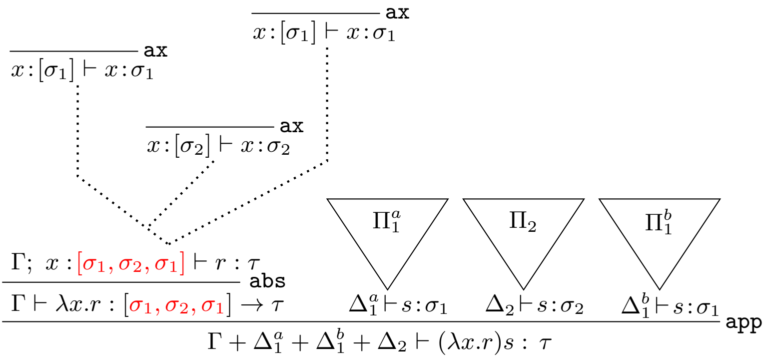


From a typing of $(\lambda x.r)s$ ... to a typing of $r[s/x]$

$$\dfrac{}{x:[\sigma_1] \vdash x:\sigma_1}\ \texttt{ax} \qquad \dfrac{}{x:[\sigma_2] \vdash x:\sigma_2}\ \texttt{ax} \qquad \dfrac{}{x:[\sigma_1] \vdash x:\sigma_1}\ \texttt{ax}$$

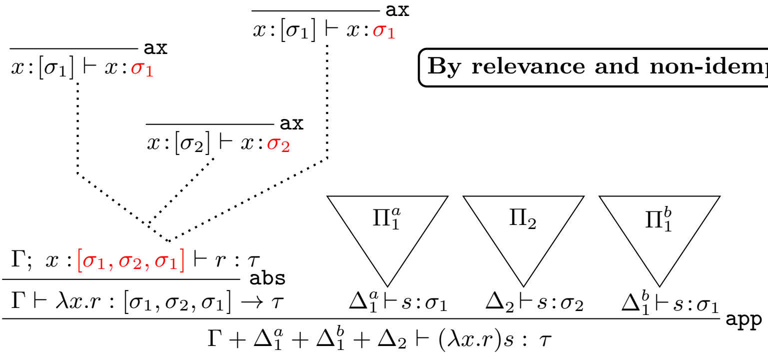
$$\dfrac{\dfrac{\Gamma;\ x:[\sigma_1,\sigma_2,\sigma_1] \vdash r:\tau}{\Gamma \vdash \lambda x.r : [\sigma_1,\sigma_2,\sigma_1] \to \tau}\ \texttt{abs} \qquad \begin{array}{c}\Pi_1^a \\ \Delta_1^a \vdash s:\sigma_1\end{array} \qquad \begin{array}{c}\Pi_2 \\ \Delta_2 \vdash s:\sigma_2\end{array} \qquad \begin{array}{c}\Pi_1^b \\ \Delta_1^b \vdash s:\sigma_1\end{array}}{\Gamma + \Delta_1^a + \Delta_1^b + \Delta_2 \vdash (\lambda x.r)s : \tau}\ \texttt{app}$$

From a typing of $(\lambda x.r)s$ ... to a typing of $r[s/x]$

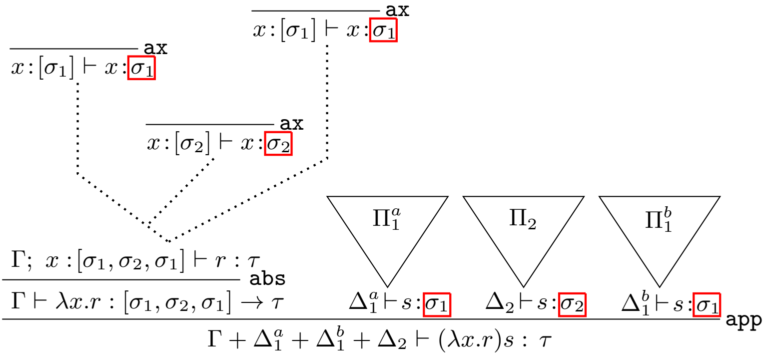


From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$

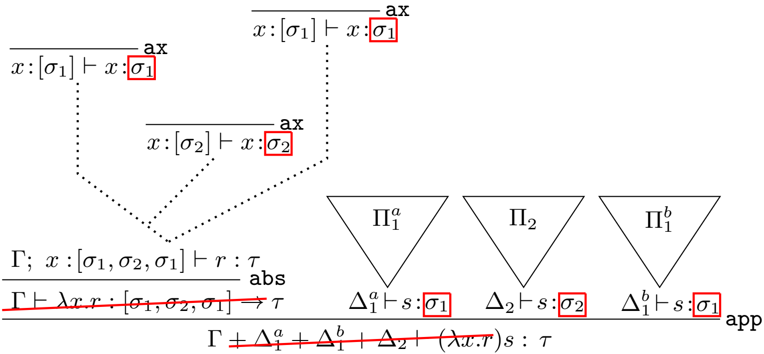
By relevance and non-idempotency!



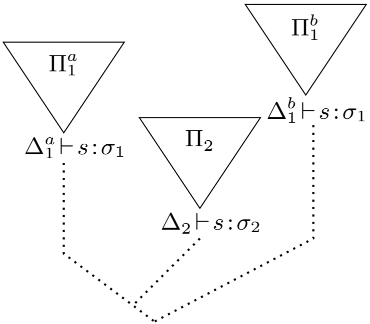
From a typing of $(\lambda x.r)s$ ... to a typing of $r[s/x]$



From a typing of $(\lambda x.r)s$ ... to a typing of $r[s/x]$

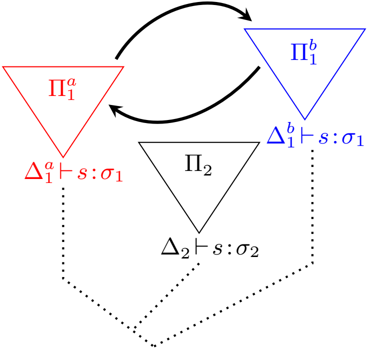


From a typing of $(\lambda x.r)s$ … to a typing of $r[s/x]$



$\Gamma + \Delta_1^a + \Delta_1^b + \Delta_2 \vdash r[s/x] : \tau$

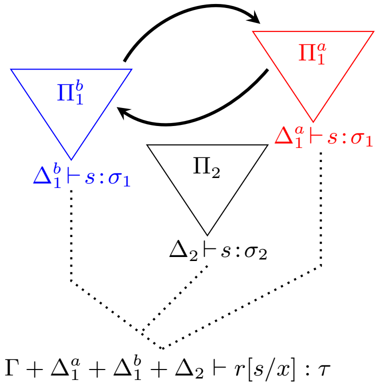
From a typing of $(\lambda x.r)s$ ... to a typing of $r[s/x]$



$\boxed{\textbf{Non-determinism of SR}}$

$$\Gamma + \Delta_1^a + \Delta_1^b + \Delta_2 \vdash r[s/x] : \tau$$

From a typing of $(\lambda x.r)s$ ... to a typing of $r[s/x]$



Non-determinism of SR

Plan

Exact measures for the λ-calculus (principles)

- Building on [Accatoli-Kesner-Lengrand,ICFP'18] and [Bernadet-Lengrand,LCMS'13]

Exact measures for the λ-calculus (principles)

- Building on [Accatoli-Kesner-Lengrand,ICFP'18] and [Bernadet-Lengrand,LCMS'13]
- Consider 3 reduction *strategies* S:

head $\rightarrow_{\mathtt{hd}}$	leftmost-o. $\rightarrow_{\mathtt{lo}}$	maximal $\rightarrow_{\mathtt{mx}}$ *(computes a longest red. path)*

Exact measures for the λ-calculus (principles)

- Building on [Accatoli-Kesner-Lengrand,ICFP'18] and [Bernadet-Lengrand,LCMS'13]
- Consider 3 reduction *strategies* S:

head $\rightarrow_{\mathtt{hd}}$	leftmost-o. $\rightarrow_{\mathtt{lo}}$	maximal $\rightarrow_{\mathtt{mx}}$ *(computes a longest red. path)*

- ***Goal***: finding a type system with annotated judg. $\Gamma \vdash^{(\ell,f)} t : \tau$ such that:

$$\boxed{t \rightarrow_S^{\ell} t' \text{ a } S\text{-norm. form with } |t'|_S = f \quad \text{iff } \Gamma \vdash_S^{(\ell,f)} t : \tau \text{ derivable}}$$

Exact measures for the λ-calculus (principles)

- Building on [Accatoli-Kesner-Lengrand,ICFP'18] and [Bernadet-Lengrand,LCMS'13]

- Consider 3 reduction *strategies* S:

head $\rightarrow_{\mathtt{hd}}$	leftmost-o. $\rightarrow_{\mathtt{lo}}$	maximal $\rightarrow_{\mathtt{mx}}$ *(computes a longest red. path)*

- ***Goal***: finding a type system with annotated judg. $\Gamma \vdash^{(\ell, f)} t : \tau$ such that:

$$\boxed{t \rightarrow_S^{\ell} t' \text{ a } S\text{-norm. form with } |t'|_S = f \quad \text{iff } \Gamma \vdash_S^{(\ell,f)} t : \tau \text{ derivable}}$$

Remark:

- `hd`-NF=HNF
  v.s. `lo`/`mx`-NF= full NF (no redex).

- $|\lambda x_1 \ldots x_p . x\, t_1 \ldots t_q|_{\mathtt{hd}} = p + q + 1$

New formalism

$\boxed{\begin{array}{l}\bullet\ \textbf{Persistent}\text{ elements} \\ \text{(remain in the NF)} \\ \bullet\ \textbf{Consuming}\text{ elements} \\ \text{(used during red.)}\end{array}}$

ex: $(\lambda x.y\,x\,x)_{@}z \rightarrow_\beta y\,z\,z$

New formalism

- **Persistent** elements (remain in the NF)
- **Consuming** elements (used during red.)

ex: $(\lambda x.y\,x\,x)_{@}z \to_\beta y\,z\,z$

- Explicit persistent arrow: $\nrightarrow$ (new type constructor)
- One type constant $\bullet$ (meaning "not applied")

ex: $\lambda x.x : [\bullet] \to \bullet$ ok
$\lambda x.x : [\bullet] \nrightarrow \bullet$ *not* ok

New formalism

- **Persistent** elements (remain in the NF)
- **Consuming** elements (used during red.)

ex: $(\lambda x.y\,x\,x)_{@}z \to_\beta y\,z\,z$

- Explicit persistent arrow: $\not\to$ (new type constructor)
- One type constant $\bullet$ (meaning "not applied")

ex: $\lambda x.x : [\bullet] \to \bullet$ ok
$\lambda x.x : [\bullet] \not\to \bullet$ *not* ok
$\lambda x.x : \bullet$ ok

New formalism

- **Persistent** elements (remain in the NF)
- **Consuming** elements (used during red.)

ex: $(\lambda x.y\,x\,x)@z \to_\beta y\,z\,z$

- Explicit persistent arrow: $\not\to$ (new type constructor)
- One type constant $\bullet$ (meaning "not applied")

ex: $\lambda x.x : [\bullet] \to \bullet$ ok
$\lambda x.x : [\bullet] \not\to \bullet$ *not* ok
$\lambda x.x : \bullet$ ok

-

(**Elementary types**)	σ, τ	$::=$	$\bullet \mid [\sigma_i]_{i\in I} \to \tau \mid [\sigma_i]_{i\in I} \not\to \tau$
(**Tight elem. types-hd**)	$\mathtt{tight}_{\mathtt{hd}}$	$::=$	$\bullet \mid [\,] \not\to \mathtt{tight}_{\mathtt{hd}}$
(**Tight elem. types-full**)	$\mathtt{tight}_{\mathtt{full}}$	$::=$	$\bullet \mid [\bullet] \not\to \mathtt{tight}_{\mathtt{full}}$

Remark: $\mathtt{Itight}_S([\sigma_i]_{i\in I})$ iff the σ_i are tight (tight intersection).

- **Persistent** elements (remain in the NF)
- **Consuming** elements (used during red.)

ex: $(\lambda x.y\,x\,x)_{@}z \to_\beta y\,z\,z$

- Explicit persistent arrow: $\not\to$ (new type constructor)
- One type constant $\bullet$ (meaning "not applied")

ex: $\lambda x.x : [\bullet] \to \bullet$ ok
$\lambda x.x : [\bullet] \not\to \bullet$ *not* ok
$\lambda x.x : \bullet$ ok

-

(**Elementary types**)	σ, τ	$::=$	$\bullet \mid [\sigma_i]_{i \in I} \to \tau \mid [\sigma_i]_{i \in I} \not\to \tau$
(**Tight elem. types-hd**)	$\mathtt{tight}_{\mathtt{hd}}$	$::=$	$\bullet \mid [\,] \not\to \mathtt{tight}_{\mathtt{hd}}$
(**Tight elem. types-full**)	$\mathtt{tight}_{\mathtt{full}}$	$::=$	$\bullet \mid [\bullet] \not\to \mathtt{tight}_{\mathtt{full}}$

Remark: $\mathtt{Itight}_S([\sigma_i]_{i \in I})$ iff the σ_i are tight (tight intersection).

This should be ok:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : [\,\bullet\,] \not\to [\,\bullet\,] \not\to \bullet}\,\mathtt{ax} \qquad u_1 : \bullet}{x\,u_1 : [\,\bullet\,] \not\to \bullet}\,\mathtt{app} \qquad u_2 : \bullet}{x\,u_1\,u_2 : \bullet}\,\mathtt{app}}{\lambda x.x\,u_1\,u_2 : \bullet}}{\lambda yx.x\,u_1\,u_2 : \bullet}$$

New formalism

- **Persistent** elements (remain in the NF)
- **Consuming** elements (used during red.)

ex: $(\lambda x.y\,x\,x)_{@}z \to_\beta y\,z\,z$

- Explicit persistent arrow: $\twoheadrightarrow$ (new type constructor)
- One type constant $\bullet$ (meaning "not applied")

ex: $\lambda x.x : [\bullet] \to \bullet$ ok
$\lambda x.x : [\bullet] \twoheadrightarrow \bullet$ *not* ok
$\lambda x.x : \bullet$ ok

-

(**Elementary types**)	σ, τ	$::=$	$\bullet \mid [\sigma_i]_{i\in I} \to \tau \mid [\sigma_i]_{i\in I} \twoheadrightarrow \tau$
(**Tight elem. types-hd**)	$\mathtt{tight}_{\mathtt{hd}}$	$::=$	$\bullet \mid [\,] \twoheadrightarrow \mathtt{tight}_{\mathtt{hd}}$
(**Tight elem. types-full**)	$\mathtt{tight}_{\mathtt{full}}$	$::=$	$\bullet \mid [\bullet] \twoheadrightarrow \mathtt{tight}_{\mathtt{full}}$

Remark: $\mathtt{Itight}_S([\sigma_i]_{i\in I})$ iff the σ_i are tight (tight intersection).

This should be ok:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : [\bullet] \twoheadrightarrow [\bullet] \twoheadrightarrow \bullet}\ \mathtt{ax} \qquad u_1 : \bullet}{x\,u_1 : [\bullet] \twoheadrightarrow \bullet}\ \mathtt{app} \qquad u_2 : \bullet}{x\,u_1\,u_2 : \bullet}\ \mathtt{app}}{\lambda x.x\,u_1\,u_2 : \bullet}}{\lambda yx.x\,u_1\,u_2 : \bullet}$$

- **Persistent** elements (remain in the NF)
- **Consuming** elements (used during red.)

ex: $(\lambda x.y\,x\,x)_{@}z \to_\beta y\,z\,z$

- Explicit persistent arrow: $\not\to$ (new type constructor)
- One type constant $\bullet$ (meaning "not applied")

ex: $\lambda x.x : [\bullet] \to \bullet$ ok
$\lambda x.x : [\bullet] \not\to \bullet$ *not* ok
$\lambda x.x : \bullet$ ok

-

(**Elementary types**)	σ, τ	$::=$	$\bullet \mid [\sigma_i]_{i\in I} \to \tau \mid [\sigma_i]_{i\in I} \not\to \tau$
(**Tight elem. types-hd**)	$\mathtt{tight}_{\mathtt{hd}}$	$::=$	$\bullet \mid [\,] \not\to \mathtt{tight}_{\mathtt{hd}}$
(**Tight elem. types-full**)	$\mathtt{tight}_{\mathtt{full}}$	$::=$	$\bullet \mid [\bullet] \not\to \mathtt{tight}_{\mathtt{full}}$

Remark: $\mathtt{Itight}_S([\sigma_i]_{i\in I})$ iff the σ_i are tight (tight intersection).

This should be ok:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : [\bullet] \not\to [\bullet] \not\to \bullet}\,\mathtt{ax} \qquad u_1 : \bullet}{x\,u_1 : [\bullet] \not\to \bullet}\,\mathtt{app} \qquad u_2 : \bullet}{x\,u_1\,u_2 : \bullet}\,\mathtt{app}}{\lambda x.x\,u_1\,u_2 : \bullet}}{\lambda yx.x\,u_1\,u_2 : \bullet}$$

- **Persistent** elements (remain in the NF)
- **Consuming** elements (used during red.)

ex: $(\lambda x.y\,x\,x)_{@}z \to_\beta y\,z\,z$

- Explicit persistent arrow: $\not\to$ (new type constructor)
- One type constant $\bullet$ (meaning "not applied")

ex: $\lambda x.x : [\bullet] \to \bullet$ ok
$\lambda x.x : [\bullet] \not\to \bullet$ *not* ok
$\lambda x.x : \bullet$ ok

-

(Elementary types)	σ, τ	$::=$	$\bullet \mid [\sigma_i]_{i\in I} \to \tau \mid [\sigma_i]_{i\in I} \not\to \tau$
(Tight elem. types-hd)	$\mathtt{tight}_{\mathtt{hd}}$	$::=$	$\bullet \mid [\,] \not\to \mathtt{tight}_{\mathtt{hd}}$
(Tight elem. types-full)	$\mathtt{tight}_{\mathtt{full}}$	$::=$	$\bullet \mid [\bullet] \not\to \mathtt{tight}_{\mathtt{full}}$

Remark: $\mathtt{Itight}_S([\sigma_i]_{i\in I})$ iff the σ_i are tight (tight intersection).

This should be ok:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : [\quad] \not\to [\quad] \not\to \bullet}\,\mathtt{ax}}{x\,u_1 : [\quad] \not\to \bullet}\,\mathtt{app}}{x\,u_1\,u_2 : \bullet}\,\mathtt{app}}{\lambda x.x\,u_1\,u_2 : \bullet}}{\lambda yx.x\,u_1\,u_2 : \bullet}$$

Rules

$$\Gamma \vdash^{({\color{red}\ell},{\color{blue}f})} t : \tau$$

number of β-steps — size of the norm. form

$$\frac{}{x : [\sigma] \vdash^{(0,1)} x : \sigma}\ (\mathtt{ax})$$

$$\frac{\Gamma \vdash^{(\ell,f)} t : \sigma}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell+1, f - \#\Gamma(x))} \lambda x.t : \Gamma(x) \to \sigma}\ (\to_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,f)} t : \mathtt{tight} \quad \mathtt{tight}(\Gamma(x))}{\Gamma \vdash^{(\ell,f+1)} \lambda x.t : \bullet}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell_t,f_t)} t : \mathcal{F} \qquad \Delta \Vdash^{(\ell_u,f_u)} u : \mathtt{dom}(\mathcal{F})}{\Gamma \wedge \Delta \vdash^{(\ell_f+\ell_u, f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\,u : \mathtt{codom}(\mathcal{F})}\ (\mathtt{app}_S)$$

with $\#_{\mathtt{p}} ? \to ? = 0$ and $\#_{\mathtt{p}} ? \nrightarrow ? = 1$.

Systems $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

$$\Gamma \vdash^{({\color{red}\ell},{\color{blue}f})} t : \tau$$

number of β-steps   size of the norm. form

$$\frac{}{x : [\sigma] \vdash^{(0,1)} x : \sigma}\ (\mathtt{ax})$$

$$\frac{\Gamma \vdash^{(\ell,f)} t : \sigma}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell+1,f-\#\Gamma(x))} \lambda x.t : \Gamma(x) \to \sigma}\ (\to_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,f)} t : \mathtt{tight} \quad \mathtt{tight}(\Gamma(x))}{\Gamma \vdash^{(\ell,f+1)} \lambda x.t : \bullet}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell_t,f_t)} t : \mathcal{F} \qquad \Delta \Vdash^{(\ell_u,f_u)} u : \mathtt{dom}(\mathcal{F})}{\Gamma \wedge \Delta \vdash^{(\ell_f+\ell_u,f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\,u : \mathtt{codom}(\mathcal{F})}\ (\mathtt{app}_S)$$

with $\#_{\mathtt{p}}? \to ? = 0$ and $\#_{\mathtt{p}}? \nrightarrow ? = 1$.

Systems $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

$$\Gamma \vdash^{(\ell, f)} t : \tau$$

number of β-steps — size of the norm. form

$$\frac{}{x : [\sigma] \vdash^{(0,1)} x : \sigma}\ (\mathtt{ax})$$

$$\frac{\Gamma \vdash^{(\ell,f)} t : \sigma}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell+1, f-\#\Gamma(x))} \lambda x.t : \Gamma(x) \to \sigma}\ (\to_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,f)} t : \mathtt{tight} \quad \mathtt{tight}(\Gamma(x))}{\Gamma \vdash^{(\ell,f+1)} \lambda x.t : \bullet}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell_t,f_t)} t : \mathcal{F} \qquad \Delta \Vdash^{(\ell_u,f_u)} u : \mathtt{dom}(\mathcal{F})}{\Gamma \wedge \Delta \vdash^{(\ell_f+\ell_u, f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\,u : \mathtt{codom}(\mathcal{F})}\ (\mathtt{app}_S)$$

with $\#_{\mathtt{p}}? \to ? = 0$ and $\#_{\mathtt{p}}? \not\to ? = 1$.

Systems $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

$$\Gamma \vdash^{(\ell,f)} t : \tau$$

number of β-steps — size of the norm. form

$$\frac{}{x : [\sigma] \vdash^{(0,1)} x : \sigma}\ (\mathtt{ax})$$

$$\frac{\Gamma \vdash^{(\ell,f)} t : \sigma}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell+1, f-\#\Gamma(x))} \lambda x.t : \Gamma(x) \to \sigma}\ (\to_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,f)} t : \mathtt{tight} \qquad \mathtt{tight}(\Gamma(x))}{\Gamma \vdash^{(\ell,f+1)} \lambda x.t : \bullet}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell_t,f_t)} t : \mathcal{F} \qquad \Delta \Vdash^{(\ell_u,f_u)} u : \mathtt{dom}(\mathcal{F})}{\Gamma \wedge \Delta \vdash^{(\ell_f+\ell_u, f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\, u : \mathtt{codom}(\mathcal{F})}\ (\mathtt{app}_S)$$

with $\#_{\mathtt{p}} ? \to ? = 0$ and $\#_{\mathtt{p}} ? \not\to ? = 1$.

Systems $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

$$\Gamma \vdash^{(\underbrace{\ell}_{\text{number of } \beta\text{-steps}}, \underbrace{f}_{\text{size of the norm. form}})} t : \tau$$

$$\frac{}{x : [\sigma] \vdash^{(0,1)} x : \sigma}\ (\mathtt{ax})$$

$$\frac{\Gamma \vdash^{(\ell,f)} t : \sigma}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell+1, f - \#\Gamma(x))} \lambda x.t : \Gamma(x) \to \sigma}\ (\to_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,f)} t : \mathtt{tight} \quad \mathtt{tight}(\Gamma(x))}{\Gamma \vdash^{(\ell,f+1)} \lambda x.t : \bullet}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell_t,f_t)} t : \mathcal{F} \qquad \Delta \Vdash^{(\ell_u,f_u)} u : \mathtt{dom}(\mathcal{F})}{\Gamma \wedge \Delta \vdash^{(\ell_f+\ell_u, f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\,u : \mathtt{codom}(\mathcal{F})}\ (\mathtt{app}_S)$$

with $\#_{\mathtt{p}} ? \to ? = 0$ and $\#_{\mathtt{p}} ? \nrightarrow ? = 1$.

Systems $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

$$\Gamma \vdash^{({\color{red}\ell},{\color{blue}f})} t : \tau$$

number of β-steps   size of the norm. form

$$\frac{}{x : [\sigma] \vdash^{(0,1)} x : \sigma}\ (\mathtt{ax})$$

$$\frac{\Gamma \vdash^{(\ell,f)} t : \sigma}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell+1, f-\#\Gamma(x))} \lambda x.t : \Gamma(x) \to \sigma}\ (\to_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,f)} t : \mathtt{tight} \quad \mathtt{tight}(\Gamma(x))}{\Gamma \vdash^{(\ell,f+1)} \lambda x.t : \bullet}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell_t,f_t)} t : \mathcal{F} \qquad \Delta \Vdash^{(\ell_u,f_u)} u : \mathtt{dom}(\mathcal{F})}{\Gamma \wedge \Delta \vdash^{(\ell_f+\ell_u, f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\, u : \mathtt{codom}(\mathcal{F})}\ (\mathtt{app}_S)$$

with $\#_{\mathtt{p}}? \to ? = 0$ and $\#_{\mathtt{p}}? \nrightarrow ? = 1$.

Systems $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

$$\Gamma \vdash^{(\ell, f)} t : \tau$$

number of β-steps — size of the norm. form

$$\frac{}{x : [\sigma] \vdash^{(0,1)} x : \sigma}\ (\mathtt{ax})$$

$$\frac{\Gamma \vdash^{(\ell,f)} t : \sigma}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell+1, f - \#\Gamma(x))} \lambda x.t : \Gamma(x) \to \sigma}\ (\to_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,f)} t : \mathtt{tight} \quad \mathtt{tight}(\Gamma(x))}{\Gamma \vdash^{(\ell,f+1)} \lambda x.t : \bullet}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell_t, f_t)} t : \mathcal{F} \qquad \Delta \Vdash^{(\ell_u, f_u)} u : \mathtt{dom}(\mathcal{F})}{\Gamma \wedge \Delta \vdash^{(\ell_f + \ell_u, f_t + f_u + \#_{\mathtt{p}} \mathcal{F})} t\, u : \mathtt{codom}(\mathcal{F})}\ (\mathtt{app}_S)$$

with $\#_{\mathtt{p}}? \to ? = 0$ and $\#_{\mathtt{p}}? \nrightarrow ? = 1$.

Systems $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

Rules

$$\Gamma \vdash^{(\ell, f)} t : \tau$$

number of β-steps (ℓ) — size of the norm. form (f)

$$\frac{}{x : [\sigma] \vdash^{(0,1)} x : \sigma} \,(\mathtt{ax})$$

$$\frac{\Gamma \vdash^{(\ell,f)} t : \sigma}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell+1, f - \#\Gamma(x))} \lambda x.t : \Gamma(x) \to \sigma} \,(\to_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,f)} t : \mathtt{tight} \quad \mathtt{tight}(\Gamma(x))}{\Gamma \vdash^{(\ell, f+1)} \lambda x.t : \bullet} \,(\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell_t, f_t)} t : \mathcal{F} \qquad \Delta \Vdash^{(\ell_u, f_u)} u : \mathtt{dom}(\mathcal{F})}{\Gamma \wedge \Delta \vdash^{(\ell_f + \ell_u, f_t + f_u + \#_{\mathtt{p}} \mathcal{F})} t\, u : \mathtt{codom}(\mathcal{F})} \,(\mathtt{app}_S)$$

with $\#_{\mathtt{p}} ? \to ? = 0$ and $\#_{\mathtt{p}} ? \nrightarrow ? = 1$.

Systems $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

Auxiliary: $$\frac{(\Gamma_k \vdash^{(\ell_k, f_k)} t : \tau_k)_{k \in K}}{\wedge_{k \in K} \Gamma_k \Vdash^{(+_{k\in K} \ell_k, +_{k \in K} f_k)} t : [\tau_k]_{k \in K}} \,\wedge$$

A simple example

Let $\mathtt{I} = \lambda x.x$.

$$(\lambda x.x\,x)\mathtt{I} \to_S \mathtt{I}\,\mathtt{I} \to_S \mathtt{I} \qquad (S \in \{\mathtt{hd}, \mathtt{lo}, \mathtt{mx}\})$$

Expected counter (2,2)

A simple example

Let $\mathtt{I} = \lambda x.x$.

$$(\lambda x.x\,x)\mathtt{I} \to_S \mathtt{I}\,\mathtt{I} \to_S \mathtt{I} \qquad (S \in \{\mathtt{hd}, \mathtt{lo}, \mathtt{mx}\})$$

Expected counter (2,2)

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : [[\bullet]\to\bullet] \vdash^{(0,1)} x : [\bullet]\to\bullet} \qquad \dfrac{}{x : [[\bullet]] \vdash^{(0,1)} x : \bullet}}{x : [[\bullet]\to\bullet, \bullet] \vdash^{(0,1+1=2)} x\,x : \bullet}}{\vdash^{(1,2-2=0)} \lambda x.x\,x : [[\bullet]\to\bullet, \bullet]\to\bullet}(\to_{\mathtt{i}}) \qquad \dfrac{\dfrac{}{x : [\bullet] \vdash^{(0,1)} x : \bullet}}{\vdash^{(1,0)} \mathtt{I} : [\bullet]\to\bullet}(\to_{\mathtt{i}}) \qquad \dfrac{\dfrac{}{x : [\bullet] \vdash^{(0,1)} x : \bullet}}{\vdash^{(0,1+1=2)} \mathtt{I} : \bullet}(\bullet_S)}{\vdash^{(1+1=2,2)} (\lambda x.x\,x)\mathtt{I}}$$

Properties of $\mathcal{X}^{\lambda}_{\texttt{hd/lo}}$

Definition:

- Tight judgment: $\Gamma \vdash^{(\ell,f)} t : \texttt{tight}$ with Γ tight.
- Tight deriv.: ccl with tight judg. (*local* criterion).

Properties of $\mathcal{X}^{\lambda}_{\texttt{hd/lo}}$

Definition:

- Tight judgment: $\Gamma \vdash^{(\ell,f)} t : \texttt{tight}$ with Γ tight.
- Tight deriv.: ccl with tight judg. (*local* criterion).

Theorem (HN/WN)

Let $t \in \Lambda$. Then:

$\Gamma \vdash^{(\ell,f)} t : \tau$ tight iff
- $t \to^{\ell}_{\texttt{hd/lo}} t' HNF$ or NF
- $|t'|_{\texttt{hd/lo}} = f$

Properties of $\mathcal{X}^{\lambda}_{\texttt{hd/lo}}$

Definition:

- Tight judgment: $\Gamma \vdash^{(\ell,f)} t : \texttt{tight}$ with Γ tight.
- Tight deriv.: ccl with tight judg. (*local* criterion).

Theorem (HN/WN)

Let $t \in \Lambda$. Then:

$\Gamma \vdash^{(\ell,f)} t : \tau$ tight iff
- $t \to^{\ell}_{\texttt{hd/lo}} t' HNF$ or NF
- $|t'|_{\texttt{hd/lo}} = f$

Theorem (SN)

Idem for SN and a **maximal** reduction strategy.

PROPERTIES OF $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

Definition:

- Tight judgment: $\Gamma \vdash^{(\ell,f)} t : \mathtt{tight}$ with Γ tight.
- Tight deriv.: ccl with tight judg. (*local* criterion).

Theorem (HN/WN)

Let $t \in \Lambda$. Then:

$$\Gamma \vdash^{(\ell,f)} t : \tau \text{ tight} \quad \text{iff} \quad \bullet\ t \to^{\ell}_{\mathtt{hd/lo}} t' HNF \text{ or NF}$$

$$\bullet\quad |t'|_{\mathtt{hd/lo}} = f$$

Theorem (SN)

Idem for SN and a **maximal** reduction strategy.

· *Just modify* $\mathtt{dom}(\mathcal{F})$ *with* $\mathtt{dom}_{\mathtt{mx}}([\,] \to \tau) = [\bullet]$

Properties of $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

Definition:

- Tight judgment: $\Gamma \vdash^{(\ell,f)} t : \mathtt{tight}$ with Γ tight.
- Tight deriv.: ccl with tight judg. (*local* criterion).

Theorem (HN/WN)

Let $t \in \Lambda$. Then:

$$\Gamma \vdash^{(\ell,f)} t : \tau \text{ tight} \quad \text{iff} \quad \begin{array}{l} \bullet\ t \to^{\ell}_{\mathtt{hd/lo}} t'\, HNF \text{ or NF} \\ \bullet\ \quad |t'|_{\mathtt{hd/lo}} = f \end{array}$$

Theorem (SN)

Idem for SN and a **maximal** reduction strategy.

· *Just modify* $\mathtt{dom}(\mathcal{F})$ *with* $\mathtt{dom}_{\mathtt{mx}}([\,] \to \tau) = [\bullet]$
· *Erasable args must now be typed*

Properties of $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

Definition:

- Tight judgment: $\Gamma \vdash^{(\ell,f)} t : \mathtt{tight}$ with Γ tight.
- Tight deriv.: ccl with tight judg. (*local* criterion).

Theorem (HN/WN)

Let $t \in \Lambda$. Then:

$$\Gamma \vdash^{(\ell,f)} t : \tau \text{ tight} \quad \text{iff} \quad \bullet\ t \to^{\ell}_{\mathtt{hd/lo}} t' HNF \text{ or NF} \qquad \bullet\ |t'|_{\mathtt{hd/lo}} = f$$

Theorem (SN)

Idem for SN and a **maximal** reduction strategy.

· *Just modify* $\mathtt{dom}(\mathcal{F})$ *with* $\mathtt{dom}_{\mathtt{mx}}([\,] \to \tau) = [\bullet]$
· *Erasable args must now be typed*
· *Specify the size of what is erased in* t

Properties of $\mathcal{X}^{\lambda}_{\mathtt{hd/lo}}$

Definition:

- Tight judgment: $\Gamma \vdash^{(\ell,f)} t : \mathtt{tight}$ with Γ tight.
- Tight deriv.: ccl with tight judg. (*local* criterion).

Theorem (HN/WN)

Let $t \in \Lambda$. Then:

$\Gamma \vdash^{(\ell,f)} t : \tau$ tight iff
- $t \to^{\ell}_{\mathtt{hd/lo}} t' HNF$ or NF
- $|t'|_{\mathtt{hd/lo}} = f$

Theorem (SN)

Idem for SN and a **maximal** reduction strategy.

Unique parametrized system

for the 3 strategies

- *Just modify* $\mathtt{dom}(\mathcal{F})$ *with* $\mathtt{dom}_{\mathtt{mx}}([\,] \to \tau) = [\bullet]$
- *Erasable args must now be typed*
- *Specify the size of what is erased in* t

Plan

1. Overview (idempotent or not intersection types)
2. Non-idempotent intersection types
3. Extracting exact lengths of reduction
4. Resources for Classical Logic
5. Exact Measures for $\lambda\mu$
6. Perspectives

The Lambda-Mu Calculus

- Intuit. logic + Peirce's Law $((A \to B) \to A) \to A$
  gives classical logic.

- Intuit. logic + Peirce's Law $((A \to B) \to A) \to A$
  gives classical logic.

- **Griffin 90**: `call–cc` and Felleisen's $\mathcal{C}$-operator typable with Peirce's Law $((A \to B) \to A) \to A$
  $\rightsquigarrow$ the **Curry-Howard** iso extends to classical logic

$$\boxed{\textbf{classical logic}} \longleftrightarrow \boxed{\textbf{backtracking}}$$

- Intuit. logic + Peirce's Law $((A \to B) \to A) \to A$
  gives classical logic.

- **Griffin 90**: `call–cc` and Felleisen's $\mathcal{C}$-operator typable with Peirce's Law $((A \to B) \to A) \to A$
  $\rightsquigarrow$ the **Curry-Howard** iso extends to classical logic

$$\boxed{\textbf{classical logic}} \longleftrightarrow \boxed{\textbf{backtracking}}$$

- **Parigot 92**: $\lambda\mu$-calculus = computational interpretation of classical *natural deduction* (*e.g.*, *vs.* $\bar{\lambda}\mu\tilde{\mu}$).
  judg. of the form $A, A \to B \vdash {\color{red}A} \mid {\color{blue}B}, {\color{blue}C}$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{(A \to B) \to A \vdash (A \to B) \to A} \qquad \dfrac{\dfrac{}{A \vdash A, B}}{\vdash A \to B, A}}{(A \to B) \to A \vdash A, A}}{(A \to B) \to A \vdash A}}{\vdash ((A \to B) \to A) \to A}$$

Standard Style

$$
\dfrac{
  \dfrac{}{(A \to B) \to A \vdash (A \to B) \to A}
  \qquad
  \dfrac{\dfrac{}{A \vdash A, \color{red}{B}}}{\vdash A \to B, A}
}{
  \dfrac{(A \to B) \to A \vdash A, A}{
    \dfrac{(A \to B) \to A \vdash A}{\vdash ((A \to B) \to A) \to A}
  }
}
$$

Standard Style

Peirce's Law in Classical Natural Deduction

$$
\dfrac{\dfrac{\dfrac{}{(A \to B) \to A \vdash (A \to B) \to A \mid} \qquad \dfrac{\dfrac{\dfrac{}{A \vdash A \mid B}}{A \vdash B \mid A}}{\vdash A \to B \mid A}\,\texttt{act}}{\dfrac{(A \to B) \to A \vdash A \mid A}{(A \to B) \to A \vdash A \mid}}}{\vdash ((A \to B) \to A) \to A \mid}
$$

Focussed Style

In the right hand-side of $\Gamma \vdash F \mid \Delta$

- 1 active formula F
- inactive formulas Δ

$$\dfrac{\dfrac{\dfrac{\dfrac{}{(A \to B) \to A \vdash (A \to B) \to A \mid} \qquad \dfrac{\dfrac{\dfrac{}{A \vdash A \mid B}}{A \vdash B \mid A}\ \texttt{act}}{\vdash A \to B \mid A}}{(A \to B) \to A \vdash A \mid A}}{(A \to B) \to A \vdash A \mid}}{\vdash ((A \to B) \to A) \to A \mid}$$

Focussed Style

In the right hand-side of $\Gamma \vdash F \mid \Delta$

- 1 active formula F
- inactive formulas Δ

The $\lambda\mu$-calculus

- **Syntax:** λ-calculus

- **Syntax:** λ-calculus
  + names α, β, γ (store inactive formulas)

$$x_1 : D, y : E \vdash t : C \mid \alpha : A, \beta : B$$

- **Syntax:** λ-calculus

  + names α, β, γ (store inactive formulas)

  $$x_1 : D, y : E \vdash t : C \mid \alpha : A, \beta : B$$

  + two constructors $[\alpha]t$ (naming) and $\mu\alpha$ (μ-abs.)

  de/activation

The $\lambda\mu$-calculus

- **Syntax:** λ-calculus

  + names α, β, γ (store inactive formulas)

$$x_1 : D, y : E \vdash t : C \mid \alpha : A, \beta : B$$

  + two constructors $[\alpha]t$ (naming) and $\mu\alpha$ (μ-abs.)

  de/activation

- Typed and untyped version

  Simply typable $\Rightarrow$ *SN*

The $\lambda\mu$-calculus

- **Syntax:** λ-calculus

  $+$ names α, β, γ (store inactive formulas)

$$x_1 : D, y : E \vdash t : C \mid \alpha : A, \beta : B$$

  $+$ two constructors $[\alpha]t$ (naming) and $\mu\alpha$ (μ-abs.)

  de/activation

- Typed and untyped version

  Simply typable $\Rightarrow$ *SN*

- `call–cc` $:= \lambda y.\mu\alpha.[\alpha]y(\lambda x.\mu\beta.[\alpha]x)$:

- **Syntax:** λ-calculus

  + names α, β, γ (store inactive formulas)

  $$x_1 : D, y : E \vdash t : C \mid \alpha : A, \beta : B$$

  + two constructors $[\alpha]t$ (naming) and $\mu\alpha$ (μ-abs.)

  de/activation

- Typed and untyped version

  Simply typable $\Rightarrow$ *SN*

- `call–cc` $:= \lambda y.\mu\alpha.[\alpha]y(\lambda x.\mu\beta.[\alpha]x) : ((A \to B) \to A) \to A$

- **Syntax:** λ-calculus

  + names α, β, γ (store inactive formulas)

$$x_1 : D, y : E \vdash t : C \mid \alpha : A, \beta : B$$

  + two constructors $[\alpha]t$ (naming) and $\mu\alpha$ (μ-abs.)

  de/activation

- Typed and untyped version

  Simply typable $\Rightarrow$ *SN*

- `call–cc` $:= \lambda y.\mu\alpha.[\alpha]y(\lambda x.\mu\beta.[\alpha]x) : ((A \to B) \to A) \to A$

- β-reduction

$$+ \ (\mu\alpha.[\beta]t)u \to_\mu \mu\alpha.[\beta]t\{u /\!\!/ \alpha\}$$

  where $t\{u /\!\!/ \alpha\}$: replace every $[\alpha]v$ in t by $[\alpha]v\,u$

The $\lambda\mu$-calculus

- **Syntax:** λ-calculus

  + names α, β, γ (store inactive formulas)

  $$x_1 : D, y : E \vdash t : C \mid \alpha : A, \beta : B$$

  + two constructors $[\alpha]t$ (naming) and $\mu\alpha$ (μ-abs.)

  de/activation

- Typed and untyped version

  Simply typable $\Rightarrow$ *SN*

- `call–cc` $:= \lambda y.\mu\alpha.[\alpha]y(\lambda x.\mu\beta.[\alpha]x) : ((A \to B) \to A) \to A$

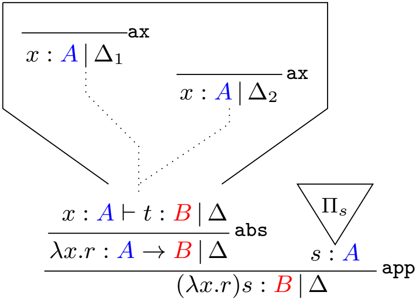
- β-reduction

  $$+ \ (\mu\alpha.[\beta]t)u \to_\mu \mu\alpha.[\beta]t\{u /\!\!/ \alpha\}$$

  where $t\{u /\!\!/ \alpha\}$: replace every $[\alpha]v$ in t by $[\alpha]v\,u$

How do we adapt the non-idempotent machinery to $\lambda\mu$?

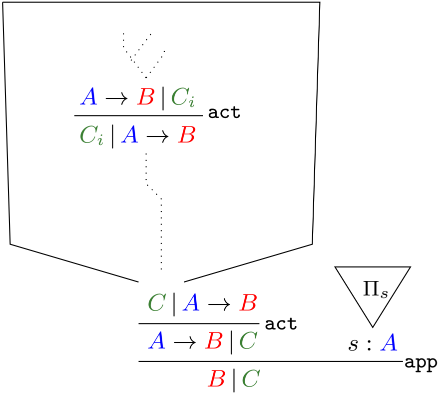
Cut-elimination steps (classical case)



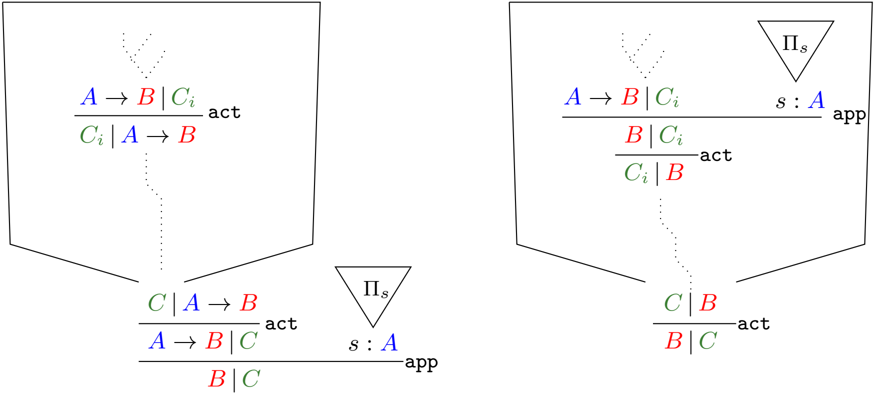
Cut-elimination steps (classical case)

$$\dfrac{\dfrac{C \mid A \to B}{A \to B \mid C}\,\texttt{act} \qquad \dfrac{\Pi_s}{s : A}}{B \mid C}\,\texttt{app}$$

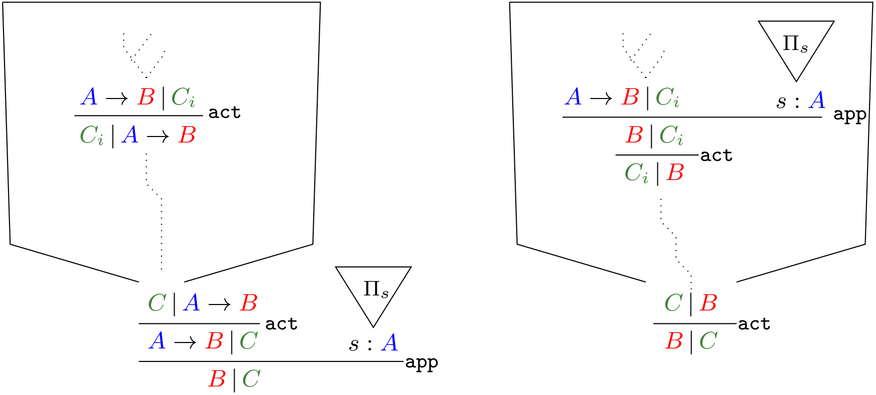
Cut-elimination steps (classical case)



Cut-elimination steps (classical case)



Cut-elimination steps (classical case)



- Duplication of s
- Creation of app-rules
- B saved instead of $A \to B$

The Typing System

Intersection: $\mathcal{I}, \mathcal{J} := [\mathcal{U}_k]_{k \in K}$ $\mathcal{U}, \mathcal{V} =: \langle \sigma_k \rangle_{k \in K}$: **Union**

Intersection: $\mathcal{I}, \mathcal{J} := [\mathcal{U}_k]_{k \in K}$ $\mathcal{U}, \mathcal{V} =: \langle \sigma_k \rangle_{k \in K}$: **Union**

$$x : [\mathcal{U}_1, \mathcal{U}_2];\ y : [\mathcal{V}] \vdash t : \mathcal{U} \mid \alpha : \langle \sigma_1, \sigma_2 \rangle, \beta : \langle \tau_1, \tau_2, \tau_3 \rangle$$

Intersection: $\mathcal{I},\mathcal{J} := [\mathcal{U}_k]_{k\in K}$ $\qquad$ $\mathcal{U},\mathcal{V} =: \langle\sigma_k\rangle_{k\in K}$: **Union**

$$x:[\mathcal{U}_1,\mathcal{U}_2];\ y:[\mathcal{V}] \vdash t:\mathcal{U} \mid \alpha:\langle\sigma_1,\sigma_2\rangle, \beta:\langle\tau_1,\tau_2,\tau_3\rangle$$

Features

Syntax-direction, relevance, multiplicative rules, **accumulation of typing information**.

The Typing System

Intersection: $\mathcal{I}, \mathcal{J} := [\mathcal{U}_k]_{k \in K}$ $\qquad$ $\mathcal{U}, \mathcal{V} =: \langle \sigma_k \rangle_{k \in K}$: **Union**

$$x : [\mathcal{U}_1, \mathcal{U}_2];\ y : [\mathcal{V}] \vdash t : \mathcal{U} \mid \alpha : \langle \sigma_1, \sigma_2 \rangle, \beta : \langle \tau_1, \tau_2, \tau_3 \rangle$$

Features

Syntax-direction, relevance, multiplicative rules, **accumulation of typing information**.

- app-rule based upon the *admissible* rule of ND:

$$\frac{A_1 \to B_1 \vee \ldots \vee A_k \to B_k \qquad A_1 \wedge \ldots \wedge A_k}{B_1 \vee \ldots \vee B_k} \qquad \left(vs. \ \frac{A \to B \quad A}{B} \right)$$

Intersection: $\mathcal{I}, \mathcal{J} := [\mathcal{U}_k]_{k \in K}$ $\qquad$ $\mathcal{U}, \mathcal{V} =: \langle \sigma_k \rangle_{k \in K}$: **Union**

$$x : [\mathcal{U}_1, \mathcal{U}_2];\ y : [\mathcal{V}] \vdash t : \mathcal{U} \mid \alpha : \langle \sigma_1, \sigma_2 \rangle, \beta : \langle \tau_1, \tau_2, \tau_3 \rangle$$

Features

Syntax-direction, relevance, multiplicative rules, **accumulation of typing information**.

- app-rule based upon the *admissible* rule of ND:

$$\frac{A_1 \to B_1 \vee \ldots \vee A_k \to B_k \qquad A_1 \wedge \ldots \wedge A_k}{B_1 \vee \ldots \vee B_k} \qquad \left(vs.\ \frac{A \to B \quad A}{B} \right)$$

$$\boxed{\texttt{call-cc} : [[[A] \to B] \to A] \to \langle A, A \rangle \qquad vs. \qquad ((A \to B) \to A) \to A}$$

Intersection: $\mathcal{I}, \mathcal{J} := [\mathcal{U}_k]_{k \in K}$ $\qquad$ $\mathcal{U}, \mathcal{V} =: \langle \sigma_k \rangle_{k \in K}$: **Union**

$$x : [\mathcal{U}_1, \mathcal{U}_2];\ y : [\mathcal{V}] \vdash t : \mathcal{U} \mid \alpha : \langle \sigma_1, \sigma_2 \rangle, \beta : \langle \tau_1, \tau_2, \tau_3 \rangle$$

Features

Syntax-direction, relevance, multiplicative rules, **accumulation of typing information**.

- app-rule based upon the *admissible* rule of ND:

$$\frac{A_1 \to B_1 \vee \ldots \vee A_k \to B_k \qquad A_1 \wedge \ldots \wedge A_k}{B_1 \vee \ldots \vee B_k} \qquad \left(vs. \ \frac{A \to B \quad A}{B} \right)$$

$$\boxed{\texttt{call-cc} : [[[A] \to B] \to A] \to \langle A, A \rangle \qquad vs. \qquad ((A \to B) \to A) \to A}$$

- **Weighted Subject Reduction + Subject Expansion**

$$\texttt{size}(\Pi) = \begin{cases} \text{number of nodes of } \Pi\ + \\ \text{size of the } \textbf{type arities} \text{ of all the names of commands } + \\ \textbf{multiplicities} \text{ of arguments in all the } \textbf{app. nodes} \end{cases}$$

System $\mathcal{H}_{\lambda\mu}$ (Head Normalization)

- **Weighted Subject Reduction + Subject Expansion**

$$\mathtt{size}(\Pi) = \begin{cases} \text{number of nodes of } \Pi \, + \\ \text{size of the } \textbf{type arities} \text{ of all the names of commands } + \\ \textbf{multiplicities} \text{ of arguments in all the } \textbf{app. nodes} \end{cases}$$

- Characterizes **Head Normalization**

*adaptable to **Strong Normalization***

Theorem [Kesner,V.,FSCD17]:
Let t be a $\lambda\mu$-term. Equiv. between:

- t is $\mathcal{H}_{\lambda\mu}$-typable
- t is HN
- The head red. strategy terminates on t

+ **quantitative info**.

System $\mathcal{H}_{\lambda\mu}$ (Head Normalization)

- **Weighted Subject Reduction + Subject Expansion**

$$\texttt{size}(\Pi) = \begin{cases} \text{number of nodes of } \Pi \,+ \\ \text{size of the } \textbf{type arities} \text{ of all the names of commands} \,+ \\ \textbf{multiplicities} \text{ of arguments in all the } \textbf{app. nodes} \end{cases}$$

- Characterizes **Head Normalization**

*adaptable to **Strong Normalization***

Theorem [Kesner,V.,FSCD17]:
Let t be a $\lambda\mu$-term. Equiv. between:

- t is $\mathcal{H}_{\lambda\mu}$-typable
- t is HN
- The head red. strategy terminates on t

\+ **quantitative info**.

- Small-step version.

Plan

$$\frac{\mathcal{U} \neq \langle\,\rangle}{x : [\mathcal{U}] \vdash^{(0,0,1)} x : \mathcal{U} \mid \emptyset}\ (\mathtt{ax}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \mathcal{U} \mid \Delta}{\Gamma \vdash^{(\ell,m,f)} [\alpha]t \mid \Delta \vee \alpha : \mathcal{U}}\ (\mathtt{c}) \qquad (\wedge)$$

$$(\rightarrow_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \mathtt{Utight}_S \mid \Delta \qquad \mathtt{Itight}_S(\Gamma(x))}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell,m,f+1)} \lambda x.t : \langle \bullet \rangle \mid \Delta}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell,m,f)} c \mid \Delta}{\Gamma \vdash^{(\ell,m+\mathtt{ar}(\Delta(\alpha)^{\uparrow}),f+1+\mathtt{rob}(\Delta(\alpha)))} \mu\alpha.c : \Delta(\alpha)^{\uparrow} \mid \Delta \setminus\!\!\setminus \alpha}\ (\mu)$$

$$\frac{\Gamma_t \vdash^{(\ell_t,m_t,f_t)} t : \mathcal{F} \mid \Delta_t \qquad \Gamma_u \Vdash^{(\ell_u,m_u,f_u)} u : \mathtt{dom}_S(\mathcal{F}) \mid \Delta_u}{\Gamma_t \wedge \Gamma_u \vdash^{(\ell_t+\ell_u,m_t+m_u,f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\,u : \mathtt{codom}(\mathcal{F}) \mid \Delta_t \vee \Delta_u}\ (\mathtt{app}_S)$$

Capturing exact measures in $\lambda\mu$

$$\frac{\mathcal{U} \neq \langle\,\rangle}{x : [\mathcal{U}] \vdash^{(0,0,1)} x : \mathcal{U} \mid \emptyset}\,(\mathtt{ax}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \mathcal{U} \mid \Delta}{\Gamma \vdash^{(\ell,m,f)} [\alpha]t \mid \Delta \vee \alpha : \mathcal{U}}\,(\mathtt{c}) \qquad (\wedge)$$

$$(\rightarrow_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \mathtt{Utight}_S \mid \Delta \qquad \mathtt{Itight}_S(\Gamma(x))}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell,m,f+1)} \lambda x.t : \langle \bullet \rangle \mid \Delta}\,(\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell,m,f)} c \mid \Delta}{\Gamma \vdash^{(\ell,m+\mathtt{ar}(\Delta(\alpha)^{\uparrow}),f+1+\mathtt{rob}(\Delta(\alpha)))} \mu\alpha.c : \Delta(\alpha)^{\uparrow} \mid \Delta \setminus\!\!\setminus \alpha}\,(\mu)$$

$$\frac{\Gamma_t \vdash^{(\ell_t,m_t,f_t)} t : \mathcal{F} \mid \Delta_t \qquad \Gamma_u \Vdash^{(\ell_u,m_u,f_u)} u : \mathtt{dom}_S(\mathcal{F}) \mid \Delta_u}{\Gamma_t \wedge \Gamma_u \vdash^{(\ell_t+\ell_u,m_t+m_u,f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\,u : \mathtt{codom}(\mathcal{F}) \mid \Delta_t \vee \Delta_u}\,(\mathtt{app}_S)$$

- $\mathcal{U}^{\uparrow}$ = transforms top-level $\not\rightarrow$ into $\rightarrow$
  e.g., $([\langle \mathcal{I} \not\rightarrow \mathcal{V} \rangle] \not\rightarrow \bar{\bullet})^{\uparrow} = ([\langle \mathcal{I} \not\rightarrow \mathcal{V} \rangle] \rightarrow \bar{\bullet}$ with $\bar{\bullet} = \langle \bullet \rangle$.

$$\frac{\mathcal{U} \neq \langle\,\rangle}{x : [\mathcal{U}] \vdash^{(0,0,1)} x : \mathcal{U} \mid \emptyset}\ (\mathtt{ax}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \mathcal{U} \mid \Delta}{\Gamma \vdash^{(\ell,m,f)} [\alpha]t \mid \Delta \vee \alpha : \mathcal{U}}\ (\mathtt{c}) \qquad (\wedge)$$

$$(\rightarrow_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \mathtt{Utight}_S \mid \Delta \qquad \mathtt{Itight}_S(\Gamma(x))}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell,m,f+1)} \lambda x.t : \langle \bullet \rangle \mid \Delta}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell,m,f)} c \mid \Delta}{\Gamma \vdash^{(\ell,m+\mathtt{ar}(\Delta(\alpha)^{\uparrow}),f+1+\mathtt{rob}(\Delta(\alpha)))} \mu\alpha.c : \Delta(\alpha)^{\uparrow} \mid \Delta \setminus\!\!\setminus \alpha}\ (\mu)$$

$$\frac{\Gamma_t \vdash^{(\ell_t,m_t,f_t)} t : \mathcal{F} \mid \Delta_t \qquad \Gamma_u \Vdash^{(\ell_u,m_u,f_u)} u : \mathtt{dom}_S(\mathcal{F}) \mid \Delta_u}{\Gamma_t \wedge \Gamma_u \vdash^{(\ell_t+\ell_u,m_t+m_u,f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\,u : \mathtt{codom}(\mathcal{F}) \mid \Delta_t \vee \Delta_u}\ (\mathtt{app}_S)$$

- $\mathcal{U}^{\uparrow}$ = transforms top-level $\nrightarrow$ into $\rightarrow$
  e.g., $([\langle \mathcal{I} \nrightarrow \mathcal{V} \rangle] \nrightarrow \bar{\bullet})^{\uparrow} = ([\langle \mathcal{I} \nrightarrow \mathcal{V} \rangle] \rightarrow \bar{\bullet}$ with $\bar{\bullet} = \langle \bullet \rangle$.

$$\frac{\mathcal{U} \neq \langle\,\rangle}{x : [\mathcal{U}] \vdash^{(0,0,1)} x : \mathcal{U} \mid \emptyset}\ (\texttt{ax}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \mathcal{U} \mid \Delta}{\Gamma \vdash^{(\ell,m,f)} [\alpha]t \mid \Delta \vee \alpha : \mathcal{U}}\ (\texttt{c}) \qquad (\wedge)$$

$$(\rightarrow_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \texttt{Utight}_S \mid \Delta \qquad \texttt{Itight}_S(\Gamma(x))}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell,m,f+1)} \lambda x.t : \langle \bullet \rangle \mid \Delta}\ (\bullet_S)$$

$$\frac{\Gamma \vdash^{(\ell,m,f)} c \mid \Delta}{\Gamma \vdash^{(\ell,m+\texttt{ar}(\Delta(\alpha)^{\uparrow}),f+1+\texttt{rob}(\Delta(\alpha)))} \mu\alpha.c : \Delta(\alpha)^{\uparrow} \mid \Delta \setminus\!\!\setminus \alpha}\ (\mu)$$

$$\frac{\Gamma_t \vdash^{(\ell_t,m_t,f_t)} t : \mathcal{F} \mid \Delta_t \qquad \Gamma_u \Vdash^{(\ell_u,m_u,f_u)} u : \texttt{dom}_S(\mathcal{F}) \mid \Delta_u}{\Gamma_t \wedge \Gamma_u \vdash^{(\ell_t+\ell_u,m_t+m_u,f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\,u : \texttt{codom}(\mathcal{F}) \mid \Delta_t \vee \Delta_u}\ (\texttt{app}_S)$$

- $\mathcal{U}^{\uparrow}$ = transforms top-level $\not\rightarrow$ into $\rightarrow$
  e.g., $([\langle \mathcal{I} \not\rightarrow \mathcal{V} \rangle] \not\rightarrow \bar{\bullet})^{\uparrow} = ([\langle \mathcal{I} \not\rightarrow \mathcal{V} \rangle] \rightarrow \bar{\bullet}$ with $\bar{\bullet} = \langle \bullet \rangle$.
- $\texttt{rob}(\mathcal{U})$ = counts top-level $\not\rightarrow$ *(= number of future @-nodes)*
  e.g., $\texttt{rob}(\langle \mathcal{I} \not\rightarrow \bar{\bullet}, \mathcal{I} \rightarrow \langle \mathcal{I} \not\rightarrow \langle \mathcal{J} \not\rightarrow \bar{\bullet} \rangle\rangle, \bar{\bullet} \rangle) = 3$

$$\frac{\mathcal{U} \neq \langle\,\rangle}{x : [\mathcal{U}] \vdash^{(0,0,1)} x : \mathcal{U} \mid \emptyset}\ (\mathtt{ax}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \mathcal{U} \mid \Delta}{\Gamma \vdash^{(\ell,m,f)} [\alpha]t \mid \Delta \vee \alpha : \mathcal{U}}\ (\mathtt{c}) \qquad (\wedge)$$

$$(\rightarrow_{\mathtt{i}}) \qquad \frac{\Gamma \vdash^{(\ell,m,f)} t : \mathtt{Utight}_{S} \mid \Delta \qquad \mathtt{Itight}_{S}(\Gamma(x))}{\Gamma \setminus\!\!\setminus x \vdash^{(\ell,m,f+1)} \lambda x.t : \langle \bullet \rangle \mid \Delta}\ (\bullet_{S})$$

$$\frac{\Gamma \vdash^{(\ell,m,f)} c \mid \Delta}{\Gamma \vdash^{(\ell,m+\mathtt{ar}(\Delta(\alpha)^{\uparrow}),f+1+\mathtt{rob}(\Delta(\alpha)))} \mu\alpha.c : \Delta(\alpha)^{\uparrow} \mid \Delta \setminus\!\!\setminus \alpha}\ (\mu)$$

$$\frac{\Gamma_t \vdash^{(\ell_t,m_t,f_t)} t : \mathcal{F} \mid \Delta_t \qquad \Gamma_u \Vdash^{(\ell_u,m_u,f_u)} u : \mathtt{dom}_{S}(\mathcal{F}) \mid \Delta_u}{\Gamma_t \wedge \Gamma_u \vdash^{(\ell_t+\ell_u,m_t+m_u,f_t+f_u+\#_{\mathtt{p}}\mathcal{F})} t\,u : \mathtt{codom}(\mathcal{F}) \mid \Delta_t \vee \Delta_u}\ (\mathtt{app}_{S})$$

Parametrization

- Parametrized system (tightness + domains)

Tight types

(spec. normal forms)

- `hd`: empty domains
- `lo/mx`: singleton domains

Domains

(spec. if erasable args are typed)

- $\mathtt{dom}_{\mathtt{hd/lo}}([\,] \to \mathcal{U}) = [\,]$
- $\mathtt{dom}_{\mathtt{mx}}([\,] \to \mathcal{U}) = \text{singleton}$

Parametrization

- Parametrized system (tightness + domains)

Tight types

(spec. normal forms)

- **hd**: empty domains
- **lo**/**mx**: singleton domains

Domains

(spec. if erasable args are typed)

- $\mathtt{dom}_{\mathtt{hd}/\mathtt{lo}}([\,] \to \mathcal{U}) = [\,]$
- $\mathtt{dom}_{\mathtt{mx}}([\,] \to \mathcal{U}) =$ singleton

Theorem (Kesner,V)

let $S \in \{\mathtt{hd}, \mathtt{lo}, \mathtt{mx}\}$ and t a $\lambda\mu$-term. Then:

$t \to_S^{(\ell,m)} t'$ a S-NF with $|t'|_S = f$

iff $\Gamma \vdash^{(\ell,m,f)} t : \mathcal{U} \mid \Delta$ tight for some Γ, $\mathcal{U}$, Δ

- Parametrized system (tightness + domains)

Tight types

(spec. normal forms)

- $\mathtt{hd}$: empty domains
- $\mathtt{lo}/\mathtt{mx}$: singleton domains

Domains

(spec. if erasable args are typed)

- $\mathtt{dom}_{\mathtt{hd}/\mathtt{lo}}([\,] \to \mathcal{U}) = [\,]$
- $\mathtt{dom}_{\mathtt{mx}}([\,] \to \mathcal{U}) =$ singleton

Theorem (Kesner,V)

let $S \in \{\mathtt{hd}, \mathtt{lo}, \mathtt{mx}\}$ and t a $\lambda\mu$-term. Then:

$t \to_S^{(\ell,m)} t'$ a S-NF with $|t'|_S = f$ $\quad$ ($f - e$ when $S = \mathtt{mx}$)

iff $\Gamma \vdash^{(\ell,m,f)} t : \mathcal{U} \mid \Delta$ tight for some Γ, $\mathcal{U}$, Δ

Parametrization

- Parametrized system (tightness + domains)

Tight types

(spec. normal forms)

- $\mathtt{hd}$: empty domains
- $\mathtt{lo}/\mathtt{mx}$: singleton domains

Domains

(spec. if erasable args are typed)

- $\mathtt{dom}_{\mathtt{hd}/\mathtt{lo}}([\,] \to \mathcal{U}) = [\,]$
- $\mathtt{dom}_{\mathtt{mx}}([\,] \to \mathcal{U}) = \text{singleton}$

Theorem (Kesner,V)

let $S \in \{\mathtt{hd}, \mathtt{lo}, \mathtt{mx}\}$ and t a $\lambda\mu$-term. Then:

$t \to_S^{(\ell,m)} t'$ a S-NF with $|t'|_S = f$ $\quad (f - e \text{ when } S = \mathtt{mx})$

iff $\Gamma \vdash^{(\ell,m,f)} t : \mathcal{U} \mid \Delta$ tight for some $\Gamma, \mathcal{U}, \Delta$

Bonus: completely factorized proofs!

Plan

Doggy bag

Doggy bag

Non-idempotency:
forbid duplication of typing deriv.

Doggy bag

Non-idempotency:
forbid duplication of typing deriv.

Simple proofs of termination.

typing brings quali. and quanti. charac.

Doggy bag

Non-idempotency:
forbid duplication of typing deriv.

Simple proofs of termination.

typing brings quali. and quanti. charac.

Very simple
operational semantics

Doggy bag

Non-idempotency:
forbid duplication of typing deriv.

Simple proofs of termination.
typing brings quali. and quanti. charac.

Very simple
operational semantics

Persistence
vs. consumption

Doggy bag

Non-idempotency:
forbid duplication of typing deriv.

Simple proofs of termination.

typing brings quali. and quanti. charac.

Very simple
operational semantics

Persistence
vs. consumption

Exact measures
(red. length + NF)

Doggy bag

Non-idempotency:
forbid duplication of typing deriv.

Simple proofs of termination.
typing brings quali. and quanti. charac.

Very simple
operational semantics

Persistence
vs. consumption

Exact measures
(red. length + NF)

Adapts to other higher-order calculi
e.g., feat. **classical logic**

Non-idempotency:
forbid duplication of typing deriv.

Simple proofs of termination.
typing brings quali. and quanti. charac.

Very simple
operational semantics

Persistence
vs. consumption

Exact measures
(red. length + NF)

Adapts to other higher-order calculi
e.g., feat. **classical logic**

Future work:
$\bar{\lambda}\mu\tilde{\mu}$ (seq. calc.)? Call-by-push-value?

Thank you for your attention!