Strong normalization through idempotent intersection types: a new syntactical approach

Pablo Barenbaum
CONICET
Universidad Nacional de Quilmes
Universidad de Buenos Aires

Simona Ronchi Della Rocca
Università di Torino

Cristian Sottile
CONICET
Universidad de Buenos Aires
Universidad Nacional de Quilmes

41st Conference on Mathematical Foundations of Programming Semantics

MFPS XLI (MFPS 2025)

June 20, 2025

Motivation

To find *simpler* proofs of *strong normalization* for idempotent intersection types

Motivation

To find *simpler* proofs of *strong normalization* for idempotent intersection types

Strong normalization proof techniques

Motivation

To find *simpler* proofs of *strong normalization* for idempotent intersection types

Strong normalization proof techniques

Semantic approach: reducibility candidates/logical relations [Tait'67,Girard'72]

- Define a **denotational semantic** for types based on **termination**
- **Prove soundness of typed terms** w.r.t. the semantics

Motivation

To find *simpler* proofs of *strong normalization* for idempotent intersection types

Strong normalization proof techniques

Semantic approach: reducibility candidates/logical relations [Tait'67,Girard'72]

- Define a **denotational semantic** for types based on **termination**
- **Prove soundness of typed terms** w.r.t. the semantics

Syntactic approach: decreasing measures [Gandy'80, de Vrijer'87]

- Define a **mapping** from terms to a **well founded order**
- Such that it **decreases along reduction**
- TLCA Problem#26 (for STLC, posed by Gödel)

Results

1. A **decreasing measure** based on enriching the calculus with memories

Definition

A mapping

$$\# : \Lambda \to \mathit{WFO}$$

satisfying

$$M \to_\beta N \implies \#(M) > \#(N)$$

Corollary

$$\nexists \quad M_1 \to_\beta M_2 \to_\beta \cdots$$

$$\#(M_1) > \#(M_2) > \cdots$$

Results

1. A **decreasing measure** based on enriching the calculus with memories

   Definition

   A mapping

   $$\# : \Lambda \to \mathit{WFO}$$

   satisfying

   $$M \to_\beta N \implies \#(M) > \#(N)$$

   Corollary

   $$\nexists \quad M_1 \to_\beta M_2 \to_\beta \cdots$$
   $$\#(M_1) > \#(M_2) > \cdots$$

2. An intrinsically typed (*i.e.* **à la Church**) version of **idempotent intersection types**
   - usually presented à la Curry
   - both systems **simulate** each other

$$\begin{array}{ccc} M & \xrightarrow[\beta]{} & N \\ \sqcup & & \sqcup \\ t & \overset{+}{\underset{\mathsf{Ch}}{\dashrightarrow}} & s \end{array} \qquad\qquad \begin{array}{ccccc} t & \xrightarrow[\mathsf{Ch}]{} & s & \overset{*}{\underset{\mathsf{Ch}}{\dashrightarrow}} & s \\ \sqcap & & & & \sqcap \\ M & & \underset{\beta}{\dashrightarrow} & & N \end{array}$$

Idempotent Intersection Types ($\Lambda_{\cap}^{\mathsf{Cu}}$)

[Coppo-Dezzani'79]

Grammar of types

$$\begin{array}{lcll} A & ::= & a \mid \vec{A} \to A & (\vec{A} \neq \varnothing) \\ \vec{A} & ::= & \{A_1, \ldots, A_n\} & (A_i \neq A_j \text{ if } i \neq j) \quad (n \in \mathbb{N}) \end{array}$$

Typing rules

$$\frac{B \in \vec{A}}{\Gamma, x : \vec{A} \vdash_{\mathsf{e}} x : B}\ \texttt{e-var} \qquad \frac{(\Gamma \vdash_{\mathsf{e}} N : A_i)_{i \in I} \quad A_i \neq A_j \text{ if } i \neq j}{\Gamma \Vdash_{\mathsf{e}} N : \{A_1, \ldots, A_n\}}\ \texttt{e-many}$$

$$\frac{\Gamma, x : \vec{A} \vdash_{\mathsf{e}} M : B}{\Gamma \vdash_{\mathsf{e}} \lambda x.M : \vec{A} \to B}\ \texttt{e-I}{\to} \qquad \frac{\Gamma \vdash_{\mathsf{e}} M : \vec{A} \to B \quad \Gamma \Vdash_{\mathsf{e}} N : \vec{A} \quad \vec{A} \neq \varnothing}{\Gamma \vdash_{\mathsf{e}} M\,N : B}\ \texttt{e-E}{\to}$$

Part I: an intrinsically typed idempotent intersection system

Idempotent Intersection Types: a Church presentation ($\Lambda_{\cap}^{\mathsf{Ch}}$)

Why?

- The measure technique is **based on redex degrees** ($\therefore$ on types of subterms)
- So: We need to **handle derivations**
- But: The technique requires **syntactical "intermediate" derivations**
- And: these are **not representable** in the presentation **à la Curry**

À la Curry

$$\dfrac{\dfrac{\dfrac{x : \{A \to A, A\} \vdash_{\mathsf{Cu}} x : A \to A \quad x : \{A \to A, A\} \vdash_{\mathsf{Cu}} x : A}{x : \{A \to A, A\} \vdash_{\mathsf{Cu}} xx : A}}{\vdash_{\mathsf{Cu}} \lambda x.xx : \{A \to A, A\} \to A} \quad \dfrac{\vdash_{\mathsf{Cu}} \lambda x.x : A \to A \quad \vdash_{\mathsf{Cu}} \lambda x.x : A}{\Vdash_{\mathsf{Cu}} \lambda x.x : \{A \to A, A\}}}{\vdash_{\mathsf{Cu}} (\lambda x.xx)(\lambda x.x) : A}$$

Idempotent Intersection Types: a Church presentation ($\Lambda_\cap^{\mathsf{Ch}}$)

À la Curry

$$\dfrac{\dfrac{\dfrac{x : \{A \to A, A\} \vdash_{\mathsf{Cu}} x : A \to A \quad x : \{A \to A, A\} \vdash_{\mathsf{Cu}} x : A}{x : \{A \to A, A\} \vdash_{\mathsf{Cu}} xx : A}}{\vdash_{\mathsf{Cu}} \lambda x.xx : \{A \to A, A\} \to A} \qquad \dfrac{\vdash_{\mathsf{Cu}} \lambda x.x : A \to A \qquad \vdash_{\mathsf{Cu}} \lambda x.x : A}{\Vdash_{\mathsf{Cu}} \lambda x.x : \{A \to A, A\}}}{\vdash_{\mathsf{Cu}} (\lambda x.xx)(\lambda x.x) : A}$$

A linearization inspired by Kfoury's

$$\dfrac{(\Gamma \vdash_{\mathsf{Cu}} N : A_i)_{i \in 1..n} \quad \ldots}{\Gamma \Vdash_{\mathsf{Cu}} N : \{A_1, \ldots, A_n\}} \qquad \Longrightarrow \qquad \dfrac{(\Gamma \vdash_{\mathsf{Ch}} s_i : A_i)_{i \in 1..n} \qquad A_i \neq A_j \text{ if } i \neq j}{\Gamma \Vdash_{\mathsf{Ch}} \{s_1, \ldots, s_n\} : \{A_1, \ldots, A_n\}}$$

Idempotent Intersection Types: a Church presentation ($\Lambda_\cap^{\mathsf{Ch}}$)

À la Curry

$$\dfrac{\dfrac{\dfrac{x : \{A \to A, A\} \vdash_{\mathsf{Cu}} x : A \to A \quad x : \{A \to A, A\} \vdash_{\mathsf{Cu}} x : A}{x : \{A \to A, A\} \vdash_{\mathsf{Cu}} xx : A}}{\vdash_{\mathsf{Cu}} \lambda x.xx : \{A \to A, A\} \to A} \qquad \dfrac{\vdash_{\mathsf{Cu}} \lambda x.x : A \to A \qquad \vdash_{\mathsf{Cu}} \lambda x.x : A}{\Vdash_{\mathsf{Cu}} \lambda x.x : \{A \to A, A\}}}{\vdash_{\mathsf{Cu}} (\lambda x.xx)(\lambda x.x) : A}$$

A linearization inspired by Kfoury's

$$\dfrac{(\Gamma \vdash_{\mathsf{Cu}} N : A_i)_{i \in 1..n} \quad \ldots}{\Gamma \Vdash_{\mathsf{Cu}} N : \{A_1, \ldots, A_n\}} \quad \Longrightarrow \quad \dfrac{(\Gamma \vdash_{\mathsf{Ch}} s_i : A_i)_{i \in 1..n} \qquad A_i \neq A_j \text{ if } i \neq j}{\Gamma \Vdash_{\mathsf{Ch}} \{s_1, \ldots, s_n\} : \{A_1, \ldots, A_n\}}$$

À la Church

$$\rightsquigarrow \qquad (\lambda x^{\{\{A\} \to A, A\}}.x^{\{A\} \to A} x^A)\{\lambda x^{\{A\} \to A}.x, \lambda x^A.x\}$$

Idempotent Intersection Types: a Church presentation ($\Lambda_{\cap}^{\mathsf{Ch}}$)

Substitution

à la Curry

$$(\lambda x. x\ x)\ (\lambda x.x)$$
$$\to_\beta$$
$$(\lambda x.x)\ (\lambda x.x)$$

à la Church

$$(\lambda x^{\{A\to A, A\}}. x^{A\to A}, x^{A})\{\lambda x^{A\to A}.x, \lambda x^{A}.x\}$$
$$\to_{\mathsf{Ch}}$$
$$(\lambda x^{A\to A}.x)\ (\lambda x^{A}.x)$$

Idempotent Intersection Types: a Church presentation ($\Lambda_{\cap}^{\mathsf{Ch}}$)

Substitution

à la Curry

$$(\lambda x.\, x\ x)\,(\lambda x.x)$$
$$\to_\beta$$
$$(\lambda x.x)\ (\lambda x.x)$$

à la Church

$$(\lambda x^{\{A\to A, A\}}.\, x^{A\to A}, x^{A})\{\lambda x^{A\to A}.x, \lambda x^{A}.x\}$$
$$\to_{\mathsf{Ch}}$$
$$(\lambda x^{A\to A}.x)\ (\lambda x^{A}.x)$$

$$(\lambda x.t)s \to_\beta t[s/x] \quad \rightsquigarrow \quad (\lambda x^{\vec{A}}.t)\vec{s} \to_{\mathsf{Ch}} t[s_1/x^{A_1}, \ldots, s_n/x^{A_n}]$$

Idempotent Intersection Types: a Church presentation ($\Lambda_\cap^{\mathsf{Ch}}$)

Substitution

à la Curry		**à la Church**
$(\lambda x. x\ x)(\lambda x.x)$		$(\lambda x^{\{A\to A, A\}}. x^{A\to A}, x^{A})\{\lambda x^{A\to A}.x, \lambda x^{A}.x\}$
$\to_\beta$		$\to_{\mathsf{Ch}}$
$(\lambda x.x)\ (\lambda x.x)$		$(\lambda x^{A\to A}.x)\ (\lambda x^{A}.x)$
$(\lambda x.t)s \to_\beta t[s/x]$	$\rightsquigarrow$	$(\lambda x^{\vec{A}}.t)\vec{s} \to_{\mathsf{Ch}} t[s_1/x^{A_1}, \ldots, s_n/x^{A_n}]$

Bijection between a set–term and its set–type

$\Rightarrow$) given $s' \in \vec{s}$, it has type some type A' by `i-many`, which is **unique** (à la Church typing)

$\Leftarrow$) given $A' \in \vec{A}$, **by injectivity** ($A_i \neq A_j$ if $i \neq j$) there is **only one derivation** $\Gamma \vdash_{\mathsf{Ch}} s' : A'$

Relating $\Lambda_\cap^{\mathsf{Cu}}$ and $\Lambda_\cap^{\mathsf{Ch}}$

Reduction difference Inside the argument of an application

$$t\{s_1, s_2, \ldots, s_n\} \to_{\mathsf{Ch}} t\{s'_1, s_2, \ldots, s_n\} \to_{\mathsf{Ch}} t\{s'_1, s'_2, \ldots, s_n\} \to_{\mathsf{Ch}} \ldots \to_{\mathsf{Ch}} t\{s'_1, s'_2, \ldots, s'_n\}$$

Relating $\Lambda_{\cap}^{\mathsf{Cu}}$ and $\Lambda_{\cap}^{\mathsf{Ch}}$

Reduction difference Inside the argument of an application

$$t\{s_1, s_2, \ldots, s_n\} \to_{\mathsf{Ch}} t\{s'_1, s_2, \ldots, s_n\} \to_{\mathsf{Ch}} t\{s'_1, s'_2, \ldots, s_n\} \to_{\mathsf{Ch}} \ldots \to_{\mathsf{Ch}} t\{s'_1, s'_2, \ldots, s'_n\}$$

Relating terms and derivations

$$x^D\{\, I^A I^B \,,\, I^B I^C \,\} \quad \sqsubset \quad x(II)$$

$$x^D\{\, I^A I^B \,,\, I^C \,\} \qquad x^D\{\, I^B \,,\, I^B I^C \,\}$$

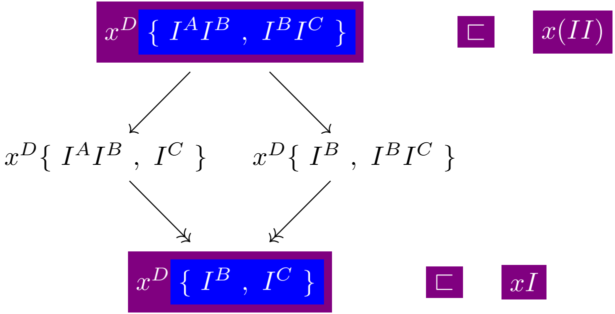
$$x^D\{\, I^B \,,\, I^C \,\} \quad \sqsubset \quad xI$$

Uniformity
set–term with "equal" subterms

Refinement
relate uniform set–terms with terms

Relating $\Lambda_{\cap}^{\mathsf{Cu}}$ and $\Lambda_{\cap}^{\mathsf{Ch}}$

Relating terms and derivations



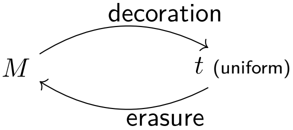
Uniformity

set–term with "equal" subterms

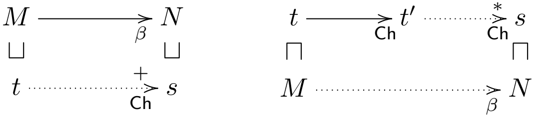
Refinement

relate uniform set–terms with terms

Correspondence



Simulation

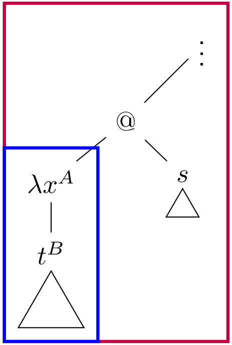


Part II: a decreasing measure for IIT

Redex degrees

Definition

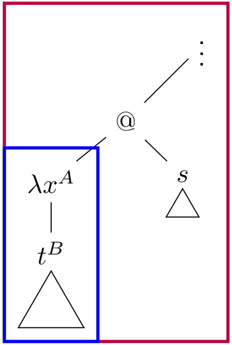
degree of the type of its abstraction



deg$(A \to B)$

Redex degrees

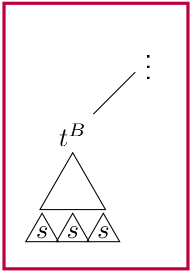
Definition
degree of the type of its abstraction



$deg(A \to B)$

$\to_\beta$

Turing's observation
a redex contraction can only create smaller degree redexes



$deg(R \text{ new}) < deg(A \to B)$

The memory $\Lambda_\cap^{\mathsf{Ch}}$

Definition

$$t \;::=\; x^{\vec{A}} \mid \lambda x.t \mid t\,\vec{t} \mid t\langle\vec{t}\rangle \qquad (\lambda x^{\vec{A}}.t)\vec{s} \rightarrow_m t[\vec{s}/x^{\vec{A}}]\langle\vec{s}\rangle$$

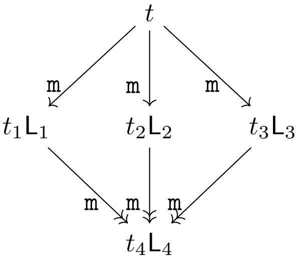
The memory $\Lambda_{\cap}^{\mathsf{Ch}}$

Definition

$$t \ ::= \ x^{\vec{A}} \mid \lambda x.t \mid t\,\vec{t} \mid t\langle\vec{t}\rangle \qquad (\lambda x^{\vec{A}}.t)\vec{s} \rightarrow_m t[\vec{s}/x^{\vec{A}}]\langle\vec{s}\rangle$$

Why not to erase?

- **Retain information**



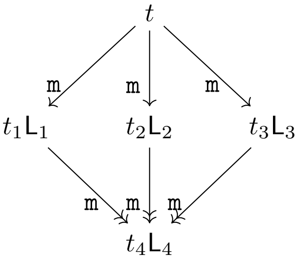
The memory $\Lambda_\cap^{\mathsf{Ch}}$

Definition

$$t \ ::= \ x^{\vec{A}} \mid \lambda x.t \mid t\,\vec{t} \mid t\langle\vec{t}\rangle \qquad (\lambda x^{\vec{A}}.t)\vec{s} \to_m t[\vec{s}/x^{\vec{A}}]\langle\vec{s}\rangle$$

Why not to erase?

- Retain information



WCR

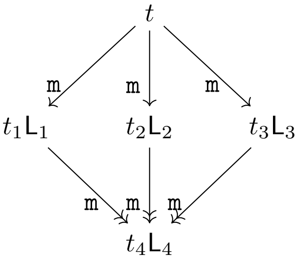
The memory $\Lambda_{\cap}^{\mathsf{Ch}}$

Definition

$$t \ ::= \ x^{\vec{A}} \mid \lambda x.t \mid t\,\vec{t} \mid t\langle\vec{t}\rangle \qquad (\lambda x^{\vec{A}}.t)\vec{s} \to_m t[\vec{s}/x^{\vec{A}}]\,\langle\vec{s}\rangle$$

Why not to erase?

- **Retain information**



WCR

Example

Let $A = \{a\} \to a$, $B = \{A\} \to A$ and $C = \{B\} \to B$:

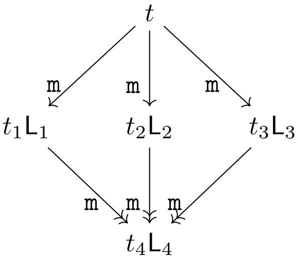
The memory $\Lambda_\cap^{\mathsf{Ch}}$

Definition

$$t \ ::= \ x^{\vec{A}} \mid \lambda x.t \mid t\,\vec{t} \mid t\langle \vec{t}\rangle \qquad (\lambda x^{\vec{A}}.t)\vec{s} \to_m t[\vec{s}/x^{\vec{A}}]\langle \vec{s}\rangle$$

Why not to erase?

- Retain information



Example

Let $A = \{a\} \to a$, $B = \{A\} \to A$ and $C = \{B\} \to B$:

$$(\lambda x^{\{B,A\}}.x^B x^A)\{I^C I^B, I^B I^A\}$$

WCR

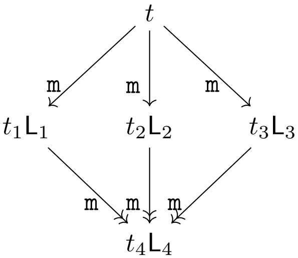
The memory $\Lambda_\cap^{\mathsf{Ch}}$

Definition

$$t \ ::= \ x^{\vec{A}} \mid \lambda x.t \mid t\,\vec{t} \mid t\langle\vec{t}\rangle \qquad (\lambda x^{\vec{A}}.t)\vec{s} \to_m t[\vec{s}/x^{\vec{A}}]\langle\vec{s}\rangle$$

Why not to erase?

- Retain information



Example

Let $A = \{a\} \to a$, $B = \{A\} \to A$ and $C = \{B\} \to B$:

$$(\lambda x^{\{B,A\}}.x^B x^A)\{I^C I^B, I^B I^A\}$$

$$\to_m \quad (\lambda x^{\{B,A\}}.x^B x^A)\{I^B\langle I^B\rangle, I^B I^A\}$$

WCR

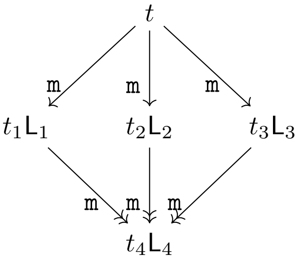
The memory $\Lambda_{\cap}^{\mathsf{Ch}}$

Definition

$$t \ ::= \ x^{\vec{A}} \mid \lambda x.t \mid t\,\vec{t} \mid t\langle\vec{t}\rangle \qquad (\lambda x^{\vec{A}}.t)\vec{s} \to_m t[\vec{s}/x^{\vec{A}}]\langle\vec{s}\rangle$$

Why not to erase?

- **Retain information**



Example

Let $A = \{a\} \to a$, $B = \{A\} \to A$ and $C = \{B\} \to B$:

$$(\lambda x^{\{B,A\}}.x^B x^A)\{I^C I^B, I^B I^A\}$$

$$\to_m \quad (\lambda x^{\{B,A\}}.x^B x^A)\{I^B\langle I^B\rangle, I^B I^A\}$$

$$\to_m \quad (\lambda x^{\{B,A\}}.x^B x^A)\{I^B\langle I^B\rangle, I\langle I\rangle^A\}$$

WCR

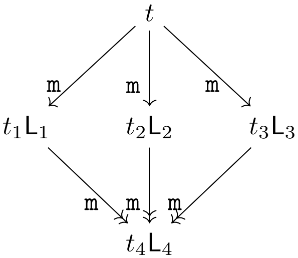
The memory $\Lambda_{\cap}^{\mathsf{Ch}}$

Definition

$$t \ ::= \ x^{\vec{A}} \mid \lambda x.t \mid t\,\vec{t} \mid t\langle\vec{t}\rangle \qquad (\lambda x^{\vec{A}}.t)\vec{s} \rightarrow_m t[\vec{s}/x^{\vec{A}}]\,\langle\vec{s}\rangle$$

Why not to erase?

- **Retain information**



Example

Let $A = \{a\} \rightarrow a$, $B = \{A\} \rightarrow A$ and $C = \{B\} \rightarrow B$:

$$(\lambda x^{\{B,A\}}.x^B x^A)\{I^C I^B, I^B I^A\}$$

$$\rightarrow_m \quad (\lambda x^{\{B,A\}}.x^B x^A)\{I^B\langle I^B\rangle, I^B I^A\}$$

$$\rightarrow_m \quad (\lambda x^{\{B,A\}}.x^B x^A)\{I^B\langle I^B\rangle, I\langle I\rangle^A\}$$

$$\rightarrow_m \quad (I^B\langle I^B\rangle)(I\langle I\rangle)\,\langle\{I^B\langle I^B\rangle, I\langle I\rangle^A\}\rangle$$

WCR

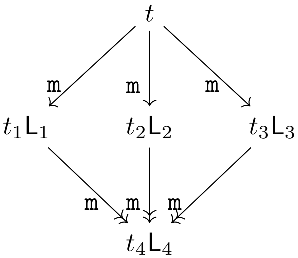
The memory $\Lambda_\cap^{\mathsf{Ch}}$

Definition

$$t \ ::= \ x^{\vec{A}} \mid \lambda x.t \mid t\,\vec{t} \mid t\langle\vec{t}\rangle \qquad (\lambda x^{\vec{A}}.t)\vec{s} \to_m t[\vec{s}/x^{\vec{A}}]\,\langle\vec{s}\rangle$$

Why not to erase?

- Retain information



WCR

Example

Let $A = \{a\} \to a$, $B = \{A\} \to A$ and $C = \{B\} \to B$:

$$
\begin{aligned}
& (\lambda x^{\{B,A\}}.x^B x^A)\{I^C I^B, I^B I^A\} \\
\to_m \quad & (\lambda x^{\{B,A\}}.x^B x^A)\{I^B\langle I^B\rangle, I^B I^A\} \\
\to_m \quad & (\lambda x^{\{B,A\}}.x^B x^A)\{I^B\langle I^B\rangle, I\langle I\rangle^A\} \\
\to_m \quad & (I^B\langle I^B\rangle)(I\langle I\rangle)\langle\{I^B\langle I^B\rangle, I\langle I\rangle^A\}\rangle \\
\to_m \quad & I\langle I^A\rangle\langle I\langle I\rangle^A\rangle\ \langle\{I^B\langle I^B\rangle, I\langle I\rangle^A\}\rangle
\end{aligned}
$$

$\mathcal{W}_\cap$-measure: definition

Operations

- **weight** of a term:
  $\mathsf{w}(t)$ = amount of memories *e.g.* $I\langle I\rangle\langle I\langle I\rangle\rangle\ \langle\{I^2\langle I^2\rangle, I\langle I\rangle\}\rangle = 6$

$\mathcal{W}_\cap$-measure: definition

Operations

- **weight** of a term:
  $\mathsf{w}(t)$ = amount of memories $\qquad$ *e.g.* $I\langle I\rangle\langle I\langle I\rangle\rangle\ \langle\{I^2\langle I^2\rangle, I\langle I\rangle\}\rangle = 6$
- **simplification** of a term for degree d:
  $\mathsf{S}_d(t)$ = "bottom-up" "contraction" of all d redexes $\qquad$ (def. by structural recursion)

$$\mathsf{S}_d((\lambda x^{\vec{A}}.t')\mathtt{L}) \;=\; \mathsf{S}_d(t')\,[\,\mathsf{S}_d(\vec{s})\,/x^{\vec{A}}]\langle\,\mathsf{S}_d(\vec{s})\,\rangle\,\mathsf{S}_d(\mathtt{L}) \qquad \text{if it is of degree } d$$

$\mathcal{W}_\cap$-measure: definition

Operations

- **weight** of a term:
  $\mathsf{w}(t)$ = amount of memories *e.g.* $I\langle I\rangle\langle I\langle I\rangle\rangle\ \langle\{I^2\langle I^2\rangle, I\langle I\rangle\}\rangle = 6$

- **simplification** of a term for degree d:
  $\mathsf{S}_d(t)$ = "bottom-up" "contraction" of all d redexes (def. by structural recursion)

$$\mathsf{S}_d((\lambda x^{\vec{A}}.t')\mathtt{L}) \ = \ \mathsf{S}_d(t')\,[\,\mathsf{S}_d(\vec{s})\,/x^{\vec{A}}]\langle\,\mathsf{S}_d(\vec{s})\,\rangle\,\mathsf{S}_d(\mathtt{L}) \qquad \text{if it is of degree } d$$

- **full simplification**

$$\mathsf{S}_*(t) = \mathsf{S}_1(\dots\mathsf{S}_{\mathsf{maxdeg}}(t)\dots)$$

$\mathcal{W}_\cap$-measure: definition

Operations

- **weight** of a term:
  $\mathsf{w}(t)$ = amount of memories *e.g.* $I\langle I\rangle\langle I\langle I\rangle\rangle\ \langle\{I^2\langle I^2\rangle, I\langle I\rangle\}\rangle = 6$

- **simplification** of a term for degree d:
  $\mathsf{S}_d(t)$ = "bottom-up" "contraction" of all d redexes (def. by structural recursion)

$$\mathsf{S}_d((\lambda x^{\vec{A}}.t')\mathtt{L}) = \mathsf{S}_d(t')\,[\,\mathsf{S}_d(\vec{s})\,/x^{\vec{A}}]\langle\,\mathsf{S}_d(\vec{s})\,\rangle\,\mathsf{S}_d(\mathtt{L}) \qquad \text{if it is of degree } d$$

- **full simplification**

$$\mathsf{S}_*(t) = \mathsf{S}_1(\ldots\mathsf{S}_{\mathsf{maxdeg}}(t)\ldots)$$

$\mathcal{W}_\cap$-measure

t

$\mathcal{W}_\cap$-measure: definition

Operations

- **weight** of a term:
  $\mathsf{w}(t)$ = amount of memories   *e.g.* $I\langle I\rangle\langle I\langle I\rangle\rangle\ \langle\{I^2\langle I^2\rangle, I\langle I\rangle\}\rangle = 6$

- **simplification** of a term for degree d:
  $\mathsf{S}_d(t)$ = "bottom-up" "contraction" of all d redexes   (def. by structural recursion)

$$\mathsf{S}_d((\lambda x^{\vec{A}}.t')\mathtt{L}) = \mathsf{S}_d(t')[\mathsf{S}_d(\vec{s})/x^{\vec{A}}]\langle\mathsf{S}_d(\vec{s})\rangle\mathsf{S}_d(\mathtt{L}) \qquad \text{if it is of degree } d$$

- **full simplification**

$$\mathsf{S}_*(t) = \mathsf{S}_1(\ldots\mathsf{S}_{\mathsf{maxdeg}}(t)\ldots)$$

$\mathcal{W}_\cap$-measure

$t \qquad \mathsf{S}_*(t)$

$\mathcal{W}_\cap$-measure: definition

Operations

- **weight** of a term:
  $\mathsf{w}(t)$ = amount of memories *e.g.* $I\langle I\rangle\langle I\langle I\rangle\rangle\ \langle\{I^2\langle I^2\rangle, I\langle I\rangle\}\rangle = 6$

- **simplification** of a term for degree d:
  $\mathsf{S}_d(t)$ = "bottom-up" "contraction" of all d redexes (def. by structural recursion)

$$\mathsf{S}_d((\lambda x^{\vec{A}}.t')\mathtt{L}) = \mathsf{S}_d(t')[\mathsf{S}_d(\vec{s})/x^{\vec{A}}]\langle\mathsf{S}_d(\vec{s})\rangle\mathsf{S}_d(\mathtt{L})$$ if it is of degree d

- **full simplification**

$$\mathsf{S}_*(t) = \mathsf{S}_1(\ldots\mathsf{S}_{\mathsf{maxdeg}}(t)\ldots)$$

$\mathcal{W}_\cap$-measure

$$t \qquad \mathsf{S}_*(t) = t'\mathsf{L}_t$$

$\mathcal{W}_\cap$-measure: definition

Operations

- **weight** of a term:
  $\mathsf{w}(t)$ = amount of memories *e.g.* $I\langle I\rangle\langle I\langle I\rangle\rangle\ \langle\{I^2\langle I^2\rangle, I\langle I\rangle\}\rangle = 6$
- **simplification** of a term for degree d:
  $\mathsf{S}_d(t)$ = "bottom-up" "contraction" of all d redexes (def. by structural recursion)

$$\mathsf{S}_d((\lambda x^{\vec{A}}.t')\mathtt{L}) \;=\; \mathsf{S}_d(t')\,[\,\mathsf{S}_d(\vec{s})\,/x^{\vec{A}}]\langle\,\mathsf{S}_d(\vec{s})\,\rangle\,\mathsf{S}_d(\mathtt{L}) \qquad \text{if it is of degree } d$$

- **full simplification**

$$\mathsf{S}_*(t) = \mathsf{S}_1(\ldots\mathsf{S}_{\mathsf{maxdeg}}(t)\ldots)$$

$\mathcal{W}_\cap$-measure

$$t \qquad \mathsf{S}_*(t) \quad = \quad t'\mathsf{L}_t \qquad \mathsf{w}(t'\mathsf{L}_t)$$

$\mathcal{W}_\cap$-measure: definition

Operations

- **weight** of a term:
  $\mathsf{w}(t)$ = amount of memories   *e.g.* $I\langle I\rangle\langle I\langle I\rangle\rangle\ \langle\{I^2\langle I^2\rangle, I\langle I\rangle\}\rangle = 6$

- **simplification** of a term for degree d:
  $\mathsf{S}_d(t)$ = "bottom-up" "contraction" of all d redexes   (def. by structural recursion)

$$\mathsf{S}_d((\lambda x^{\vec{A}}.t')\mathtt{L}) \;=\; \mathsf{S}_d(t')\,[\,\mathsf{S}_d(\vec{s})\,/x^{\vec{A}}]\langle\,\mathsf{S}_d(\vec{s})\,\rangle\,\mathsf{S}_d(\mathtt{L}) \qquad \text{if it is of degree } d$$

- **full simplification**

$$\mathsf{S}_*(t) = \mathsf{S}_1(\ldots\mathsf{S}_{\mathsf{maxdeg}}(t)\ldots)$$

$\mathcal{W}_\cap$-measure

$t \qquad \mathsf{S}_*(t) \;=\; t'\mathsf{L}_t \qquad \mathsf{w}(t'\mathsf{L}_t)$

$$\mathcal{W}_\cap(t) \;=\; \mathsf{w}(\mathsf{S}_*(t))$$

$\mathcal{W}_{\cap}$-measure: proof

Properties

- Reduction arrives at simplification $t \rightarrow^*_m \mathsf{S}_*(t)$

$\mathcal{W}_\cap$-measure: proof

Properties

- Reduction arrives at simplification $t \to_m^* \mathsf{S}_*(t)$
- Simplification is normal form $\mathsf{S}_*(t) = \mathtt{nf}(t)$

$\mathcal{W}_\cap$-measure: proof

Properties

- Reduction arrives at simplification $t \to_m^* \mathsf{S}_*(t)$
- Simplification is normal form $\mathsf{S}_*(t) = \mathtt{nf}(t)$
- Max–degree simplification decreases max–degree $D_{max}(t) > D_{max}(\mathsf{S}_{D_{max}}(t))$

$\mathcal{W}_\cap$-measure: proof

Properties

- Reduction arrives at simplification $t \to_m^* \mathsf{S}_*(t)$
- Simplification is normal form $\mathsf{S}_*(t) = \mathtt{nf}(t)$
- Max–degree simplification decreases max–degree $D_{max}(t) > D_{max}(\mathsf{S}_{D_{max}}(t))$

Theorem: $\mathcal{W}_\cap$ decreases along reduction

$$t \to_{\mathsf{Ch}} s \quad \Longrightarrow \quad \mathcal{W}_\cap(t) > \mathcal{W}_\cap(s)$$

$\mathcal{W}_\cap$-measure: proof

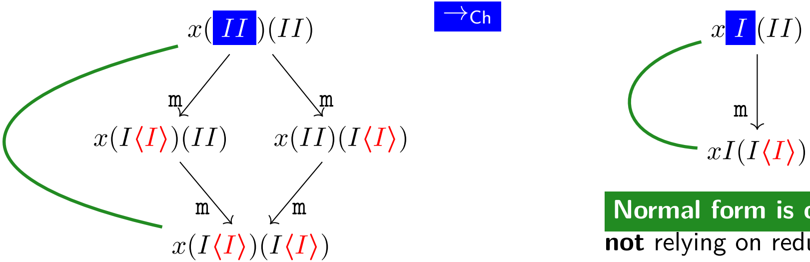
Properties

- Reduction arrives at simplification $\quad t \to_m^* \mathsf{S}_*(t)$
- Simplification is normal form $\quad \mathsf{S}_*(t) = \mathtt{nf}(t)$
- Max–degree simplification decreases max–degree $\quad D_{max}(t) > D_{max}(\mathsf{S}_{D_{max}}(t))$

Theorem: $\mathcal{W}_\cap$ decreases along reduction

$$t \to_{\mathsf{Ch}} s \quad \Longrightarrow \quad \mathcal{W}_\cap(t) > \mathcal{W}_\cap(s)$$

Intuitively



Normal form is obtained through S_*
not relying on reduction

Part III: Conclusion

Conclusion: lifting the result

$\mathcal{W}^{\mathsf{Cu}}_{\cap}$-measure $\mathcal{W}^{\mathsf{Cu}}_{\cap}(M) = \mathsf{w}(\mathsf{S}_*(M^{\mathsf{Ch}}))$

Conclusion: lifting the result

$\mathcal{W}_\cap^{\mathsf{Cu}}$-measure $\mathcal{W}_\cap^{\mathsf{Cu}}(M) = \mathsf{w}(\mathsf{S}_*(M^{\mathsf{Ch}}))$

$\mathcal{W}_\cap^{\mathsf{Cu}}$ decreases along reduction $M \to N \implies \mathcal{W}_\cap^{\mathsf{Cu}}(M) > \mathcal{W}_\cap^{\mathsf{Cu}}(N)$

$$
\begin{array}{ccc}
\boldsymbol{M} & \to_\beta & \boldsymbol{N} \\
\sqcup & & \sqcup \\
t & \to^+_{\mathsf{Ch}} & s \\
| & & | \\
\mathcal{W}_\cap(t) & & \mathcal{W}_\cap(s)
\end{array}
$$

Conclusion: lifting the result

$\mathcal{W}_\cap^{\mathsf{Cu}}$-measure $\mathcal{W}_\cap^{\mathsf{Cu}}(M) = \mathsf{w}(\mathsf{S}_*(M^{\mathsf{Ch}}))$

$\mathcal{W}_\cap^{\mathsf{Cu}}$ decreases along reduction $M \to N \implies \mathcal{W}_\cap^{\mathsf{Cu}}(M) > \mathcal{W}_\cap^{\mathsf{Cu}}(N)$

$$\begin{array}{ccc} \boldsymbol{M} & \to_\beta & \boldsymbol{N} \\ \sqcup & & \sqcup \\ t & \to^+_{\mathsf{Ch}} & s \\ | & & | \\ \mathcal{W}_\cap(t) & & \mathcal{W}_\cap(s) \end{array}$$

$\Lambda_\cap^{\mathsf{Cu}}$ is strongly normalizing $\Gamma \vdash_{\mathsf{Cu}} M : A \implies M \in SN$

Related works

Existing decreasing measures for idempotent intersection types

Related works

Existing decreasing measures for idempotent intersection types

[Kfoury & Wells'95]

- **Domain of DM:** **multiset of natural numbers**
- **Methodology:** **indirect**, *i.e.* decreases for specific strategy
- **Base calculus:** à la Church, ad hoc

[Boudol'03]

- **Domain of DM:** **pair of natural numbers**
- **Methodology:** **indirect**, *i.e.* decreases for specific strategy
- **Base calculus:** à la Curry

Related works

Existing decreasing measures for idempotent intersection types

[Kfoury & Wells'95]

- **Domain of DM: multiset of natural numbers**
- **Methodology: indirect**, *i.e.* decreases for specific strategy
- **Base calculus:** à la Church, ad hoc

[Boudol'03]

- **Domain of DM: pair of natural numbers**
- **Methodology: indirect**, *i.e.* decreases for specific strategy
- **Base calculus:** à la Curry

Our proposal

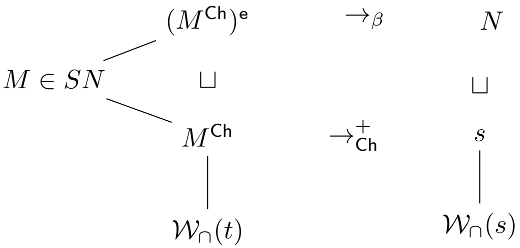
- **Domain of DM: natural number**
- **Methodology:** DM proving SN **(direct)**
- **Base calculus:** à la Church, **proven *in simulation*** with Curry

Future work

- **Refinement** of the measure **to exactness**, *i.e.* such that $\mathcal{W}_\cap(M)$ is the amount of reduction steps of the longest reduction chain

- **Adaptation** of the technique to the idempotent intersection type system characterizing ***head normalization***

Conclusion


$(M^{\mathsf{Ch}})^{\mathsf{e}}$
$\rightarrow_\beta$
N
$M \in SN$
$\sqcup$
$\sqcup$
M^{Ch}
$\rightarrow^+_{\mathsf{Ch}}$
s
$\mathcal{W}_\cap(t)$
$\mathcal{W}_\cap(s)$