Logic and Computation Sessions

XVI Summer Workshop in Mathematics, Universidade de Brasília

Quantitative Weak Linearisation

Daniel Ventura
EFFA/INF, Universidade Federal de Goiás
(joint work with Sandra Alves)

February 2024

Quantitative Weak Linearisation

Linearisation: What do we mean by linearisation:

"Linearisation as the process of transforming/relating/simulating non-linear functions in/to/using **equivalent** *linear functions"*

Weak: We consider a restricted class of linear terms:

"A λ-term t is weak-linear if every β-redex in any reduction sequence starting from t are non duplicating".

Quantitative: Non-idempotent intersection types, introduced independently by Gardner and Kfoury. Its relation with linear logic was highlighted in De Carvalho's thesis.

Quantitative Types

The λ-calculus

Proposed by Church in 1932[1].

$$\textit{Terms} \qquad t := x \mid t\,t \mid \lambda x.t$$

Computations (reductions) executed by a unique rule:

$$(\lambda x.t)\, s \longrightarrow t\{x\backslash s\} \qquad (\beta)$$

Some renaming may be needed:

$$\lambda x.t \longrightarrow \lambda y.t\{x\backslash y\} \qquad (\alpha)$$

---

[1] A. Church: A set of postulates for the foundation of logic. *Annals of Math* 33(2):346–366, 1932.

Intersection Types Systems (ITS)

Terms in an ITS can have more than one type:

$$x : \alpha \to \beta \cap \alpha$$

where $\cap$ is commutative, associative and idempotent:

$$\tau \cap \tau = \tau$$

$$\dfrac{\dfrac{\dfrac{}{x : \{\alpha \to \beta\} \vdash x : \alpha \to \beta} \quad \dfrac{}{x : \{\alpha\} \vdash x : \alpha}}{x : \{\alpha \to \beta, \alpha\} \vdash xx : \beta}}{\vdash \lambda x.xx : \{\alpha \to \beta, \alpha\} \to \beta}$$

Intersection Types Systems (ITS)

Terms in an ITS can have more than one type:

$$x : \alpha \rightarrow \beta \cap \alpha$$

where $\cap$ is commutative, associative and idempotent:

$$\tau \cap \tau = \tau$$

$$\dfrac{\dfrac{}{x : \{\alpha \rightarrow \beta\} \vdash x : \alpha \rightarrow \beta} \quad \dfrac{}{x : \{\alpha\} \vdash x : \alpha}}{\dfrac{x : \{\alpha \rightarrow \beta, \alpha\} \vdash xx : \beta}{\vdash \lambda x.xx : \{\alpha \rightarrow \beta, \alpha\} \rightarrow \beta}}$$

ITS for Strong Normalising Terms

$$(\textbf{types}) \quad \sigma, \tau \quad ::= \quad \alpha \mid \mathcal{R} \rightarrow \sigma \qquad (\textbf{int-types}) \quad \mathcal{R} \quad ::= \quad \{\sigma_k\}_{k \in K}$$

$$\frac{\Gamma \vdash t : \tau}{\Gamma \setminus\!\!\setminus x \vdash \lambda x.t : \Gamma(x) \rightarrow \tau} \qquad \frac{}{x : \{\tau\} \vdash x : \tau}$$

$$\frac{\Gamma \vdash t : \mathcal{R} \rightarrow \tau \qquad \Delta \vdash u : \mathcal{R}}{\Gamma + \Delta \vdash t\,u : \tau} \qquad \frac{\Delta \vdash t : \sigma}{\Delta \vdash t : \{\}}$$

$$\frac{(\Delta_k \vdash t : \sigma_k)_{k \in K} \qquad |K| > 0}{+_{k \in K} \Delta_k \vdash t : \{\sigma_k\}_{k \in K}}$$

Quantitative information is obtained with a non-idempotent $\cap$:

$$\tau \cap \tau \neq \tau$$

Idempotent	Non-idempotent
$\{x : \sigma \to \sigma \to \tau, y : \sigma\} \vdash xyy : \tau$	$\{x : \sigma \to \sigma \to \tau, y : \sigma \cap \sigma\} \vdash xyy : \tau$

For $(\lambda x.\lambda y.xyy)uv$ there is a single derivation for v in the idempotent system, but two copies in its reduct uvv

Reduction decreases the size of derivations in the non-idempotent system

Quantitative Types

Quantitative information is obtained with a non-idempotent $\cap$:

$$\tau \cap \tau \neq \tau$$

Idempotent	Non-idempotent
$\{x : \sigma \to \sigma \to \tau, y : \sigma\} \vdash xyy : \tau$	$\{x : \sigma \to \sigma \to \tau, y : \sigma \cap \sigma\} \vdash xyy : \tau$

For $(\lambda x.\lambda y.xyy)uv$ there is a single derivation for v in the idempotent system, but two copies in its reduct uvv

Reduction decreases the size of derivations in the non-idempotent system

Quantitative Types

Quantitative information is obtained with a non-idempotent $\cap$:

$$\tau \cap \tau \neq \tau$$

Idempotent	**Non-idempotent**
$\{x : \sigma \to \sigma \to \tau, y : \sigma\} \vdash xyy : \tau$	$\{x : \sigma \to \sigma \to \tau, y : \sigma \cap \sigma\} \vdash xyy : \tau$

For $(\lambda x.\lambda y.xyy)uv$ there is a single derivation for v in the idempotent system, but two copies in its reduct uvv

Reduction decreases the size of derivations in the non-idempotent system

NITS for Strong Normalising Terms

$$\textbf{(types)} \quad \sigma, \tau \quad ::= \quad \alpha \mid \mathcal{A} \to \sigma \quad \textbf{(multi-types)} \quad \mathcal{A} \quad ::= \quad [\sigma_k]_{k \in K}$$

$$\frac{\Gamma \vdash t : \tau}{\Gamma \setminus\!\!\setminus x \vdash \lambda x.t : \Gamma(x) \to \tau} \qquad \frac{}{x : [\tau] \vdash x : \tau}$$

$$\frac{\Gamma \vdash t : \mathcal{A} \to \tau \qquad \Delta \vdash u : \mathcal{A}}{\Gamma + \Delta \vdash t\, u : \tau} \qquad \frac{\Delta \vdash t : \sigma}{\Delta \vdash t : [\,]}$$

$$\frac{(\Delta_k \vdash t : \sigma_k)_{k \in K} \qquad |K| > 0}{+_{k \in K} \Delta_k \vdash t : [\sigma_k]_{k \in K}}$$

Quantitative Types

- Antonio Bucciarelli, Delia Kesner, Daniel Ventura: Non-idempotent intersection types for the Lambda-Calculus. Log. J. IGPL 25(4): 431-464 (2017)
- Delia Kesner, Daniel Ventura: Quantitative Types for the Linear Substitution Calculus. IFIP TCS 2014: 296-310
- Delia Kesner, Daniel Ventura: A resource aware semantics for a focused intuitionistic calculus. Math. Struct. Comput. Sci. 29(1): 93-126 (2019)
- Delia Kesner, Loïc Peyrot, Daniel Ventura: Node Replication: Theory And Practice. Log. Methods Comput. Sci. 20(1) (2024)

Tight Types and Exact Measures

Minimal typings = all and only information

Tightness was introduced by Accattoli, Graham-Lengrand and Kesner, to effectively capture minimal typings

This technique has been used in the λ-calculus to extract exact measures for several strategies

- call-by-value, call-by-need, linear-head, etc...

Tight types are used to type persistent terms:

$$(\lambda x.x)(\lambda x.x)$$

We say that $(\lambda x.x)$ is consuming and $(\lambda x.x)$ is persistent.

Tight Types and Exact Measures

Minimal typings = all and only information

Tightness was introduced by Accattoli, Graham-Lengrand and Kesner, to effectively capture minimal typings

This technique has been used in the λ-calculus to extract exact measures for several strategies

- call-by-value, call-by-need, linear-head, etc...

Tight types are used to type persistent terms:

$$(\lambda x.x)(\lambda x.x)$$

We say that $(\lambda x.x)$ is consuming and $(\lambda x.x)$ is persistent.

Tight Types and Exact Measures

Minimal typings = all and only information

Tightness was introduced by Accattoli, Graham-Lengrand and Kesner, to effectively capture minimal typings

This technique has been used in the λ-calculus to extract exact measures for several strategies

- call-by-value, call-by-need, linear-head, etc...

Tight types are used to type persistent terms:

$$(\lambda x.x)(\lambda x.x)$$

We say that $(\lambda x.x)$ is consuming and $(\lambda x.x)$ is persistent.

Quantitative (Tight) Types

The **sets of types** ($\mathcal{T}$) and **multi-types** are given by the following grammars:

$$
\begin{array}{llcl}
\textbf{(tight-types)} & \mathtt{t} & ::= & \bullet_{\mathcal{M}} \mid \bullet_{\mathcal{N}} \\
\textbf{(types)} & \sigma, \tau & ::= & \mathtt{t} \mid \mathcal{A} \to \sigma \\
\textbf{(multi-types)} & \mathcal{A} & ::= & [\sigma_k]_{k \in K}
\end{array}
$$

- Use different typing rules for persistent and consuming terms
- A derivation $\Gamma \vdash M : \tau$ is tight if both Γ and τ are tight

Tight constants $\bullet_{\mathcal{M}}$ and $\bullet_{\mathcal{N}}$ are related to normal/neutral forms:

$$\mathcal{M} ::= \mathcal{N} \mid \lambda x.\mathcal{M} \qquad \mathcal{N} ::= x \mid \mathcal{N}\,\mathcal{M}$$

Quantitative (Tight) Types

The **sets of types** ($\mathcal{T}$) and **multi-types** are given by the following grammars:

$$\begin{array}{lrcl} (\textbf{tight-types}) & \mathtt{t} & ::= & \bullet_{\mathcal{M}} \mid \bullet_{\mathcal{N}} \\ (\textbf{types}) & \sigma, \tau & ::= & \mathtt{t} \mid \mathcal{A} \to \sigma \\ (\textbf{multi-types}) & \mathcal{A} & ::= & [\sigma_k]_{k \in K} \end{array}$$

- Use different typing rules for persistent and consuming terms
- A derivation $\Gamma \vdash M : \tau$ is tight if both Γ and τ are tight

Tight constants $\bullet_{\mathcal{M}}$ and $\bullet_{\mathcal{N}}$ are related to normal/neutral forms:

$$\mathcal{M} \ ::= \ \mathcal{N} \mid \lambda x.\mathcal{M} \qquad\qquad \mathcal{N} \ ::= \ x \mid \mathcal{N}\,\mathcal{M}$$

Quantitative (Tight) Types

The **sets of types** ($\mathcal{T}$) and **multi-types** are given by the following grammars:

$$\begin{array}{lrcl} (\textbf{tight-types}) & \mathtt{t} & ::= & \bullet_{\mathcal{M}} \mid \bullet_{\mathcal{N}} \\ (\textbf{types}) & \sigma, \tau & ::= & \mathtt{t} \mid \mathcal{A} \rightarrow \sigma \\ (\textbf{multi-types}) & \mathcal{A} & ::= & [\sigma_k]_{k \in K} \end{array}$$

- Use different typing rules for persistent and consuming terms
- A derivation $\Gamma \vdash M : \tau$ is tight if both Γ and τ are tight

Tight constants $\bullet_{\mathcal{M}}$ and $\bullet_{\mathcal{N}}$ are related to normal/neutral forms:

$$\mathcal{M} \;::=\; \mathcal{N} \mid \lambda x.\mathcal{M} \qquad\qquad \mathcal{N} \;::=\; x \mid \mathcal{N}\,\mathcal{M}$$

Quantitative (Tight) Types

$$\frac{}{x : [\tau] \vdash x : \tau}$$

$$\frac{\Gamma \vdash t : \tau}{\Gamma \setminus\!\!\setminus x \vdash \lambda x.t : \Gamma(x) \to \tau} \qquad \frac{\Gamma \vdash t : \mathtt{t} \qquad \mathtt{tight}(\Gamma(x))}{\Gamma \setminus\!\!\setminus x \vdash \lambda x.t : \bullet_{\mathcal{M}}}$$

$$\frac{\Gamma \vdash t : \mathcal{A} \to \tau \qquad \Delta \vdash u : \mathcal{A}}{\Gamma + \Delta \vdash t\, u : \tau} \qquad \frac{\Gamma \vdash t : \bullet_{\mathcal{N}} \qquad \Delta \vdash u : \mathtt{t}}{\Gamma + \Delta \vdash t\, u : \bullet_{\mathcal{N}}}$$

$$\frac{(\Delta_k \vdash t : \sigma_k)_{k \in K} \qquad |K| > 0}{+_{k \in K} \Delta_k \vdash t : [\sigma_k]_{k \in K}} \qquad \frac{\Delta \vdash t : \sigma}{\Delta \vdash t : [\,]}$$

Example

Consider $t \equiv (\lambda x.x\mathtt{I}x)\Delta$, with $\mathtt{I} \equiv \lambda z.z$ and $\Delta \equiv \lambda y.yy$. Let $\mathcal{B} = [\underbrace{[[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}] \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_1}, \underbrace{[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_2}]$ and $\mathcal{A} = [\bullet_{\mathcal{M}}, \underbrace{\mathcal{B} \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_3}]$. Let Φ be:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : [\bullet_{\mathcal{N}}] \vdash x : \bullet_{\mathcal{N}}} \quad \dfrac{}{x : [\bullet_{\mathcal{N}}] \vdash x : \bullet_{\mathcal{N}}}}{x : [\bullet_{\mathcal{N}}, \bullet_{\mathcal{N}}] \vdash xx : \bullet_{\mathcal{N}}}}{\vdash \Delta : \bullet_{\mathcal{M}}} \qquad \dfrac{\dfrac{\dfrac{}{x : [\tau_1] \vdash x : \tau_1} \quad \dfrac{\dfrac{}{x : [\tau_2] \vdash x : \tau_2}}{x : [\tau_2] \vdash x : [\tau_2]}}{x : \mathcal{B} \vdash xx : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}}{\vdash \Delta : \tau_3}}{\vdash \Delta : \mathcal{A}}$$

Example (cont.)

Let then $\Phi_{\mathtt{I}}$ be:

$$\dfrac{\dfrac{x : [[\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}] \vdash x : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}}{\vdash \mathtt{I} : \tau_1} \qquad \dfrac{y : [\bullet_{\mathcal{M}}] \vdash y : \bullet_{\mathcal{M}}}{\vdash \mathtt{I} : \tau_2}}{\vdash \mathtt{I} : \mathcal{B}}$$

We have the following tight derivation for t:

$$\dfrac{\dfrac{\dfrac{\dfrac{x : [\tau_3] \vdash x : \tau_3 \quad \Phi_{\mathtt{I}}}{x : [\tau_3] \vdash x\mathtt{I} : [\bullet_{\mathcal{M}}]\to\bullet_{\mathcal{M}}} \qquad \dfrac{x : [\bullet_{\mathcal{M}}] \vdash x : \bullet_{\mathcal{M}}}{x : [\bullet_{\mathcal{M}}] \vdash x : [\bullet_{\mathcal{M}}]}}{x : \mathcal{A} \vdash x\mathtt{I}x : \bullet_{\mathcal{M}}}}{\vdash (\lambda x.x\mathtt{I}x) : \mathcal{A}\to\bullet_{\mathcal{M}}} \qquad \Phi}{\vdash (\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}}$$

$$\vdash^{(4,2)} (\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}$$

Example (cont.)

Let then $\Phi_{\mathtt{I}}$ be:

$$\dfrac{\dfrac{x : [[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}] \vdash x : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}{\vdash \mathtt{I} : \tau_1} \quad \dfrac{y : [\bullet_{\mathcal{M}}] \vdash y : \bullet_{\mathcal{M}}}{\vdash \mathtt{I} : \tau_2}}{\vdash \mathtt{I} : \mathcal{B}}$$

We have the following tight derivation for t:

$$\dfrac{\dfrac{\dfrac{\dfrac{x : [\tau_3] \vdash x : \tau_3 \quad \Phi_{\mathtt{I}}}{x : [\tau_3] \vdash x\mathtt{I} : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}} \quad \dfrac{x : [\bullet_{\mathcal{M}}] \vdash x : \bullet_{\mathcal{M}}}{x : [\bullet_{\mathcal{M}}] \vdash x : [\bullet_{\mathcal{M}}]}}{x : \mathcal{A} \vdash x\mathtt{I}x : \bullet_{\mathcal{M}}}}{\vdash (\lambda x.x\mathtt{I}x) : \mathcal{A} \to \bullet_{\mathcal{M}}} \quad \Phi}{\vdash (\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}}$$

$$\vdash^{(4,2)} (\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}$$

Example (cont.)

Let then $\Phi_{\mathtt{I}}$ be:

$$\dfrac{\dfrac{x : [[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}] \vdash x : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}{\vdash \mathtt{I} : \tau_1} \qquad \dfrac{y : [\bullet_{\mathcal{M}}] \vdash y : \bullet_{\mathcal{M}}}{\vdash \mathtt{I} : \tau_2}}{\vdash \mathtt{I} : \mathcal{B}}$$

We have the following tight derivation for t:

$$\dfrac{\dfrac{\dfrac{\dfrac{x : [\tau_3] \vdash x : \tau_3 \quad \Phi_{\mathtt{I}}}{x : [\tau_3] \vdash x\mathtt{I} : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}} \qquad \dfrac{x : [\bullet_{\mathcal{M}}] \vdash x : \bullet_{\mathcal{M}}}{x : [\bullet_{\mathcal{M}}] \vdash x : [\bullet_{\mathcal{M}}]}}{x : \mathcal{A} \vdash x\mathtt{I}x : \bullet_{\mathcal{M}}}}{\vdash (\lambda x.x\mathtt{I}x) : \mathcal{A} \to \bullet_{\mathcal{M}}} \qquad \Phi}{\vdash (\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}}$$

$$\vdash^{(4,2)} (\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}$$

- Beniamino Accattoli, Stéphane Graham-Lengrand, Delia Kesner: Tight typings and split bounds, fully developed. J. Funct. Program. 30: e14 (2020)
- Delia Kesner, Pierre Vial: Consuming and Persistent Types for Classical Logic. LICS 2020: 619-632
- Antonio Bucciarelli, Delia Kesner, Alejandro Ríos, Andrés Viso: The bang calculus revisited. Inf. Comput. 293: 105047 (2023)
- Sandra Alves, Delia Kesner, Daniel Ventura: A Quantitative Understanding of Pattern Matching. TYPES 2019: 3:1-3:36

Linearisation

Kfoury's Linearisation [2000]

"Can the standard λ-calculus be simulated by a calculus with a linearity condition on function evaluation?"

Kfoury defined a new "linear" calculus $\Lambda^\wedge$:

"If the formal parameter x of an abstraction $(\lambda x.t)$, is not dummy, then the free occurrences of x in the body t of the abstraction are in a one-one correspondence with the arguments to which the function is applied.

$$t, u \in \Lambda^\wedge ::= x \mid \lambda x.t \mid t.u_1 \wedge \cdots \wedge u_n$$

$\beta^\wedge$-reduction:

$$((\lambda x.t).u_1 \wedge \cdots \wedge u_n) \to t[u_1/x^{(1)}, \ldots, u_n/x^{(n)}]$$

Kfoury's Linearisation [2000]

"Can the standard λ-calculus be simulated by a calculus with a linearity condition on function evaluation?"

Kfoury defined a new "linear" calculus $\Lambda^{\wedge}$:

"If the formal parameter x of an abstraction $(\lambda x.t)$, is not dummy, then the free occurrences of x in the body t of the abstraction are in a one-one correspondence with the arguments to which the function is applied.

$$t, u \in \Lambda^{\wedge} ::= x \mid \lambda x.t \mid t.u_1 \wedge \cdots \wedge u_n$$

$\beta^{\wedge}$-reduction:

$$((\lambda x.t).u_1 \wedge \cdots \wedge u_n) \to t[u_1/x^{(1)}, \ldots, u_n/x^{(n)}]$$

Kfoury's Linearisation [2000]

"Can the standard λ-calculus be simulated by a calculus with a linearity condition on function evaluation?"

Kfoury defined a new "linear" calculus $\Lambda^\wedge$:

"If the formal parameter x of an abstraction $(\lambda x.t)$, is not dummy, then the free occurrences of x in the body t of the abstraction are in a one-one correspondence with the arguments to which the function is applied.

$$t, u \in \Lambda^\wedge ::= x \mid \lambda x.t \mid t.u_1 \wedge \cdots \wedge u_n$$

$\beta^\wedge$-reduction:

$$((\lambda x.t).u_1 \wedge \cdots \wedge u_n) \to t[u_1/x^{(1)}, \ldots, u_n/x^{(n)}]$$

Kfoury's Linearisation [2000]

"Can the standard λ-calculus be simulated by a calculus with a linearity condition on function evaluation?"

Kfoury defined a new "linear" calculus $\Lambda^\wedge$:

"If the formal parameter x of an abstraction $(\lambda x.t)$, is not dummy, then the free occurrences of x in the body t of the abstraction are in a one-one correspondence with the arguments to which the function is applied.

$$t, u \in \Lambda^\wedge ::= x \mid \lambda x.t \mid t.u_1 \wedge \cdots \wedge u_n$$

$\beta^\wedge$-reduction:

$$((\lambda x.t).u_1 \wedge \cdots \wedge u_n) \to t[u_1/x^{(1)}, \ldots, u_n/x^{(n)}]$$

Properties and Conjecture

"Well-formed terms of the new calculus are those for which there is a contracted term in the λ-calculus."

$$
\begin{array}{lcl}
|x| & = & x \\
|\lambda x.t| & = & \lambda x.|t| \\
 & & \textit{provided that } |t| \textit{ is defined} \\
|(t.u_1 \wedge \cdots \wedge u_n)| & = & |t||u_1| \\
 & & \textit{provided that } |t|, |u_1|, \ldots, |u_n| \textit{ are defined} \\
 & & \textit{and } |u_1| \equiv \cdots \equiv |u_n|
\end{array}
$$

Kfoury's conjecture:

"Let t be a standard λ-term. t is β-SN iff there is a well-formed expanded λ-term u such that $t \equiv |u|$ and every β-reduction from t can be lifted to a $\beta^{\wedge}$-reduction from u".

Properties and Conjecture

"Well-formed terms of the new calculus are those for which there is a contracted term in the λ-calculus."

$$
\begin{array}{lcl}
|x| & = & x \\
|\lambda x.t| & = & \lambda x.|t| \\
 & & \textit{provided that } |t| \textit{ is defined} \\
|(t.u_1 \wedge \cdots \wedge u_n)| & = & |t||u_1| \\
 & & \textit{provided that } |t|, |u_1|, \ldots, |u_n| \textit{ are defined} \\
 & & \textit{and } |u_1| \equiv \cdots \equiv |u_n|
\end{array}
$$

Kfoury's conjecture:

"Let t be a standard λ-term. t is β-SN iff there is a well-formed expanded λ-term u such that $t \equiv |u|$ and every β-reduction from t can be lifted to a $\beta^{\wedge}$-reduction from u".

Linearisation by Expansion - Damas and Florido (2004)

Expansion of terms typable with intersection types

- $\mathcal{E}_I(x : \tau) \lhd (y, \{x : \{y : \tau\}\}), \quad \text{if } x \neq y$
- $\mathcal{E}_I(\lambda x.t : \tau_1 \cap \cdots \cap \tau_n \to \sigma) \lhd (\lambda x_1 \ldots x_n.t^*, A)$
  - if $\mathcal{E}_I(t : \sigma) \lhd (t^*, A \cup \{x : \{x_1 : \tau_1, \ldots, x_n : \tau_n\}\})$
- $\mathcal{E}_I(tu : \sigma) \lhd (t_0 u_1 \ldots u_k, A_0 \uplus A_1 \uplus \cdots \uplus A_k)$
  - if for some $k > 0$ and $\tau_1, \ldots \tau_k$,
  - $\mathcal{E}_I(t : \tau_1 \cap \cdots \cap \tau_k \to \sigma) \lhd (t_0, A_0)$
  - and $\mathcal{E}_I(u : \tau_i) \lhd (u_i, A_i), (1 \leq i \leq k)$

Linearisation by Expansion - Damas and Florido (2004)

Expansion of terms typable with intersection types

- $\mathcal{E}_I(x : \tau) \lhd (y, \{x : \{y : \tau\}\}), \quad \text{if } x \neq y$
- $\mathcal{E}_I(\lambda x.t : \tau_1 \cap \cdots \cap \tau_n \to \sigma) \lhd (\lambda x_1 \ldots x_n.t^*, A)$
  - if $\mathcal{E}_I(t : \sigma) \lhd (t^*, A \cup \{x : \{x_1 : \tau_1, \ldots, x_n : \tau_n\}\})$
- $\mathcal{E}_I(tu : \sigma) \lhd (t_0 u_1 \ldots u_k, A_0 \uplus A_1 \uplus \cdots \uplus A_k)$
  - if for some $k > 0$ and $\tau_1, \ldots \tau_k$,
  - $\mathcal{E}_I(t : \tau_1 \cap \cdots \cap \tau_k \to \sigma) \lhd (t_0, A_0)$
  - and $\mathcal{E}_I(u : \tau_i) \lhd (u_i, A_i), (1 \leq i \leq k)$

Linearisation by Expansion - Damas and Florido (2004)

Expansion of terms typable with intersection types

- $\mathcal{E}_I(x : \tau) \lhd (y, \{x : \{y : \tau\}\}), \quad \text{if } x \neq y$
- $\mathcal{E}_I(\lambda x.t : \tau_1 \cap \cdots \cap \tau_n \to \sigma) \lhd (\lambda x_1 \ldots x_n.t^*, A)$
  - if $\mathcal{E}_I(t : \sigma) \lhd (t^*, A \cup \{x : \{x_1 : \tau_1, \ldots, x_n : \tau_n\}\})$
- $\mathcal{E}_I(tu : \sigma) \lhd (t_0 u_1 \ldots u_k, A_0 \uplus A_1 \uplus \cdots \uplus A_k)$
  - if for some $k > 0$ and $\tau_1, \ldots \tau_k$,
  - $\mathcal{E}_I(t : \tau_1 \cap \cdots \cap \tau_k \to \sigma) \lhd (t_0, A_0)$
  - and $\mathcal{E}_I(u : \tau_i) \lhd (u_i, A_i), (1 \leq i \leq k)$

Expansion and Algebraic Properties of Intersection

Considering different properties of the intersection relation:

$\cap$	**Source**	**Target**	**Preserves reductions**
ACI	λ	Simple Types	Weak Head Reduction
ACI	λI	Relevant Types	β-reduction
AC	λ	Affine Types	Weak Head Reduction
AC	λI	Linear Types	β-reduction
A	λI	Ordered Types	β-reduction

Sandra Alves, Mário Florido: *Structural Rules and Algebraic Properties of Intersection Types.* ICTAC 2022.

Weak (Linearisation)

The Weak Linear Lambda Calculus

A term t is weak linear if in any reduction sequence of t, when there is a contraction of a β-redex $(\lambda x.u)v$, then x occurs free in u at most once.

Example:

$(\lambda x.xx)(\lambda x.x) \longrightarrow_\beta$	$(\lambda x_1 x_2.x_1 x_2)(\lambda x.x)(\lambda x.x) \longrightarrow_\beta^*$
$(\lambda x.x)(\lambda x.x) \longrightarrow_\beta$	$(\lambda x.x)(\lambda x.x) \longrightarrow_\beta$
$(\lambda x.x)$	$(\lambda x.x)$

That is:

$(\lambda x_1 x_2.x_1 x_2)(\lambda x.x)(\lambda x.x)$ is weak linear, and $(\lambda x.xx)(\lambda x.x)$ is not

The Weak Linear Lambda Calculus

A term t is weak linear if in any reduction sequence of t, when there is a contraction of a β-redex $(\lambda x.u)v$, then x occurs free in u at most once.

Example:

$(\lambda x.xx)(\lambda x.x) \longrightarrow_\beta$	$(\lambda x_1 x_2.x_1 x_2)(\lambda x.x)(\lambda x.x) \longrightarrow^*_\beta$
$(\lambda x.x)(\lambda x.x) \longrightarrow_\beta$	$(\lambda x.x)(\lambda x.x) \longrightarrow_\beta$
$(\lambda x.x)$	$(\lambda x.x)$

That is:

$(\lambda x_1 x_2.x_1 x_2)(\lambda x.x)(\lambda x.x)$ is weak linear, and $(\lambda x.xx)(\lambda x.x)$ is not

The Weak Linear Lambda Calculus

A term t is weak linear if in any reduction sequence of t, when there is a contraction of a β-redex $(\lambda x.u)v$, then x occurs free in u at most once.

Example:

$(\lambda x.xx)(\lambda x.x) \longrightarrow_\beta$	$(\lambda x_1 x_2.x_1 x_2)(\lambda x.x)(\lambda x.x) \longrightarrow^*_\beta$
$(\lambda x.x)(\lambda x.x) \longrightarrow_\beta$	$(\lambda x.x)(\lambda x.x) \longrightarrow_\beta$
$(\lambda x.x)$	$(\lambda x.x)$

That is:

$(\lambda x_1 x_2.x_1 x_2)(\lambda x.x)(\lambda x.x)$ is weak linear, and $(\lambda x.xx)(\lambda x.x)$ is not

Weak linear terms have nice properties

Strong normalization:

- non-duplicating reduction
- weak linear reduction cannot have more steps than the size of the term

It is decidable to know if a λ-term is weak linear

Type inference for weak linear terms is both decidable and polynomial

Hence, the good properties of linear terms...

Weak linear terms have nice properties

Strong normalization:

- non-duplicating reduction
- weak linear reduction cannot have more steps than the size of the term

It is decidable to know if a λ-term is weak linear

Type inference for weak linear terms is both decidable and polynomial

Hence, the good properties of linear terms...

Weak linear terms have nice properties

Strong normalization:

- non-duplicating reduction
- weak linear reduction cannot have more steps than the size of the term

It is decidable to know if a λ-term is weak linear

Type inference for weak linear terms is both decidable and polynomial

Hence, the good properties of linear terms...

What happens with linearisation?

One β-redex:

$$(\lambda \mathbf{x}.xx)(\lambda \mathbf{y}.\mathbf{y})$$

$$\downarrow$$

$$(\lambda \mathbf{x_1 x_2}.x_1 x_2)(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y})$$

One redex created by the reduction (virtual):

$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z}.zz) \quad \rightarrow \quad (\lambda \mathbf{z}.zz)(\lambda \mathbf{y}.\mathbf{y})$$

$$\downarrow$$

$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z_1 z_2}.z_1 z_2)$$

Virtual redexes are characterised as (legal) paths in the initial term:

"For any legal path φ in a term t ending in an abstraction, there exists a degree I of a redex originated along some reduction of t such that $path(I) = \varphi$." - *Asperti and Laneve [1995]*

What happens with linearisation?

One β-redex:

$$(\lambda \mathbf{x}.xx)(\lambda \mathbf{y}.\mathbf{y})$$
$$\downarrow$$
$$(\lambda \mathbf{x_1 x_2}.x_1 x_2)(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y})$$

One redex created by the reduction (virtual):

$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z}.zz) \quad \rightarrow \quad (\lambda \mathbf{z}.zz)(\lambda \mathbf{y}.\mathbf{y})$$
$$\downarrow$$
$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z_1 z_2}.z_1 z_2)$$

Virtual redexes are characterised as (legal) paths in the initial term:

"For any legal path φ in a term t ending in an abstraction, there exists a degree I of a redex originated along some reduction of t such that $path(I) = \varphi$." - *Asperti and Laneve [1995]*

What happens with linearisation?

One β-redex:

$$(\lambda \mathbf{x}.xx)(\lambda \mathbf{y}.\mathbf{y})$$
$$\downarrow$$
$$(\lambda \mathbf{x_1 x_2}.x_1 x_2)(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y})$$

One redex created by the reduction (virtual):

$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z}.zz) \quad \rightarrow \quad (\lambda \mathbf{z}.zz)(\lambda \mathbf{y}.\mathbf{y})$$
$$\downarrow$$
$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z_1 z_2}.z_1 z_2)$$

Virtual redexes are characterised as (legal) paths in the initial term:

"For any legal path φ in a term t ending in an abstraction, there exists a degree I of a redex originated along some reduction of t such that $path(I) = \varphi$." - *Asperti and Laneve [1995]*

What happens with linearisation?

One β-redex:

$$(\lambda \mathbf{x}.xx)(\lambda \mathbf{y}.\mathbf{y})$$
$$\downarrow$$
$$(\lambda \mathbf{x_1 x_2}.x_1 x_2)(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y})$$

One redex created by the reduction (virtual):

$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z}.zz) \quad \rightarrow \quad (\lambda \mathbf{z}.zz)(\lambda \mathbf{y}.\mathbf{y})$$
$$\downarrow$$
$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z_1 z_2}.z_1 z_2)$$

Virtual redexes are characterised as (legal) paths in the initial term:

"For any legal path φ in a term t ending in an abstraction, there exists a degree I of a redex originated along some reduction of t such that $path(I) = \varphi$." - *Asperti and Laneve [1995]*

What happens with linearisation?

One β-redex:

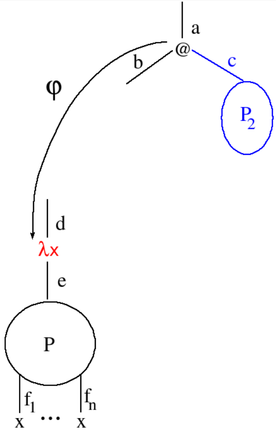
$$(\lambda \mathbf{x}.xx)(\lambda \mathbf{y}.\mathbf{y})$$
$$\downarrow$$
$$(\lambda \mathbf{x_1 x_2}.x_1 x_2)(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y})$$

One redex created by the reduction (virtual):

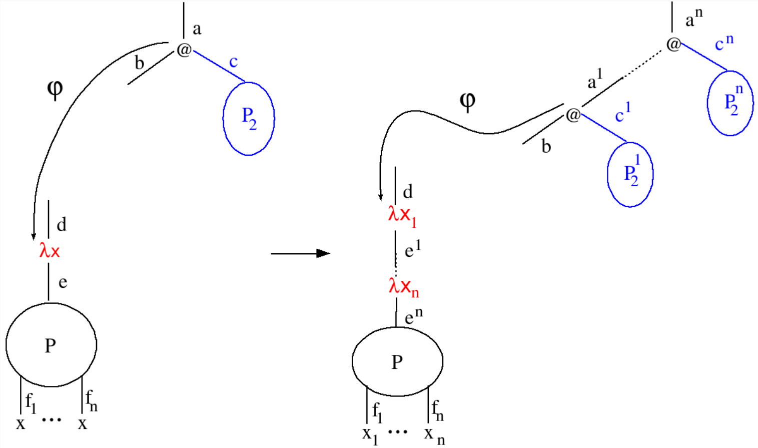
$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z}.zz) \quad \rightarrow \quad (\lambda \mathbf{z}.zz)(\lambda \mathbf{y}.\mathbf{y})$$
$$\downarrow$$
$$(\lambda x.x(\lambda \mathbf{y}.\mathbf{y})(\lambda \mathbf{y}.\mathbf{y}))(\lambda \mathbf{z_1 z_2}.z_1 z_2)$$

Virtual redexes are characterised as (legal) paths in the initial term:

"For any legal path φ in a term t ending in an abstraction, there exists a degree I of a redex originated along some reduction of t such that $path(I) = \varphi$." - *Asperti and Laneve [1995]*


a
@
b
c
φ
P2
d
λx
e
P
f1
fn
x
...
x

$\mathcal{L}(t)$



a
b
c
φ
P2
d
λx
e
P
f1
fn
x ... x
an
cn
a1
c1
@
φ
b
P2n
P21
d
λx1
e1
λxn
en
P
f1
fn
x1 ... xn

Transformation

$$\mathcal{T}(t) = \begin{cases} t & \text{if all\_linear}(\mathcal{LP}) \\ \mathcal{T}(\mathcal{L}(t)) & \text{otherwise} \end{cases}$$

all_linear($\mathcal{LP}$) returns *true* if all the legal paths in $\mathcal{LP}$ end in a linear abstraction, and *false* otherwise.

Let $D = \lambda y_1 y_2 . y_1 y_2$, then:

$\mathcal{T}(\lambda x.x(\lambda y.yy)v)(\lambda fz.f(fz))$

$= (\lambda x.xDDDvvvv)(\lambda f_1 f_2 f_3 z_1 z_2 z_3 z_4 . f_1(f_2 z_1 z_2)(f_3 z_3 z_4))$

and

$(\lambda x.x(\lambda y.yy)v)(\lambda fz.f(fz)) \rightarrow_{\beta*} (vv)(vv)$

$(\lambda x.xDDDvvvv)(\lambda f_1 f_2 f_3 z_1 z_2 z_3 z_4 . f_1(f_2 z_1 z_2)(f_3 z_3 z_4)) \rightarrow_{\beta*} (vv)(vv)$

Transformation

$$\mathcal{T}(t) = \begin{cases} t & \text{if all\_linear}(\mathcal{LP}) \\ \mathcal{T}(\mathcal{L}(t)) & \text{otherwise} \end{cases}$$

all_linear($\mathcal{LP}$) returns *true* if all the legal paths in $\mathcal{LP}$ end in a linear abstraction, and *false* otherwise.

Let $D = \lambda y_1 y_2 . y_1 y_2$, then:

$\mathcal{T}(\lambda x.x(\lambda y.yy)v)(\lambda fz.f(fz))$

$= (\lambda x.xDDDvvvv)(\lambda f_1 f_2 f_3 z_1 z_2 z_3 z_4 . f_1(f_2 z_1 z_2)(f_3 z_3 z_4))$

and

$(\lambda x.x(\lambda y.yy)v)(\lambda fz.f(fz)) \to_{\beta *} (vv)(vv)$

$(\lambda x.xDDDvvvv)(\lambda f_1 f_2 f_3 z_1 z_2 z_3 z_4 . f_1(f_2 z_1 z_2)(f_3 z_3 z_4)) \to_{\beta *} (vv)(vv)$

Transformation

$$\mathcal{T}(t) = \begin{cases} t & \text{if all\_linear}(\mathcal{LP}) \\ \mathcal{T}(\mathcal{L}(t)) & \text{otherwise} \end{cases}$$

all_linear($\mathcal{LP}$) returns *true* if all the legal paths in $\mathcal{LP}$ end in a linear abstraction, and *false* otherwise.

Let $D = \lambda y_1 y_2.y_1 y_2$, then:

$\mathcal{T}(\lambda x.x(\lambda y.yy)v)(\lambda fz.f(fz))$

$= (\lambda x.xDDDvvvv)(\lambda f_1 f_2 f_3 z_1 z_2 z_3 z_4.f_1(f_2 z_1 z_2)(f_3 z_3 z_4))$

and

$(\lambda x.x(\lambda y.yy)v)(\lambda fz.f(fz)) \rightarrow_{\beta *} (vv)(vv)$

$(\lambda x.xDDDvvvv)(\lambda f_1 f_2 f_3 z_1 z_2 z_3 z_4.f_1(f_2 z_1 z_2)(f_3 z_3 z_4)) \rightarrow_{\beta *} (vv)(vv)$

Transformation

$$\mathcal{T}(t) = \begin{cases} t & \text{if all\_linear}(\mathcal{LP}) \\ \mathcal{T}(\mathcal{L}(t)) & \text{otherwise} \end{cases}$$

all_linear($\mathcal{LP}$) returns *true* if all the legal paths in $\mathcal{LP}$ end in a linear abstraction, and *false* otherwise.

Let $D = \lambda y_1 y_2 . y_1 y_2$, then:

$\mathcal{T}(\lambda x.x(\lambda y.yy)v)(\lambda fz.f(fz))$

$\quad = (\lambda x.xDDDvvvv)(\lambda f_1 f_2 f_3 z_1 z_2 z_3 z_4 . f_1(f_2 z_1 z_2)(f_3 z_3 z_4))$

and

$(\lambda x.x(\lambda y.yy)v)(\lambda fz.f(fz)) \rightarrow_{\beta *} (vv)(vv)$

$(\lambda x.xDDDvvvv)(\lambda f_1 f_2 f_3 z_1 z_2 z_3 z_4 . f_1(f_2 z_1 z_2)(f_3 z_3 z_4)) \rightarrow_{\beta *} (vv)(vv)$

Properties of $\mathcal{T}$ and yet another conjecture

- $\mathcal{T}$ preserves β-normal forms
- If $\mathcal{T}$ terminates then $\mathcal{T}(t)$ is weak linear

But when does $\mathcal{T}$ terminates?

Let $\Delta = \lambda x.xx$, $D = \lambda x_1 x_2.x_1 x_2$, and $\Omega = \Delta\Delta$. We have:

$$\begin{aligned}\mathcal{T}(\Omega) &= \mathcal{T}(D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2.x_1 x_2 x_2)D\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3)D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3 x_3)DD\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3 x_4.x_1 x_2 x_3 x_4)DD\Delta\Delta) = \cdots\end{aligned}$$

Our conjecture:

"$\mathcal{T}$ is a total function for strongly normalising terms"

- $\mathcal{T}$ preserves β-normal forms
- If $\mathcal{T}$ terminates then $\mathcal{T}(t)$ is weak linear

But when does $\mathcal{T}$ terminates?

Let $\Delta = \lambda x.xx$, $D = \lambda x_1 x_2.x_1 x_2$, and $\Omega = \Delta\Delta$. We have:

$$\begin{aligned}\mathcal{T}(\Omega) &= \mathcal{T}(D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2.x_1 x_2 x_2)D\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3)D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3 x_3)DD\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3 x_4.x_1 x_2 x_3 x_4)DD\Delta\Delta) = \cdots\end{aligned}$$

Our conjecture:

"$\mathcal{T}$ is a total function for strongly normalising terms"

Properties of $\mathcal{T}$ and yet another conjecture

- $\mathcal{T}$ preserves β-normal forms
- If $\mathcal{T}$ terminates then $\mathcal{T}(t)$ is weak linear

But when does $\mathcal{T}$ terminates?

Let $\Delta = \lambda x.xx$, $D = \lambda x_1 x_2.x_1 x_2$, and $\Omega = \Delta\Delta$. We have:

$$\begin{aligned}\mathcal{T}(\Omega) &= \mathcal{T}(D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2.x_1 x_2 x_2)D\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3)D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3 x_3)DD\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3 x_4.x_1 x_2 x_3 x_4)DD\Delta\Delta) = \cdots\end{aligned}$$

Our conjecture:

"$\mathcal{T}$ is a total function for strongly normalising terms"

Properties of $\mathcal{T}$ and yet another conjecture

- $\mathcal{T}$ preserves β-normal forms
- If $\mathcal{T}$ terminates then $\mathcal{T}(t)$ is weak linear

But when does $\mathcal{T}$ terminates?

Let $\Delta = \lambda x.xx$, $D = \lambda x_1 x_2 . x_1 x_2$, and $\Omega = \Delta\Delta$. We have:

$$\begin{aligned}\mathcal{T}(\Omega) &= \mathcal{T}(D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2 . x_1 x_2 x_2) D\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3 . x_1 x_2 x_3) D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2 x_3 . x_1 x_2 x_3 x_3) DD\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3 x_4 . x_1 x_2 x_3 x_4) DD\Delta\Delta) = \cdots\end{aligned}$$

Our conjecture:

"$\mathcal{T}$ is a total function for strongly normalising terms"

Properties of $\mathcal{T}$ and yet another conjecture

- $\mathcal{T}$ preserves β-normal forms
- If $\mathcal{T}$ terminates then $\mathcal{T}(t)$ is weak linear

But when does $\mathcal{T}$ terminates?

Let $\Delta = \lambda x.xx$, $D = \lambda x_1 x_2.x_1 x_2$, and $\Omega = \Delta\Delta$. We have:

$$\begin{aligned}\mathcal{T}(\Omega) &= \mathcal{T}(D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2.x_1 x_2 x_2)D\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3)D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3 x_3)DD\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3 x_4.x_1 x_2 x_3 x_4)DD\Delta\Delta) = \cdots\end{aligned}$$

Our conjecture:

"$\mathcal{T}$ is a total function for strongly normalising terms"

Properties of $\mathcal{T}$ and yet another conjecture

- $\mathcal{T}$ preserves β-normal forms
- If $\mathcal{T}$ terminates then $\mathcal{T}(t)$ is weak linear

But when does $\mathcal{T}$ terminates?

Let $\Delta = \lambda x.xx$, $D = \lambda x_1x_2.x_1x_2$, and $\Omega = \Delta\Delta$. We have:

$$\begin{aligned}\mathcal{T}(\Omega) &= \mathcal{T}(D\Delta\Delta) = \mathcal{T}(\lambda x_1x_2.x_1x_2x_2)D\Delta) \\ &= \mathcal{T}(\lambda x_1x_2x_3.x_1x_2x_3)D\Delta\Delta) = \mathcal{T}(\lambda x_1x_2x_3.x_1x_2x_3x_3)DD\Delta) \\ &= \mathcal{T}(\lambda x_1x_2x_3x_4.x_1x_2x_3x_4)DD\Delta\Delta) = \cdots\end{aligned}$$

Our conjecture:

"$\mathcal{T}$ is a total function for strongly normalising terms"

Properties of $\mathcal{T}$ and yet another conjecture

- $\mathcal{T}$ preserves β-normal forms
- If $\mathcal{T}$ terminates then $\mathcal{T}(t)$ is weak linear

But when does $\mathcal{T}$ terminates?

Let $\Delta = \lambda x.xx$, $D = \lambda x_1 x_2.x_1 x_2$, and $\Omega = \Delta\Delta$. We have:

$$\begin{aligned}\mathcal{T}(\Omega) &= \mathcal{T}(D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2.x_1 x_2 x_2)D\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3)D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3 x_3)DD\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3 x_4.x_1 x_2 x_3 x_4)DD\Delta\Delta) = \cdots\end{aligned}$$

Our conjecture:

"$\mathcal{T}$ is a total function for strongly normalising terms"

Properties of $\mathcal{T}$ and yet another conjecture

- $\mathcal{T}$ preserves β-normal forms
- If $\mathcal{T}$ terminates then $\mathcal{T}(t)$ is weak linear

But when does $\mathcal{T}$ terminates?

Let $\Delta = \lambda x.xx$, $D = \lambda x_1 x_2.x_1 x_2$, and $\Omega = \Delta\Delta$. We have:

$$\begin{aligned}\mathcal{T}(\Omega) &= \mathcal{T}(D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2.x_1 x_2 x_2)D\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3)D\Delta\Delta) = \mathcal{T}(\lambda x_1 x_2 x_3.x_1 x_2 x_3 x_3)DD\Delta) \\ &= \mathcal{T}(\lambda x_1 x_2 x_3 x_4.x_1 x_2 x_3 x_4)DD\Delta\Delta) = \cdots\end{aligned}$$

Our conjecture:

"$\mathcal{T}$ is a total function for strongly normalising terms"

Virtual redexes and Persistent versus Consuming

Virtual redexes involve abstractions that are consumed by reduction

Consider the term $t \equiv (\lambda x.xx)(\lambda x.x) \to (\lambda x.x)(\lambda x.x)$

The set of legal paths of t contains two paths of type $@ - \lambda$:

- one ends in $(\lambda x.xx)$, corresponding to the redex $(\lambda x.xx)(\lambda x.x)$
- one ends in $(\lambda x.x)$, corresponding to the redex $(\lambda x.x)(\lambda x.x)$

But only one copy of $(\lambda x.x)$ is going to be consumed by reduction, whereas the other will persist in the normal form.

Note that, after one step of $\mathcal{T}$ we obtain $(\lambda x_1 x_1.x_1 x_2)(\lambda x.x)(\lambda x.x)$

And only one copy of $(\lambda x.x)$ is now the end of a $@ - \lambda$ path.

Virtual redexes and Persistent versus Consuming

Virtual redexes involve abstractions that are consumed by reduction

Consider the term $t \equiv (\lambda x.xx)(\lambda x.x) \to (\lambda x.x)(\lambda x.x)$

The set of legal paths of t contains two paths of type $@ - \lambda$:

- one ends in $(\lambda x.xx)$, corresponding to the redex $(\lambda x.xx)(\lambda x.x)$
- one ends in $(\lambda x.x)$, corresponding to the redex $(\lambda x.x)(\lambda x.x)$

But only one copy of $(\lambda x.x)$ is going to be consumed by reduction, whereas the other will persist in the normal form.

Note that, after one step of $\mathcal{T}$ we obtain $(\lambda x_1 x_1.x_1 x_2)(\lambda x.x)(\lambda x.x)$

And only one copy of $(\lambda x.x)$ is now the end of a $@ - \lambda$ path.

Virtual redexes and Persistent versus Consuming

Virtual redexes involve abstractions that are consumed by reduction

Consider the term $t \equiv (\lambda x.xx)(\lambda x.x) \to (\lambda x.x)(\lambda x.x)$

The set of legal paths of t contains two paths of type $@ - \lambda$:

- one ends in $(\lambda x.xx)$, corresponding to the redex $(\lambda x.xx)(\lambda x.x)$
- one ends in $(\lambda x.x)$, corresponding to the redex $(\lambda x.x)(\lambda x.x)$

But only one copy of $(\lambda x.x)$ is going to be consumed by reduction, whereas the other will persist in the normal form.

Note that, after one step of $\mathcal{T}$ we obtain $(\lambda x_1 x_1.x_1 x_2)(\lambda x.x)(\lambda x.x)$

And only one copy of $(\lambda x.x)$ is now the end of a $@ - \lambda$ path.

Virtual redexes and Persistent versus Consuming

Virtual redexes involve abstractions that are consumed by reduction

Consider the term $t \equiv (\lambda x.xx)(\lambda x.x) \to (\lambda x.x)(\lambda x.x)$

The set of legal paths of t contains two paths of type $@ - \lambda$:

- one ends in $(\lambda x.xx)$, corresponding to the redex $(\lambda x.xx)(\lambda x.x)$
- one ends in $(\lambda x.x)$, corresponding to the redex $(\lambda x.x)(\lambda x.x)$

But only one copy of $(\lambda x.x)$ is going to be consumed by reduction, whereas the other will persist in the normal form.

Note that, after one step of $\mathcal{T}$ we obtain $(\lambda x_1 x_1.x_1 x_2)(\lambda x.x)(\lambda x.x)$

And only one copy of $(\lambda x.x)$ is now the end of a $@ - \lambda$ path.

Quantitative Weak Linearisation

Expansion of Consuming Terms

$$\mathsf{E}(x : \sigma) \quad \triangleleft \quad (y, \{x : [y : \sigma]\}), \; \textcolor{blue}{y \text{ fresh}}$$

$$\begin{aligned}
\mathsf{E}(\lambda x.t : [\tau_i]_{i=1\ldots n} \to \sigma) \quad &\triangleleft \quad (\lambda x_1 \ldots x_n.t^*, A), \\
&\quad \text{if for } n > 0 \text{ and fresh } x_1, \ldots, x_n \\
&\quad \mathsf{E}(t : \sigma) \triangleleft (t^*, A; \{x : [x_1 : \tau_1, \ldots, x_n : \tau_n]\})
\end{aligned}$$

$$\begin{aligned}
\mathsf{E}(tu : \sigma) \quad &\triangleleft \quad (t_0 u_1 \ldots u_m, +_{j=0\ldots m} A_j), \\
&\quad \text{if for some } m > 0 \text{ and } \tau_1, \ldots, \tau_m \\
&\quad \mathsf{E}(t : [\tau_j]_{j=1\ldots m} \to \sigma) \triangleleft (t_0, A_0) \\
&\quad \text{and } \left(\mathsf{E}(u : \tau_j) \triangleleft (u_j, A_j)\right)_{j=1\ldots m}
\end{aligned}$$

Expansion of Persistent Terms

$$\mathsf{E}(x : \mathtt{t}) \quad \triangleleft \quad (x, \{x : [x : \mathtt{t}]\})$$

$$\mathsf{E}(\lambda x.t : \bullet_{\mathcal{M}}) \quad \triangleleft \quad (\lambda x.t^*, A),$$

if for some tight type $\mathtt{t}$ and $n \geq 0$

$\mathsf{E}(t : \mathtt{t}) \triangleleft (t^*, A; \{x : [x : \mathtt{t}_1, \ldots, x : \mathtt{t}_n]\})$

$$\mathsf{E}(tu : \bullet_{\mathcal{N}}) \quad \triangleleft \quad (t^* u^*, A_1 + A_2),$$

if for some tight type $\mathtt{t}$

$\mathsf{E}(t : \bullet_{\mathcal{N}}) \triangleleft (t^*, A_1)$ and $\mathsf{E}(u : \mathtt{t}) \triangleleft (u^*, A_2)$

Expansion of Persistent Terms

$$\mathsf{E}(x : \mathtt{t}) \triangleleft (x, \{x : [x : \mathtt{t}]\})$$

$$\mathsf{E}(\lambda x.t : \bullet_{\mathcal{M}}) \triangleleft (\lambda x.t^*, A),$$

if for some tight type $\mathtt{t}$ and $n \geq 0$

$$\mathsf{E}(t : \mathtt{t}) \triangleleft (t^*, A; \{x : [x : \mathtt{t}_1, \ldots, x : \mathtt{t}_n]\})$$

$$\mathsf{E}(tu : \bullet_{\mathcal{N}}) \triangleleft (t^* u^*, A_1 + A_2),$$

if for some tight type $\mathtt{t}$

$$\mathsf{E}(t : \bullet_{\mathcal{N}}) \triangleleft (t^*, A_1) \text{ and } \mathsf{E}(u : \mathtt{t}) \triangleleft (u^*, A_2)$$

Expansion of Persistent Terms

$$\mathsf{E}(x : \mathtt{t}) \quad \triangleleft \quad (x, \{x : [x : \mathtt{t}]\})$$

$$\mathsf{E}(\lambda x.t : \bullet_{\mathcal{M}}) \quad \triangleleft \quad (\lambda x.t^*, A),$$

if for some tight type $\mathtt{t}$ and $n \geq 0$

$$\mathsf{E}(t : \mathtt{t}) \triangleleft (t^*, A; \{x : [x : \mathtt{t}_1, \ldots, x : \mathtt{t}_n]\})$$

$$\mathsf{E}(tu : \bullet_{\mathcal{N}}) \quad \triangleleft \quad (t^* u^*, A_1 + A_2),$$

if for some tight type $\mathtt{t}$

$\mathsf{E}(t : \bullet_{\mathcal{N}}) \triangleleft (t^*, A_1)$ and $\mathsf{E}(u : \mathtt{t}) \triangleleft (u^*, A_2)$

Example

Recall $t \equiv (\lambda x.x\mathtt{I}x)\Delta$, with $\mathtt{I} \equiv \lambda z.z$ and $\Delta \equiv \lambda y.yy$, and

$$\mathcal{B} = [\underbrace{[[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}] \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_1}, \underbrace{[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_2}] \text{ and}$$

$$\mathcal{A} = [\bullet_{\mathcal{M}}, \underbrace{\mathcal{B} \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_3}].$$

$$E(\lambda x.xx : \tau_3) \lhd (\lambda x_1 x_2.x_1 x_2, \varnothing)$$
$$E(xx : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}) \lhd (x_1 x_2, \{x : [x_1 : \tau_1, x_2 : \tau_2]\})$$
$$E(x : \tau_1) \lhd (x_1, \{x : [x_1 : \tau_1]\})$$
$$E(x : \tau_2) \lhd (x_2, \{x : [x_2 : \tau_2]\})$$

$$E(\lambda x.xx : \bullet_{\mathcal{M}}) \lhd (\lambda x.xx, \varnothing)$$
$$E(xx : \bullet_{\mathcal{N}}) \lhd (xx, \{x : [x : \bullet_{\mathcal{N}}, x : \bullet_{\mathcal{N}}]\})$$
$$E(x : \bullet_{\mathcal{N}}) \lhd (x, \{x : [x : \bullet_{\mathcal{N}}]\})$$
$$E(x : \bullet_{\mathcal{N}}) \lhd (x, \{x : [x : \bullet_{\mathcal{N}}]\})$$

Example

Recall $t \equiv (\lambda x.x\mathtt{I}x)\Delta$, with $\mathtt{I} \equiv \lambda z.z$ and $\Delta \equiv \lambda y.yy$, and

$\mathcal{B} = [\underbrace{[[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}] \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_1}, \underbrace{[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_2}]$ and

$\mathcal{A} = [\bullet_{\mathcal{M}}, \underbrace{\mathcal{B} \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_3}]$.

$$
\begin{array}{l}
\mathsf{E}(\lambda x.xx : \tau_3) \lhd (\lambda x_1 x_2.x_1 x_2, \varnothing) \\
\quad \mathsf{E}(xx : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}) \lhd (x_1 x_2, \{x : [x_1 : \tau_1, x_2 : \tau_2]\}) \\
\quad\quad \mathsf{E}(x : \tau_1) \lhd (x_1, \{x : [x_1 : \tau_1]\}) \\
\quad\quad \mathsf{E}(x : \tau_2) \lhd (x_2, \{x : [x_2 : \tau_2]\})
\end{array}
$$

$$
\begin{array}{l}
\mathsf{E}(\lambda x.xx : \bullet_{\mathcal{M}}) \lhd (\lambda x.xx, \varnothing) \\
\quad \mathsf{E}(xx : \bullet_{\mathcal{N}}) \lhd (xx, \{x : [x : \bullet_{\mathcal{N}}, x : \bullet_{\mathcal{N}}]\}) \\
\quad\quad \mathsf{E}(x : \bullet_{\mathcal{N}}) \lhd (x, \{x : [x : \bullet_{\mathcal{N}}]\}) \\
\quad\quad \mathsf{E}(x : \bullet_{\mathcal{N}}) \lhd (x, \{x : [x : \bullet_{\mathcal{N}}]\})
\end{array}
$$

Example

Recall $t \equiv (\lambda x.x\mathtt{I}x)\Delta$, with $\mathtt{I} \equiv \lambda z.z$ and $\Delta \equiv \lambda y.yy$, and

$\mathcal{B} = [\underbrace{[[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}] \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_1}, \underbrace{[\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_2}]$ and

$\mathcal{A} = [\bullet_{\mathcal{M}}, \underbrace{\mathcal{B} \to [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}}_{\tau_3}]$.

$$
\begin{array}{l}
\mathsf{E}(\lambda x.xx : \tau_3) \lhd (\lambda x_1 x_2.x_1 x_2, \varnothing) \\
\quad \mathsf{E}(xx : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}) \lhd (x_1 x_2, \{x : [x_1 : \tau_1, x_2 : \tau_2]\}) \\
\qquad \mathsf{E}(x : \tau_1) \lhd (x_1, \{x : [x_1 : \tau_1]\}) \\
\qquad \mathsf{E}(x : \tau_2) \lhd (x_2, \{x : [x_2 : \tau_2]\})
\end{array}
$$

$$
\begin{array}{l}
\mathsf{E}(\lambda x.xx : \bullet_{\mathcal{M}}) \lhd (\lambda x.xx, \varnothing) \\
\quad \mathsf{E}(xx : \bullet_{\mathcal{N}}) \lhd (xx, \{x : [x : \bullet_{\mathcal{N}}, x : \bullet_{\mathcal{N}}]\}) \\
\qquad \mathsf{E}(x : \bullet_{\mathcal{N}}) \lhd (x, \{x : [x : \bullet_{\mathcal{N}}]\}) \\
\qquad \mathsf{E}(x : \bullet_{\mathcal{N}}) \lhd (x, \{x : [x : \bullet_{\mathcal{N}}]\})
\end{array}
$$

Example

$$
\begin{array}{l}
\mathsf{E}((\lambda x.x\mathtt{I}x)\Delta : \bullet_{\mathcal{M}}) \lhd ((\lambda x_3 x_4.x_3\mathtt{II}x_4)(\lambda x_1 x_2.x_1 x_2)(\lambda x.xx), \emptyset) \\
\quad \mathsf{E}(\lambda x.x\mathtt{I}x : \mathcal{A} \to \bullet_{\mathcal{M}}) \lhd (\lambda x_3 x_4.x_3\mathtt{II}x_4, \emptyset) \\
\quad\quad \mathsf{E}(x\mathtt{I}x : \bullet_{\mathcal{M}}) \lhd (x_3\mathtt{II}x_4, \{x : [x_3 : \tau_3, x_4 : \bullet_{\mathcal{M}}]\}) \\
\quad\quad\quad \mathsf{E}(x\mathtt{I} : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}) \lhd (x_3\mathtt{II}, \{x : [x_3 : \tau_3]\}) \\
\quad\quad\quad\quad \mathsf{E}(x : \tau_3) \lhd (x_3, \{x : [x_3 : \tau_3]\}) \\
\quad\quad\quad\quad \mathsf{E}(\mathtt{I} : \tau_1) \lhd (\lambda x_5.x_5, \emptyset) \\
\quad\quad\quad\quad\quad \mathsf{E}(x : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}) \lhd (x_5, \{x : [x_5 : [\bullet_{\mathcal{M}}] \to \bullet_{\mathcal{M}}]\}) \\
\quad\quad\quad\quad \mathsf{E}(\mathtt{I} : \tau_2) \lhd (\lambda x_6.x_6, \emptyset) \\
\quad\quad\quad\quad\quad \mathsf{E}(x : \bullet_{\mathcal{M}}) \lhd (x_6, \{x : [x_6 : \bullet_{\mathcal{M}}]\}) \\
\quad\quad\quad \mathsf{E}(x : \bullet_{\mathcal{M}}) \lhd (x_4, \{x : [x_4 : \bullet_{\mathcal{M}}]\}) \\
\quad \mathsf{E}(\lambda x.xx : \tau_3) \lhd (\lambda x_1 x_2.x_1 x_2, \emptyset) \\
\quad \mathsf{E}(\lambda x.xx : \bullet_{\mathcal{M}}) \lhd (\lambda x.xx, \emptyset)
\end{array}
$$

Properties of E

Let $\mathsf{E}(t_1 : \mathtt{t}) \triangleleft (u_1, A_1)$ be a tight expansion and $t_1 \to_{\mathsf{nmx}} t_2$:

1. There is a term u_2 such that $\mathsf{E}(t_2 : \mathtt{t}) \triangleleft (u_2, A_2)$ is tight, $u_1 \to^*_{\mathsf{nmx}} u_2$ and $A_2 \subseteq A_1$.
2. If $\neg\mathtt{abs}(u_1)$ then for any $u' \neq u_2$ s.t. $u_1 \to^*_{\mathsf{nmx}} u_2 = u_1 \to^*_{\mathsf{nmx}} u' \to^*_{\mathsf{nmx}} u_2$, $\neg\mathtt{abs}(u')$.

where $\to_{\mathsf{nmx}}$ is a non-deterministic maximal strategy.

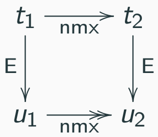
Thus E commutes with $\to_{\mathsf{nmx}}$.

Let $\mathsf{E}(t_1 : \mathtt{t}) \lhd (u_1, A_1)$ be a tight expansion and $t_1 \rightarrow_{\mathsf{nmx}} t_2$:

1. There is a term u_2 such that $\mathsf{E}(t_2 : \mathtt{t}) \lhd (u_2, A_2)$ is tight, $u_1 \rightarrow^*_{\mathsf{nmx}} u_2$ and $A_2 \subseteq A_1$.
2. If $\neg\mathtt{abs}(u_1)$ then for any $u' \neq u_2$ s.t. $u_1 \rightarrow^*_{\mathsf{nmx}} u_2 = u_1 \rightarrow^*_{\mathsf{nmx}} u' \rightarrow^*_{\mathsf{nmx}} u_2$, $\neg\mathtt{abs}(u')$.

where $\rightarrow_{\mathsf{nmx}}$ is a non-deterministic maximal strategy.

Thus E commutes with $\rightarrow_{\mathsf{nmx}}$.



A Typing Characterisation of Weak Terms: System $\mathcal{WL}$

$$\frac{}{x : [\tau] \vdash_{\mathsf{wl}} x : \tau} \qquad \frac{\Delta \vdash_{\mathsf{wl}} t : \sigma}{\Delta \vdash_{\mathsf{wl}} t : [\sigma]} \qquad \frac{\Delta \vdash_{\mathsf{wl}} t : \sigma}{\Delta \vdash_{\mathsf{wl}} t : [\,]}$$

$$\frac{\Gamma \vdash_{\mathsf{wl}} t : \tau \qquad |\Gamma(x)| \leq 1}{\Gamma \setminus\!\!\setminus x \vdash_{\mathsf{wl}} \lambda x.t : \Gamma(x) \to \tau} \qquad \frac{\Gamma \vdash_{\mathsf{wl}} t : \mathtt{t} \qquad \mathtt{tight}(\Gamma(x))}{\Gamma \setminus\!\!\setminus x \vdash_{\mathsf{wl}} \lambda x.t : \bullet_{\mathcal{M}}}$$

$$\frac{\Gamma \vdash_{\mathsf{wl}} t : \mathcal{A} \to \tau \qquad \Delta \vdash_{\mathsf{wl}} u : \mathcal{A}}{\Gamma + \Delta \vdash_{\mathsf{wl}} t\,u : \tau} \qquad \frac{\Gamma \vdash_{\mathsf{wl}} t : \bullet_{\mathcal{N}} \qquad \Delta \vdash_{\mathsf{wl}} u : \mathtt{t}}{\Gamma + \Delta \vdash_{\mathsf{wl}} t\,u : \bullet_{\mathcal{N}}}$$

Properties of System $\mathcal{WL}$

A term is weak-linear iff it is tight-typable in system $\mathcal{WL}$.

If $\mathsf{E}(t:\sigma) \lhd (t_1, A)$, then t_1 is typable in $\mathcal{WL}$. Moreover, if the expansion is tight, so is the derivation.

If $\mathsf{E}(t:\sigma) \lhd (t', A)$ is tight, then t' is weak-linear.

$\mathcal{WL}$ gives a typing characterization to weak-linear λ-terms, unlike the typing system in Alves and Florido (2005), which typed all (but not exactly) the weak-linear terms.

Properties of System $\mathcal{WL}$

A term is weak-linear iff it is tight-typable in system $\mathcal{WL}$.

If $\mathsf{E}(t : \sigma) \lhd (t_1, A)$, then t_1 is typable in $\mathcal{WL}$. Moreover, if the expansion is tight, so is the derivation.

If $\mathsf{E}(t : \sigma) \lhd (t', A)$ is tight, then t' is weak-linear.

$\mathcal{WL}$ gives a typing characterization to weak-linear λ-terms, unlike the typing system in Alves and Florido (2005), which typed all (but not exactly) the weak-linear terms.

Properties of System $\mathcal{WL}$

A term is weak-linear iff it is tight-typable in system $\mathcal{WL}$.

If $\mathsf{E}(t : \sigma) \lhd (t_1, A)$, then t_1 is typable in $\mathcal{WL}$. Moreover, if the expansion is tight, so is the derivation.

If $\mathsf{E}(t : \sigma) \lhd (t', A)$ is tight, then t' is weak-linear.

$\mathcal{WL}$ gives a typing characterization to weak-linear λ-terms, unlike the typing system in Alves and Florido (2005), which typed all (but not exactly) the weak-linear terms.

We have omitted measures from this presentation, although they are present in both systems: $\mathcal{MX}$ and $\mathcal{WL}$.

- The number of β steps can be obtained from the number of times the abstraction rule for consumed terms is used.
- The size of the normal form can be calculated from the number of times persisting rules are applied.

Furthermore...

...it is possible to obtain measures of the expanded term from $\mathcal{MX}$.

We have omitted measures from this presentation, although they are present in both systems: $\mathcal{MX}$ and $\mathcal{WL}$.

- The number of β steps can be obtained from the number of times the abstraction rule for consumed terms is used.
- The size of the normal form can be calculated from the number of times persisting rules are applied.

Furthermore...

...it is possible to obtain measures of the expanded term from $\mathcal{MX}$.

We have omitted measures from this presentation, although they are present in both systems: $\mathcal{MX}$ and $\mathcal{WL}$.

- The number of β steps can be obtained from the number of times the abstraction rule for consumed terms is used.
- The size of the normal form can be calculated from the number of times persisting rules are applied.

Furthermore...

...it is possible to obtain measures of the expanded term from $\mathcal{MX}$.

Conclusions and Future Work

Conclusions:

- We use a quantitative system that explores the difference between persisting and consuming terms.
- We have presented an expansion relation between strongly normalising λ-terms and weak linear λ-terms preserving β-normal-forms.
- Quantitative types give an exact typing characterisation of the class of weak linear λ-terms.

Future work:

- Explore other evalutation strategies.
- What is the exact relation between the approach by Kfoury and our approach.

Conclusions and Future Work

Conclusions:

- We use a quantitative system that explores the difference between persisting and consuming terms.
- We have presented an expansion relation between strongly normalising λ-terms and weak linear λ-terms preserving β-normal-forms.
- Quantitative types give an exact typing characterisation of the class of weak linear λ-terms.

Future work:

- Explore other evalutation strategies.
- What is the exact relation between the approach by Kfoury and our approach.

Thank you!