On the Meaning of Algebraic Weights given by the GoI

Marc de Falco

Institut de Mathématiques de Luminy

GTI Workshop, Kyoto

Our problem

- GoI for MELL is **almost** sound. where almost means no ? in conclusion...

Our problem

- GoI for MELL is **almost** sound. where almost means no ? in conclusion...
- but after all, we don't really need those why not... do we?

Our problem

- GoI for MELL is **almost** sound. where almost means no ? in conclusion. . .
- but after all, we don't really need those why not... do we?
- Take λ-calculus encoding into MELL: $A \Rightarrow B \equiv ?A^{\perp} \parr B$

Our problem

- GoI for MELL is **almost** sound. where almost means no ? in conclusion. . .
- but after all, we don't really need those why not... do we?
- Take λ-calculus encoding into MELL: $A \Rightarrow B \equiv ?A^{\perp} \parr B$
- So GoI is sound for. . . computations to free variables applied to normal terms!

Our problem

- GoI for MELL is **almost** sound. where almost means no ? in conclusion. . .
- but after all, we don't really need those why not... do we?
- Take λ-calculus encoding into MELL: $A \Rightarrow B \equiv ?A^{\perp} \parr B$
- So GoI is sound for. . . computations to free variables applied to normal terms!
- Functional programming without functions is quite limiting.

Our problem

- Is it really that bad?

Our problem

- Is it really that bad?
- Take $t \rightarrow_{\beta}^{*} t'$ with $\vdash t, t' : A_1 \rightarrow \cdots \rightarrow A_n \rightarrow B$ where B is atomic.

Our problem

- Is it really that bad?
- Take $t \rightarrow_\beta^* t'$ with $\vdash t, t' : A_1 \rightarrow \cdots \rightarrow A_n \rightarrow B$ where B is atomic.
- Maybe $\mathrm{Ex}(t) \neq \mathrm{Ex}(t')$.

Our problem

- Is it really that bad?
- Take $t \rightarrow_{\beta}^{*} t'$ with $\vdash t, t' : A_1 \rightarrow \cdots \rightarrow A_n \rightarrow B$ where B is atomic.
- Maybe $\mathrm{Ex}(t) \neq \mathrm{Ex}(t')$.
- With $\vdash u_i : A_i$ we will have $\vdash (t)u_1 \cdots u_n : B$ and $\vdash (t')u_1 \cdots u_n : B$.

Our problem

- Is it really that bad?
- Take $t \rightarrow_{\beta}^{*} t'$ with $\vdash t, t' : A_1 \rightarrow \cdots \rightarrow A_n \rightarrow B$ where B is atomic.
- Maybe $\mathrm{Ex}(t) \neq \mathrm{Ex}(t')$.
- With $\vdash u_i : A_i$ we will have $\vdash (t)u_1 \cdots u_n : B$ and $\vdash (t')u_1 \cdots u_n : B$.
- So $\mathrm{Ex}((t)u_1 \cdots u_n) = \mathrm{Ex}((t')u_1 \cdots u_n)$.

Our problem

- Is it really that bad?
- Take $t \rightarrow_\beta^* t'$ with $\vdash t, t' : A_1 \rightarrow \cdots \rightarrow A_n \rightarrow B$ where B is atomic.
- Maybe $\mathrm{Ex}(t) \neq \mathrm{Ex}(t')$.
- With $\vdash u_i : A_i$ we will have $\vdash (t)u_1 \cdots u_n : B$ and $\vdash (t')u_1 \cdots u_n : B$.
- So $\mathrm{Ex}((t)u_1 \cdots u_n) = \mathrm{Ex}((t')u_1 \cdots u_n)$.
- Associativity of Ex $\Rightarrow$ we can compute it with the left formula using $\mathrm{Ex}(t)$ or with the right one using $\mathrm{Ex}(t')$.

Our problem

- Is it really that bad?
- Take $t \to_\beta^* t'$ with $\vdash t, t' : A_1 \to \cdots \to A_n \to B$ where B is atomic.
- Maybe $\mathrm{Ex}(t) \neq \mathrm{Ex}(t')$.
- With $\vdash u_i : A_i$ we will have $\vdash (t)u_1 \cdots u_n : B$ and $\vdash (t')u_1 \cdots u_n : B$.
- So $\mathrm{Ex}((t)u_1 \cdots u_n) = \mathrm{Ex}((t')u_1 \cdots u_n)$.
- Associativity of Ex $\Rightarrow$ we can compute it with the left formula using $\mathrm{Ex}(t)$ or with the right one using $\mathrm{Ex}(t')$.
- So computationnaly $\mathrm{Ex}(t) \sim \mathrm{Ex}(t')$.

Our problem

- Focus on Danos-Regnier view of Geometry of Interaction.

Our problem

- Focus on Danos-Regnier view of Geometry of Interaction.
- GoI semantics sliced into a set of algebraic weights with a one-to-one correspondence with meaningful paths in proofs/programs.

Our problem

- Focus on Danos-Regnier view of Geometry of Interaction.
- GoI semantics sliced into a set of algebraic weights with a one-to-one correspondence with meaningful paths in proofs/programs.

Problem

Can we deduce that $\text{Ex}(t) \sim \text{Ex}(t')$ by looking at weights only?

Our problem

- Focus on Danos-Regnier view of Geometry of Interaction.
- GoI semantics sliced into a set of algebraic weights with a one-to-one correspondence with meaningful paths in proofs/programs.

Problem

Can we deduce that $\mathrm{Ex}(t) \sim \mathrm{Ex}(t')$ by looking at weights only?

Equivalent Problem

Can we give a meaning to each weight such that the global meaning is preserved by reduction?

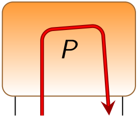
Outline

Paths and reduction

- Representation of a proof/program via a graph-like syntax: *e.g. proofnets, interaction nets*.
- We consider *straights* and *maximal* paths: $\mathfrak{P}(R)$.
- Cut-elimination is translated into a path reduction:
  $\delta_{\mathcal{R}} : \mathfrak{P}(R) \mapsto \mathfrak{P}(R')$ for $R \xrightarrow{\mathcal{R}} R'$.

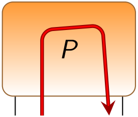
Persistent paths

- Throughout reduction paths can be



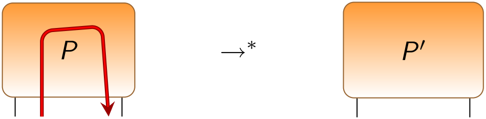
Persistent paths

- Throughout reduction paths can be



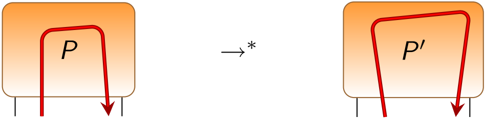
Persistent paths

- Throughout reduction paths can be



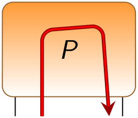
Persistent paths

- Throughout reduction paths can be deformed (possibly duplicated)

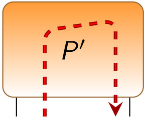


Persistent paths

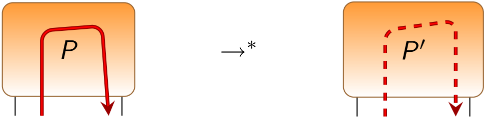
- Throughout reduction paths can be deformed (possibly duplicated) or destroyed.



$\longrightarrow^*$



Persistent paths

- Throughout reduction paths can be deformed (possibly duplicated) or destroyed. 

- Persistent paths: survive all possible reductions (needs a confluent and normalizing system to make sense).

Regular paths

- M is an *inverse monoid with zero* (imz) iff

Regular paths

- M is an *inverse monoid with zero* (imz) iff
  - M is a monoid

Regular paths

- M is an *inverse monoid with zero* (imz) iff
  - M is a monoid
  - with a zero element: $\forall x, x0 = 0x = 0$

Regular paths

- M is an *inverse monoid with zero* (imz) iff
  - M is a monoid
  - with a zero element: $\forall x, x0 = 0x = 0$
  - with an inverse for each element: $\forall x, \exists x^\star, xx^\star x = x$ and $x^\star x x^\star = x^\star$.

Regular paths

- M is an *inverse monoid with zero* (imz) iff
  - M is a monoid
  - with a zero element: $\forall x, x0 = 0x = 0$
  - with an inverse for each element: $\forall x, \exists x^\star, xx^\star x = x$ and $x^\star xx^\star = x^\star$.
- Path weighting: a function $w : \mathfrak{P}(R) \mapsto M$ such that

Regular paths

- M is an *inverse monoid with zero* (imz) iff
  - M is a monoid
  - with a zero element: $\forall x, x0 = 0x = 0$
  - with an inverse for each element: $\forall x, \exists x^\star, xx^\star x = x$ and $x^\star x x^\star = x^\star$.
- Path weighting: a function $\mathrm{w} : \mathfrak{P}(R) \mapsto \mathrm{M}$ such that
  - $\mathrm{w}(\varphi_1\varphi_2) = \mathrm{w}(\varphi_2)\mathrm{w}(\varphi_1)$

Regular paths

- M is an *inverse monoid with zero* (imz) iff
  - M is a monoid
  - with a zero element: $\forall x, x0 = 0x = 0$
  - with an inverse for each element: $\forall x, \exists x^\star, xx^\star x = x$ and $x^\star x x^\star = x^\star$.
- Path weighting: a function $\mathrm{w} : \mathfrak{P}(R) \mapsto \mathrm{M}$ such that
  - $\mathrm{w}(\varphi_1\varphi_2) = \mathrm{w}(\varphi_2)\mathrm{w}(\varphi_1)$
  - $\mathrm{w}(\varphi^r) = \mathrm{w}(\varphi)^\star$ where φ^r is the reversal of φ.

Regular paths

- M is an *inverse monoid with zero* (imz) iff
  - M is a monoid
  - with a zero element: $\forall x, x0 = 0x = 0$
  - with an inverse for each element: $\forall x, \exists x^\star, xx^\star x = x$ and $x^\star x x^\star = x^\star$.
- Path weighting: a function $\mathrm{w} : \mathfrak{P}(R) \mapsto \mathrm{M}$ such that
  - $\mathrm{w}(\varphi_1\varphi_2) = \mathrm{w}(\varphi_2)\mathrm{w}(\varphi_1)$
  - $\mathrm{w}(\varphi^r) = \mathrm{w}(\varphi)^\star$ where φ^r is the reversal of φ.
- *Regular* paths: $\mathrm{w}(\varphi) \neq 0$

GoI from the Danos-Regnier point of view

Sketch of Definition

We call *Geometry of Interaction* for a logical system $\mathcal{L}$ a family of weighting functions w_R for all $R \in \mathcal{L}$ targetting the same imz and such that:

$$\varphi \text{ persistent } \iff \varphi \text{ regular}$$

GoI from the Danos-Regnier point of view

Sketch of Definition

We call *Geometry of Interaction* for a logical system $\mathcal{L}$ a family of weighting functions w_R for all $R \in \mathcal{L}$ targetting the same imz and such that:

$$\varphi \text{ persistent} \iff \varphi \text{ regular}$$

- Let $\mathbb{C}[\mathrm{M}]$ be the $\mathbb{C}$-algebra generated by M where the zero of M is also the zero of the algebra (its closure is a $\mathbb{C}^\star$-algebra)

GoI from the Danos-Regnier point of view

Sketch of Definition

We call *Geometry of Interaction* for a logical system $\mathcal{L}$ a family of weighting functions w_R for all $R \in \mathcal{L}$ targetting the same imz and such that:

$$\varphi \text{ persistent} \iff \varphi \text{ regular}$$

- Let $\mathbb{C}[\mathrm{M}]$ be the $\mathbb{C}$-algebra generated by M where the zero of M is also the zero of the algebra (its closure is a $\mathbb{C}^\star$-algebra)
- For any R we can define $\mathrm{Ex}(R) = \sum_{\varphi \in \mathfrak{P}(R)} \mathrm{w}(\varphi)$

GoI from the Danos-Regnier point of view

Sketch of Definition

We call *Geometry of Interaction* for a logical system $\mathcal{L}$ a family of weighting functions w_R for all $R \in \mathcal{L}$ targetting the same imz and such that:

$$\varphi \text{ persistent } \iff \varphi \text{ regular}$$

- Let $\mathbb{C}[\mathrm{M}]$ be the $\mathbb{C}$-algebra generated by M where the zero of M is also the zero of the algebra (its closure is a $\mathbb{C}^\star$-algebra)
- For any R we can define $\mathrm{Ex}(R) = \sum_{\varphi \in \mathfrak{P}(R)} \mathrm{w}(\varphi)$
- We recover the standard definition of the geometry of interaction.

GoI from the Danos-Regnier point of view

Sketch of Definition

We call *Geometry of Interaction* for a logical system $\mathcal{L}$ a family of weighting functions w_R for all $R \in \mathcal{L}$ targetting the same imz and such that:

$$\varphi \text{ persistent} \iff \varphi \text{ regular}$$

- Let $\mathbb{C}[\mathrm{M}]$ be the $\mathbb{C}$-algebra generated by M where the zero of M is also the zero of the algebra (its closure is a $\mathbb{C}^\star$-algebra)
- For any R we can define $\mathrm{Ex}(R) = \sum_{\varphi \in \mathfrak{P}(R)} \mathrm{w}(\varphi)$
- We recover the standard definition of the geometry of interaction.
- $\mathrm{Ex}(R)$ is not necessarily equal to $\mathrm{Ex}(R')$ when $R \to R'$

Outline

Nets for λ-calculus

- We are going to represent λ-calculus via a translation into MELL proofnets

Nets for λ-calculus

- We are going to represent λ-calculus via a translation into MELL proofnets
- MELL proofnets are going to be presented via a mix between sharing graphs (i.e. numbered interaction nets) and Regnier's new syntax for MELL proofnets

Nets for λ-calculus

- We are going to represent λ-calculus via a translation into MELL proofnets
- MELL proofnets are going to be presented via a mix between sharing graphs (i.e. numbered interaction nets) and Regnier's new syntax for MELL proofnets
- This syntax is a simplification for studying GoI without dealing with unnecessary issues.

Nets for λ-calculus

- We are going to represent λ-calculus via a translation into MELL proofnets
- MELL proofnets are going to be presented via a mix between sharing graphs (i.e. numbered interaction nets) and Regnier's new syntax for MELL proofnets
- This syntax is a simplification for studying GoI without dealing with unnecessary issues.
- Disclaimer: it might be a little technical. . .

The translation

- We construct nets made from numbered cells:

where $n \in \mathbb{N}$ is called the level and x is a λ-calculus variable.

The translation

- We construct nets made from numbered cells:

  where $n \in \mathbb{N}$ is called the level and x is a λ-calculus variable.
- For t a λ-term, $\mathrm{FV}_o(t)$ = the set of occurrences of free variables in t.

The translation

- We construct nets made from numbered cells:

  where $n \in \mathbb{N}$ is called the level and x is a λ-calculus variable.
- For t a λ-term, $\mathrm{FV}_o(t) =$ the set of occurrences of free variables in t.
- For any $n \in \mathbb{N}$ and λ-term t we build a net $[t]_n$ with a one-to-one labelling of conclusions by $\mathrm{FV}_o(t) \cup \{\bullet\}$.

The translation

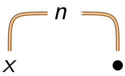
- We construct nets made from numbered cells:

where $n \in \mathbb{N}$ is called the level and x is a λ-calculus variable.

- For t a λ-term, $\mathrm{FV}_o(t) =$ the set of occurrences of free variables in t.
- For any $n \in \mathbb{N}$ and λ-term t we build a net $[t]_n$ with a one-to-one labelling of conclusions by $\mathrm{FV}_o(t) \cup \{\bullet\}$.
- The translation of t will be the net $[t] = [t]_0$.

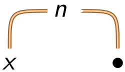
The translation

variable $[x]_n =$

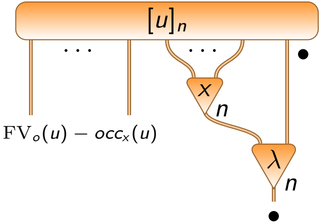


The translation

variable $[x]_n =$



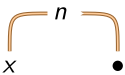
abstraction $[\lambda x.u]_n =$



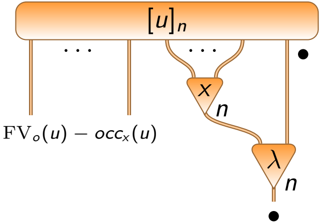
where we have collected all free occurrences of x.

The translation

variable $[x]_n =$

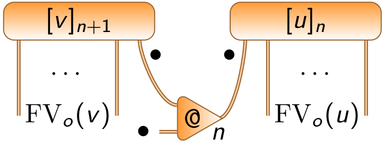


abstraction $[\lambda x.u]_n =$



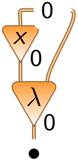
where we have collected all free occurrences of x.

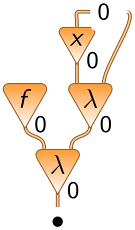
application $[(u)v]^\bullet =$



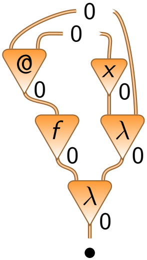
Some examples

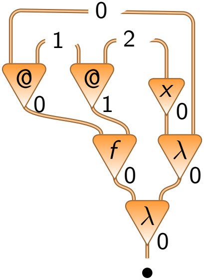
nth Church integer: $\overline{n} = \lambda f.\lambda x.(f)^n x$

$[\lambda x.x] =$ 

$, [\overline{0}] = [\lambda f.\lambda x.x] =$ 

,

$[\overline{1}] = [\lambda f.\lambda x.(f)x] =$ 

$, [\overline{2}] =$ 

$, \ldots$

Reduction

- We can mimic β-reduction with a big-step reduction coming from MELL proofnets.
- To do so we need to rebuild boxes: connected components of a minimum level.
- We can define in an obvious way paths and their reductions.

The weighting imz

Let M be the imz generated

- by constants: $\{p, q\} \cup \{x_{n,c}\ ,\ x \in V, n, c \in \mathbb{N}\}$
- a morphism !
- and relations:

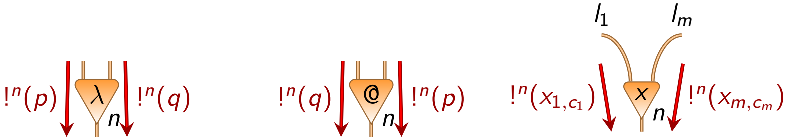
$$p^\star p = q^\star q = 1$$

$$q^\star p = p^\star q = 0$$

$$x^\star_{i,c} x_{j,d} = \delta_{ij}\delta_{cd}$$

$$!(u)x_{i,c} = x_{i,c}!^c(u), \forall u \in \mathrm{M}$$

We define a weighting of paths with



where $c_i = l_i - n$

Bentobako semantics

- A *bentobako* is a finite sequence of elements of the form 0, 1 or $x_i[b]$ where $x \in V, i \in \mathbb{N}$ and b is a bentobako.

Bentobako semantics

- A *bentobako* is a finite sequence of elements of the form 0, 1 or $x_i[b]$ where $x \in V, i \in \mathbb{N}$ and b is a bentobako.
- We write $a \circ b$ for the concatenate of a and b,

Bentobako semantics

- A *bentobako* is a finite sequence of elements of the form 0, 1 or $x_i[b]$ where $x \in V, i \in \mathbb{N}$ and b is a bentobako.
- We write $a \circ b$ for the concatenate of a and b,
- if $b = e_1, \cdots, e_n, e_{n+1}, \cdots, e_m$, we write $b_n = e_1, \cdots, e_n$ and $b^n = e_{n+1}, \cdots, e_m$.

Bentobako semantics

- A *bentobako* is a finite sequence of elements of the form 0, 1 or $x_i[b]$ where $x \in V, i \in \mathbb{N}$ and b is a bentobako.
- We write $a \circ b$ for the concatenate of a and b,
- if $b = e_1, \cdots, e_n, e_{n+1}, \cdots, e_m$, we write $b_n = e_1, \cdots, e_n$ and $b^n = e_{n+1}, \cdots, e_m$.
- We interpret weights by partial and reversible operations on bentobakos.

Bentobako semantics

- A *bentobako* is a finite sequence of elements of the form 0, 1 or $x_i[b]$ where $x \in V, i \in \mathbb{N}$ and b is a bentobako.
- We write $a \circ b$ for the concatenate of a and b,
- if $b = e_1, \cdots, e_n, e_{n+1}, \cdots, e_m$, we write $b_n = e_1, \cdots, e_n$ and $b^n = e_{n+1}, \cdots, e_m$.
- We interpret weights by partial and reversible operations on bentobakos.
- $\mathbf{p}(b) = 0 \circ b$, $\mathbf{q}(b) = 1 \circ b$

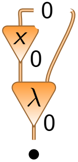
Bentobako semantics

- A *bentobako* is a finite sequence of elements of the form 0, 1 or $x_i[b]$ where $x \in V, i \in \mathbb{N}$ and b is a bentobako.
- We write $a \circ b$ for the concatenate of a and b,
- if $b = e_1, \cdots, e_n, e_{n+1}, \cdots, e_m$, we write $b_n = e_1, \cdots, e_n$ and $b^n = e_{n+1}, \cdots, e_m$.
- We interpret weights by partial and reversible operations on bentobakos.
- $\mathbf{p}(b) = 0 \circ b$, $\mathbf{q}(b) = 1 \circ b$
- $\mathbf{x_{i,c}}(b) = x_i[b_c] \circ b^c$

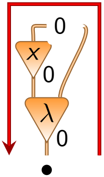
Bentobako semantics

- A *bentobako* is a finite sequence of elements of the form 0, 1 or $x_i[b]$ where $x \in V, i \in \mathbb{N}$ and b is a bentobako.
- We write $a \circ b$ for the concatenate of a and b,
- if $b = e_1, \cdots, e_n, e_{n+1}, \cdots, e_m$, we write $b_n = e_1, \cdots, e_n$ and $b^n = e_{n+1}, \cdots, e_m$.
- We interpret weights by partial and reversible operations on bentobakos.
- $\mathbf{p}(b) = 0 \circ b$, $\mathbf{q}(b) = 1 \circ b$
- $\mathbf{x_{i,c}}(b) = x_i[b_c] \circ b^c$
- $\mathbf{!^n}(\mathbf{w})(b) = b_n \circ \mathbf{w}(b^n)$

Example: $\lambda x.x$

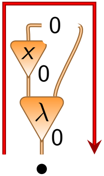
- $[\lambda x.x] =$ 

Example: $\lambda x.x$

- $[\lambda x.x] =$ 

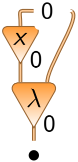
- Two regular paths of weights: $i = px_{1,0}q^{\star}$

Example: $\lambda x.x$

- $[\lambda x.x] =$ 

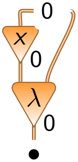
- Two regular paths of weights: $i = px_{1,0}q^\star$ and $i^\star = qx^\star_{1,0}p^\star$.

Example: $\lambda x.x$

- $[\lambda x.x] =$ 

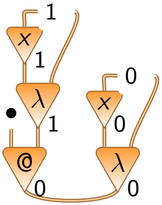
- Two regular paths of weights: $i = px_{1,0}q^\star$ and $i^\star = qx^\star_{1,0}p^\star$.
- Operations: $\mathbf{i}(1 \circ b) = 0 \circ x_1[] \circ b$ and $\mathbf{i}^\star(0 \circ x_1[] \circ b) = 1 \circ b$.

Example: $\lambda x.x$

- $[\lambda x.x] =$ 

- Two regular paths of weights: $i = px_{1,0}q^\star$ and $i^\star = qx^\star_{1,0}p^\star$.
- Operations: $\mathbf{i}(1 \circ b) = 0 \circ x_1[] \circ b$ and $\mathbf{i}^\star(0 \circ x_1[] \circ b) = 1 \circ b$.
- The number of regular paths is always even. We will only present half of them.

Example: $(\lambda x.x)\lambda x.x$

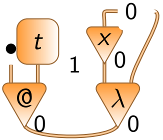
- $[(\lambda x.x)\lambda x.x] =$ 

- $1 \circ b \xrightarrow{\mathbf{q}} 1 \circ 1 \circ b \xrightarrow{\mathbf{i}} 0 \circ x_1[] \circ 1 \circ b$
  $\xrightarrow{\mathbf{p}^\star} x_1[] \circ 1 \circ b \xrightarrow{!(\mathbf{i})} x_1[] \circ 0 \circ x_1[] \circ b \xrightarrow{\mathbf{p}} 0 \circ x_1[] \circ 0 \circ x_1[] \circ b$
  $\xrightarrow{\mathbf{i}^\star} 1 \circ 0 \circ x_1[] \circ b \xrightarrow{\mathbf{q}^\star} 0 \circ x_1[] \circ b$
- $q^\star i^\star p!(i)p^\star iq = q^\star q x^\star_{1,0} p^\star p!(px_{1,0}q^\star)p^\star p x_{1,0} q^\star q = x^\star_{1,0}!(px_{1,0}q^\star)x_{1,0} =$
  $px_{1,0}q^\star x^\star_{1,0} x_{1,0} = i$
- The two methods are equivalent thanks to

Lemma (Danos-Regnier)

If w and w' are weights of paths in a λ-term then $w = w' \iff \mathbf{w} = \mathbf{w}'$

Example: $(\lambda x.x)t$

- Direct generalization of the previous case.
- $[(\lambda x.x)t] =$ 

- For any path of weight w in t, we have:
- $b \xrightarrow{\mathbf{p^\star iq}} x_1[] \circ b \xrightarrow{\mathbf{!(w)}} x_1[] \circ \mathbf{w}(b) \xrightarrow{\mathbf{q^\star i^\star p}} \mathbf{w}(b)$
- $q^\star i^\star p!(w)p^\star iq = q^\star q x^\star_{1,0} p^\star p!(w)p^\star p x_{1,0} q^\star q = x^\star_{1,0}!(w)x_{1,0} = w x^\star_{1,0} x_{1,0} = w$

Example: $(\lambda x.x)t$

- Finding the meaning of $i = px_{1,0}q^\star$.
- Interactive procedure:

$$
\begin{array}{rcl}
b\bullet & \longrightarrow & \bullet 1 \circ b \\
 & & \big\downarrow \mathbf{i} \\
x_1[] \circ b\bullet & \longleftarrow & \bullet 0 \circ x_1[] \circ b \\
\big\downarrow & & \\
x_1[] \circ \mathbf{w}(b)\bullet & \longrightarrow & \bullet 0 \circ x_1[] \circ \mathbf{w}(b) \\
 & & \big\downarrow \mathbf{i}^\star \\
\mathbf{w}(b)\bullet & \longleftarrow & \bullet 1 \circ \mathbf{w}(b)
\end{array}
$$

- $\lambda x.x$ acts as a perfect intermediary, it prepends and postpends any path, in a reversible way.
- During computation each part of the token has a meaning:

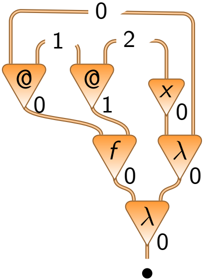
$$1 \circ b \rightarrow 0 \circ x_1[] \circ b$$

Query for value (1)

Query for argument value (0)

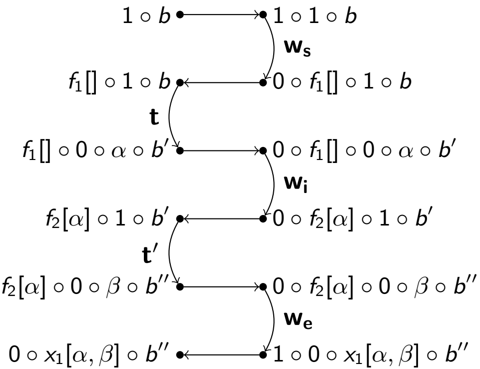
Internal state ($x_1[]$)

Example: $\overline{2}$

- $[\overline{2}] =$ 

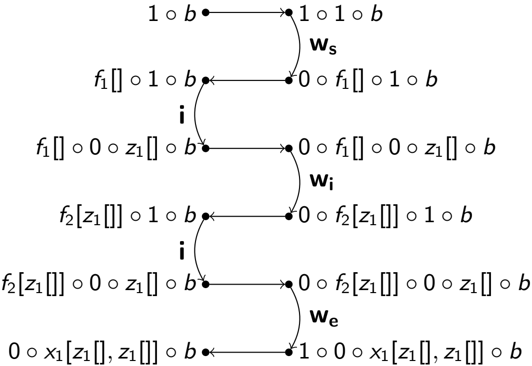
- Regular weights: $w_s = pf_{1,0}qq^\star q^\star$, $w_i = pf_{2,1}!(q)p^\star f^\star_{1,0}p^\star$, $w_e = qpx_{0,2}!(p^\star)f^\star_{2,1}p^\star$
- Operations: $\mathbf{w_s}(1 \circ 1 \circ b) = 0 \circ f_1[] \circ 1 \circ b$,
$\mathbf{w_i}(0 \circ f_1[] \circ 0 \circ e \circ b) = 0 \circ f_2[e] \circ 1 \circ b$
$\mathbf{w_e}(0 \circ f_2[e] \circ 0 \circ e' \circ b) = 1 \circ 0 \circ x_1[e, e'] \circ b$

Example: $(\overline{2})t$



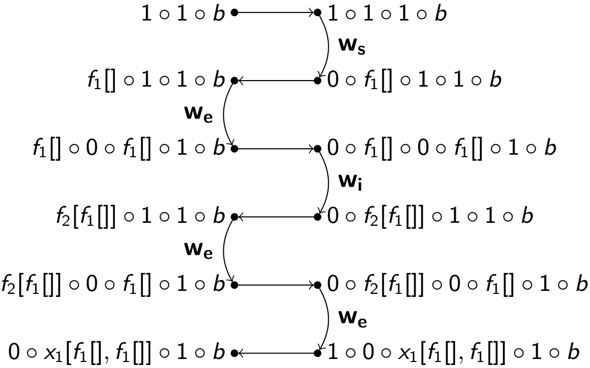
- Paths in t are expected to be of the shape $1 \circ b \rightarrow 0 \circ \sigma \circ b'$.
- This is the case when t is a function answering to a query by querying the value of its argument.
- $\lambda x.x$ and $\overline{2}$ are of this kind.

Example: $(\bar{2})\lambda z.z$



- $(\bar{2})\lambda z.z \rightarrow \lambda x.(\lambda z.z)(\lambda z.z)x \rightarrow^2 \lambda x.x$
- $px_{1,2}z_{1,0}z_{1,0}q^\star$ is different from $i = px_{1,0}q^\star$.
- But they have the same meaning!

Example: $(\overline{2})\overline{2}$



- Computation: $q^\star w_e p!(w_s) p^\star w_i p!(w_s) p^\star w_s q$
- Weight: $w_{ss} = p x_{1,2} f_{1,0} f_{1,0} q q^\star q^\star$

Example: $(\overline{2})\overline{2}$

-
  - Computation: $q^\star w_e p!(w_s) p^\star w_i p!(w_s) p^\star w_s q$
  - Weight: $w_{ss} = p x_{1,2} f_{1,0} f_{1,0} q q^\star q^\star$
  - Operation: $1 \circ 1 \circ b \longrightarrow 0 \circ x_1[f_1[], f_1[]] \circ 1 \circ b$
- The other paths are a lot more complicated. Let's use a program!
-
  - Computation: $q^\star w_e p!(w_i) p^\star w_e^\star q$
  - Weight: $w_{.i} = p x_{1,2}!(f_{2,1}!(q) p^\star f_{1,0}^\star) x_{1,2}^\star p^\star$
  - Operation: $0 \circ x_1[\alpha, f_1[]] \circ 0 \circ \beta \circ b \longrightarrow 0 \circ x_1[\alpha, f_2[\beta]] \circ 1 \circ b$
-
  - Computation: $q^\star w_e p!(w_s) p^\star w_i p!(w_i) p^\star w_i^\star p!(w_e) p^\star w_e^\star q$
  - Weight: $w_{i.} = p x_{1,2} f_{2,1} x_{1,2}!(!(f_{1,0} q) p^\star) f_{2,1}^\star f_{1,0}^\star x_{1,2}^\star p^\star$.
  - Operation: $0 \circ x_1[f_1[], f_2[\alpha]] \circ 0 \circ \beta \circ b \longrightarrow 0 \circ x_1[f_2[x_1[\alpha, \beta]], f_1[]] \circ 1 \circ b$
-
  - Computation: $q^\star w_s^\star p!(w_e) p^\star w_i^\star p!(w_e) p^\star w_e^\star q$
  - Weight: $w_{ee} = q p x_{1,2}!(x_{1,2}!(p^\star) f_{2,1}^\star) f_{2,1}^\star x_{1,2}^\star p^\star$
  - Operation: $0 \circ x_1[f_2[\alpha], f_2[\beta]] \circ 0 \circ \gamma \circ b \longrightarrow 1 \circ 0 \circ x_1[\alpha, x_1[\beta, \gamma]] \circ b$

Example: $\overline{4}$

- $v_s = pf_{1,0}qq^\star q^\star$, $1 \circ 1 \circ b \rightarrow 0 \circ f_1[] \circ 1 \circ b$
- $v_{i1} = pf_{2,1}!(q)p^\star f^\star_{1,0}p^\star$, $0 \circ f_1[] \circ 0 \circ \alpha \circ b \rightarrow 0 \circ f_2[\alpha] \circ 1 \circ b$
- $v_{i2} = pf_{3,2}!(!(q)p^\star)f^\star_{2,1}p^\star$, $0 \circ f_2[\alpha] \circ 0 \circ \beta \circ b \rightarrow 0 \circ f_3[\alpha,\beta] \circ 1 \circ b$
- $v_{i3} = pf_{4,3}!^2(!(q)p^\star)f^\star_{3,2}p^\star$, $0 \circ f_3[\alpha,\beta] \circ 0 \circ \gamma \circ b \rightarrow 0 \circ f_4[\alpha,\beta,\gamma] \circ 1 \circ b$
- $v_e = qpx_{1,4}!^3(p^\star)f^\star_{4,3}p^\star$, $0 \circ f_4[\alpha,\beta,\gamma] \circ 0 \circ \delta \circ b \rightarrow 1 \circ 0 \circ x_1[\alpha,\beta,\gamma,\delta] \circ b$
- It has the same meaning as $(\overline{2})\overline{2}$ while using an extra path: $w_{.i}$ is a compression of v_{i1} and v_{i3}.

General situation

Theorem

There are $4 + 2n(n-1)$ *regular paths in* $(\overline{n})\overline{n}$ *of weight:*

$$\sigma_n = px_{1,n}f_{1,0}^n qq^\star q^\star$$

for $k < n-1$ *and* $l \leq n-1$*:*

$$\iota_n(k,l) = px_{1,n}!^{n-l-1}(f_{k+2,k+1}!^k(\beta^l(!(f_{1,0}^l q)p^\star))f_{k+1,k}^\star)x_{1,n}^\star p^\star$$

$$\epsilon_n = qp\beta^n(p^\star)x_{1,n}^\star p^\star$$

with $\beta(w) = x_{1,n}!^{n-1}(w)f_{n,n-1}^\star$.

Sketch of proof:

- Show that there exists regular paths of these weights by computation.
- Show that they are the only ones by showing that there is no room for others in a semantics.

Outline

What is the meaning of weights

From the previous example we have a candidate notion for the meaning.

Sketch of definition

$E, F \subseteq \mathrm{M}$, we have $E \sim F$ if they lead to the same kind of computations.

First try at a definition

What we would like to define as having the same meaning:

Definition

Let $E, F \subseteq \mathrm{M}$ be self-dual, we say that $E \sim F$ when for all $g \in \mathrm{M}$ there exists a bijective function from $\{e \in E \ , \ eg \neq 0\}$ to $\{f \in F \ , \ fg \neq 0\}$

Unfortunately it is too restrictive because g can be anything:

- $i = px_{1,0}q^\star$
- $j = px_{1,2}z_{1,0}z_{1,0}q^\star$
- Take $g = px_{1,0}$, we have $i^\star g, ig \neq 0$ but $j^\star g = 0$.
- So $\{i, i^\star\} \not\sim \{j, j^\star\}$.

The problem comes from the fact that g can never appear in the context of λ-terms.

A proper definition for λ-caclulus

Lemma

Let t be a λ-term and $C\langle\rangle$ a context with one hole. For all path φ in $C\langle t\rangle$, long enough with respect to t, there is a function of maximal arity $f_\varphi : \mathrm{M}^n \to \mathrm{M}$ such that there exists $e_1, \cdots, e_n \in \mathrm{Ex}([t])$ with $\mathrm{w}(\varphi) = f_\varphi(e_1, \cdots, e_n)$.

Definition

Let t and t be λ-term and $C\langle\rangle$ a context with one hole. We say that $t \leq t'$ when for all φ in $C\langle t\rangle$, long enough with respect to t, there exists $e'_1, \cdots, e'_n \in \mathrm{Ex}([t'])$ such that

$$f_\varphi(e_1, \cdots, e_n) = 0 \iff f_\varphi(e'_1, \cdots, e'_n) = 0$$

$t \sim t'$ when $t \leq t'$ and $t' \leq t$

Can we decide this relation?

Is $t \sim t'$ implying that there exists t_0 such that $t \to^* t_0$ and $t' \to^* t_0$?