The atomic $\lambda\mu$-calculus

Fanny He
f.he@bath.ac.uk

UNIVERSITY OF BATH

The atomic $\lambda\mu$-calculus

Classical logic and computation

Deep Inference and computation

The atomic $\lambda\mu$-calculus: $\Lambda\mu S_a$

Properties of $\Lambda\mu S_a$

Classical Logic and Curry-Howard

Logic	Computation
Intuitionistic	
Formula	Program type
Proof	Program
Natural Deduction	The λ-calculus
$A \rightarrow (B \rightarrow A)$	$A \rightarrow (B \rightarrow A)$ types $\lambda x^A.\lambda y^B.x$
Classical	
Natural Deduction	?
$\neg\neg A \rightarrow A$	?

Lafont's critical pair

$$
\dfrac{\dfrac{\begin{array}{c}\Pi_1\\ \vdots\\ \Gamma \vdash \Delta\end{array}}{\Gamma, A \vdash \Delta}\ \texttt{Lwk} \qquad \dfrac{\begin{array}{c}\Pi_2\\ \vdots\\ \Gamma' \vdash \Delta'\end{array}}{\Gamma' \vdash \Delta', A}\ \texttt{Rwk}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \texttt{Cut}
$$

There are two proofs of $\Gamma, \Gamma' \vdash \Delta, \Delta'$ without cuts:

$$
\dfrac{\begin{array}{c}\Pi_1\\ \vdots\\ \Gamma \vdash \Delta\end{array}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \texttt{wk}
$$

$$
\dfrac{\begin{array}{c}\Pi_2\\ \vdots\\ \Gamma' \vdash \Delta'\end{array}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \texttt{wk}
$$

Curry-Howard correspondence

Logic	Computation
Intuitionistic	
Formula	Program type
Proof	Program
Natural Deduction	The λ-calculus
$A \to (B \to A)$	$A \to (B \to A)$ types $\lambda x^A.\lambda y^B.x$
Classical	
Natural Deduction	The $\lambda\mu$-calculus [Parigot]
$\neg\neg A \to A$ $(;\bot)$	Types $\lambda y.\mu\alpha.((y)\ \lambda x.\mu\delta.(x)\alpha)\phi$
$((A \to B) \to A) \to A$	Types $\lambda y.\mu\alpha.((y)\lambda x.\mu\delta.(x)\alpha)\alpha$
Double negation translation	CPS translation

Deep Inference and Curry-Howard

- Apply inference rules at any depth
- Open deduction [Guglielmi, Gundersen, Parigot]

$$\cfrac{\cfrac{A \wedge B}{A \wedge D}}{C \wedge D} \equiv \cfrac{\cfrac{A \wedge B}{C \wedge B}}{C \wedge D} \equiv \frac{A}{C} \wedge \frac{B}{D}$$

Logic	Computation
Intuitionistic	
Formula	Program type
Proof	Program
Open Deduction	The atomic λ-calculus Λ_a [Gundersen, Heijltjes, Parigot]
$A \to (B \to A)$	$A \to (B \to A)$ types $\lambda x^A.\lambda y^B.x[\leftarrow y]$
Classical	
Open Deduction	?
$\neg\neg A \to A$	?

The atomic λ-calculus Λ_a (informally)

- A linear λ-calculus
  - with sharings (similar to explicit substitutions)
  - with atomic duplications (similar to optimal reduction graphs)

Type system in Open Deduction:

$$\begin{array}{c} A \\ \Downarrow \\ C \end{array} ::= A \mid \begin{array}{c} A \\ \Downarrow \\ C \end{array} \wedge \begin{array}{c} B \\ \Downarrow \\ D \end{array} \mid \begin{array}{c} A \\ \Downarrow \\ C \end{array} \to \begin{array}{c} B \\ \Downarrow \\ D \end{array} \mid \begin{array}{c} A \\ \Downarrow \\ \dfrac{B}{B'}\,\mathsf{r} \\ \Downarrow \\ C \end{array}$$

(for $\to$, $A = C \to B$ and $C = A \to D$)

Deep Inference and Λ_a

Medial rule:

$$\frac{(A \vee B) \to (C \wedge D)}{(A \to C) \wedge (B \to D)}\,\mathsf{m}$$

Allows atomic contraction:

$$\frac{A \to B}{(A \to B) \wedge (A \to B)}\,\triangle \rightsquigarrow \frac{\dfrac{A \vee A}{A}\,\triangledown \to \dfrac{B}{B \wedge B}\,\triangle}{(A \to B) \wedge (A \to B)}\,\mathsf{m}$$

The atomic λ-calculus allows duplication of individual constructors:

$$\frac{\lambda x.t : A \to B}{(\lambda x.t : A \to B) \wedge (\lambda x.t : A \to B)}\,\triangle \rightsquigarrow \frac{\dfrac{(x : A) \vee (x : A)}{(x : A)}\,\triangledown \to \dfrac{(t : B)}{(t : B) \wedge (t : B)}\,\triangle}{(\lambda x.t : A \to B) \wedge (\lambda x.t : A \to B)}\,\mathsf{m}$$

Properties of Λ_a

- Confluence, typed SN, subject reduction
- PSN
- Fully-lazy sharing
  - Share resources
  - Restrict duplications

Deep Inference and Curry-Howard

Logic	Computation
Rules reduced to their atomic forms	Sharing of individual constructors
Intuitionistic	
Formula	Program type
Proof	Program
Open Deduction	Λ_a
$A \rightarrow (B \rightarrow A)$	$A \rightarrow (B \rightarrow A)$ types $\lambda x^A.\lambda y^B.x[\leftarrow y]$
Classical	
Open Deduction	The atomic $\lambda\mu$-calculus $\Lambda\mu S_a$
$\neg\neg A \rightarrow A$	$\lambda y.\mu\alpha.((y)\ \lambda x.\mu\delta.(x)\alpha[\leftarrow \delta])\phi$

How to construct the atomic $\lambda\mu$-calculus

Aims:

- Keep the good properties of $\lambda\mu$ and Λ_a
- Give a computational interpretation of a classical deep inference system

Approach:

- Naturally extend $\lambda\mu$ and Λ_a
- Adapt classical proofs with multiple conclusions to open deduction

Syntax of $\lambda\mu$ and Λ_a

- The $\lambda\mu$-calculus:

$$\begin{aligned} \textit{Terms } t,u &::= x \mid \lambda x.t \mid (t)u \mid \mu\alpha.n \\ \mu\textit{-variables } &::= \alpha \\ \textit{Names } n &::= (t)\alpha \end{aligned}$$

$$(\mu\alpha.n)u \longrightarrow_\mu \mu\alpha.\ n\{(w)\ u\alpha/(w)\alpha\}$$

- The atomic λ-calculus:

$$\begin{aligned} \textit{Terms } t,u &::= x \mid \lambda x.t \mid (t)u \mid u[\phi] \\ \lambda\textit{-tuples } t^p &::= \langle\, t_1,\ldots,t_p \,\rangle \mid t^p[\phi] \\ \textit{Closures } [\phi],[\psi] &::= [\vec{x_q} \leftarrow t] \mid [\vec{x_q} \twoheadleftarrow \lambda y.t^q] \end{aligned}$$

Syntax of $\lambda\mu$ and Λ_a

- The atomic λ-calculus:

$$\begin{aligned} &\textit{Terms } t, u ::= \ x \mid \lambda x.t \mid (t)u \mid u[\phi] \\ &\lambda\textit{-tuples } t^p ::= \langle\, t_1, \ldots, t_p \,\rangle \mid t^p[\phi] \\ &\textit{Closures } [\phi], [\psi] ::= \ [\vec{x_q} \leftarrow t] \mid [\vec{x_q} \twoheadleftarrow \lambda y.t^q] \end{aligned}$$

Examples:

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

$$(\lambda x.(y)\, x_1\, x_2[x_1, x_2 \leftarrow x])\, t \to_\beta (y)\, x_1\, x_2[x_1, x_2 \leftarrow t]$$

$$\begin{aligned} [x_1, x_2 \leftarrow \lambda z.z] &\rightsquigarrow [x_1, x_2 \twoheadleftarrow \lambda z.\langle z_1, z_2\rangle[z_1, z_2 \leftarrow z]] \\ &\rightsquigarrow \{(\lambda z_1.z_1)/x_1\}\{(\lambda z_2.z_2)/x_2\} \end{aligned}$$

First syntax towards $\Lambda\mu S_a$

- How to define sharings with names/μ-variables?

$$
\begin{aligned}
&\mu\text{-}variables \ ::= \ \alpha \\
&Names\ n ::= (t)\alpha \mid n[\phi] \\
&Terms\ t, u ::= \ x \mid \lambda x.t \mid (t)u \mid \mu\alpha.n \mid u[\phi] \\
&\lambda\text{-}tuples\ t^p ::= \langle\, t_1, \ldots, t_p \,\rangle \mid t^p[\phi] \\
&Closures\ [\phi], [\psi] ::= \ [\vec{x_q} \leftarrow t] \mid [\vec{x_q} \twoheadleftarrow \lambda y.t^q] \mid ????
\end{aligned}
$$

First syntax towards $\Lambda\mu S_a$

An example:

- β-reduction in Λ_a:

$$(\lambda x.(y)\, x\, x)\; t \rightarrow_\beta (y)\; t\; t$$

$$(\lambda x.(y)\, x_1\, x_2[x_1, x_2 \leftarrow x])\; t \rightarrow_\beta (y)\, x_1\, x_2[x_1, x_2 \leftarrow t]$$

- μ-reduction with explicit sharings:

$$(\mu\alpha.(\mu\beta.(x)\alpha)\alpha)t \rightarrow_\mu \mu\alpha.(\mu\beta.(x)\; t\; \alpha)t\; \alpha$$

$$(\mu\alpha.(\mu\beta.(x)\alpha_1)\alpha_2[\alpha_1, \alpha_2 \leftarrow \alpha])t \rightarrow_\mu ????$$

  - $[\alpha_1, \alpha_2 \leftarrow (t)\; \alpha]$ would not work

Syntax with streams

- Left associativity of application: no direct way to share names
- Three kinds of expressions: terms, names, μ-variables
- Solution: introduce streams [Saurin-Gaboardi], get rid of names

- The $\Lambda\mu S$-calculus:

$$\begin{aligned} \mathit{Streams}\ S, T &::= \alpha \mid t \circ S \\ \mathit{Terms}\ t, u &::= x \mid \lambda x.t \mid (t)u \mid (t)S \mid \mu\alpha.t \end{aligned}$$

$$(\mu\alpha.t)u \longrightarrow_{\mu} \mu\alpha.\ t\{u \circ \alpha/\alpha\}$$

Syntax with streams

- The atomic $\lambda\mu$-calculus $\Lambda\mu S_a$:

$$\begin{aligned}
&\textit{Closures } [\phi], [\psi] ::= \ [\vec{x_q} \leftarrow t] \mid [\vec{\gamma_q} \leftarrow S] \mid [\vec{x_q} \twoheadleftarrow \lambda y.t^q] \mid [\vec{x_q} \twoheadleftarrow \mu\beta.t^q] \\
&\textit{Streams } S, T ::= \ \alpha \mid t \circ S \mid S[\phi] \\
&\textit{Terms } t, u ::= \ x \mid \lambda x.t \mid (t)u \mid (t)S \mid \mu\alpha.t \mid u[\phi] \\
&\lambda\textit{-tuples } t^p ::= \langle\, t_1, \ldots, t_p \,\rangle \mid t^p[\phi]
\end{aligned}$$

Reduction rules of $\Lambda\mu S_a$

- β-reductions
- μ-reductions
- Sharing reductions $\rightsquigarrow_s$:
  - Moving closures, compounding sharings
  - Duplication rules: allows duplications of (smaller) subterms

Type system for Λ_a

$$\begin{array}{c} A \\ \Downarrow \\ C \end{array} ::= A \mid \begin{array}{c} A \\ \Downarrow \\ C \end{array} \wedge \begin{array}{c} B \\ \Downarrow \\ D \end{array} \mid \begin{array}{c} A \\ \Downarrow \\ C \end{array} \to \begin{array}{c} B \\ \Downarrow \\ D \end{array} \mid \begin{array}{c} A \\ \Downarrow \\ \dfrac{B}{B'}\,\mathsf{r} \\ \Downarrow \\ C \end{array}$$

Inference rules:

$$\frac{A}{A \wedge \cdots \wedge A}\,\Delta \qquad \frac{B}{A \to (A \wedge B)}\,\lambda \qquad \frac{A \wedge (A \to B)}{B}\,\mathsf{@}$$

- Medial rule:

$$\frac{(A \vee B) \to (C \wedge D)}{(A \to C) \wedge (B \to D)}\,\mathbf{m}$$

Type system for $\Lambda\mu S_a$: first approach

- Classical Natural Deduction:

$$\frac{\Gamma \vdash C; \Delta}{\Gamma' \vdash C'; \Delta'}\; r$$

- Multiple conclusions/disjunctions: needs switch rules

$$\frac{(A; \Delta) \wedge (B; \Delta')}{(A \wedge B); \Delta \vee \Delta'}\; \mathsf{s_1} \qquad \frac{A \rightarrow (B; \Delta)}{(A \rightarrow B); \Delta}\; \mathsf{s_2}$$

Type system for $\Lambda\mu S_a$: first approach

Example (Double negation elimination):

$$\dfrac{y : \neg\neg A \vdash \neg\neg A \qquad \dfrac{\dfrac{\dfrac{\dfrac{x : A \vdash A}{(x)\alpha : A \vdash \bot; A}\,@_n}{(x)\alpha : A \vdash \bot; A, \bot^\delta}\,w}{\mu\delta.(x)\alpha : A \vdash \bot; A}\,\mu}{\lambda x.\mu\delta.(x)\alpha :\vdash \neg A; A}\,\lambda}{\dfrac{\dfrac{\dfrac{(y)\,\lambda x.\mu\delta.(x)\alpha : \neg\neg A \vdash \bot; A}{((y)\,\lambda x.\mu\delta.(x)\alpha)\phi : \neg\neg A \vdash \bot; A, \bot}\,@_n}{\mu\alpha.((y)\,\lambda x.\mu\delta.(x)\alpha)\phi : \neg\neg A \vdash A; \bot}\,\mu}{\lambda y.\mu\alpha.((y)\,\lambda x.\mu\delta.(x)\alpha)\phi :\vdash \neg\neg A \to A; \bot}\,\lambda}\,@$$

$$\dfrac{\dfrac{\top}{\neg\neg A^y \to \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg A; A^\alpha}{\neg\neg A^y \wedge A^x \to \dfrac{\dfrac{\dfrac{A^x \wedge (\neg A; A^\alpha)}{\dfrac{(A \wedge \neg A)}{\bot}\,@_{\mathbf{n}}; A^\alpha}\,\mathbf{s_1}}{\bot; A^\alpha \vee \bot^\delta}\,\mathbf{w}}{\bot^\delta; A^\alpha}\,\mu}\,\lambda}{A \to \bot; A^\alpha}\,\mathbf{s_2}}{\neg\neg A^y \wedge (A \to \bot; A^\alpha)}}{\dfrac{(\neg\neg A^y \wedge \neg A)}{\bot}\,@_{\mathbf{n}}; A^\alpha}\,\mathbf{s_1}}{\bot; A^\alpha \vee \bot^\phi}\,\mathbf{w}}{\bot; A^\alpha \vee \bot}\,@_{\mathbf{n}}}{A^\alpha; \bot}\,\mu}\,\lambda}{(\neg\neg A^y \to A); \bot}\,\mathbf{s_2}$$

Type system for $\Lambda\mu S_a$

- Multiple conclusions/disjunctions: needs switch rules
- Multiple conclusions come from μ-variables or body of μ-abstractions (of type $\perp$)

If

$$t \equiv \begin{array}{c} \Gamma \wedge A^x \\ {\scriptstyle \mathbf{t}}\Big\Downarrow \\ C \end{array}$$

Then

$$\lambda x.t \equiv A^x \to \begin{array}{c} \dfrac{\Gamma}{\Gamma \wedge A^x}\,\lambda \\ {\scriptstyle \mathbf{t}}\Big\Downarrow \\ C \end{array}$$

and A^x remains untouched

If

$$t \equiv \begin{array}{c} \Gamma \wedge (\neg A)^\alpha \\ {\scriptstyle \mathbf{t}}\Big\Downarrow \\ \perp \end{array}$$

We could similarly write:

$$\mu\alpha.t \equiv \dfrac{\begin{array}{c} \dfrac{\Gamma}{\Gamma \wedge \neg A}\,\mu \\ {\scriptstyle \mathbf{t}}\Big\Downarrow \quad \vee\ A^\alpha \\ \perp \end{array}}{A}$$

and A^α remains untouched

Type system

$$x \equiv A^x \qquad \alpha \equiv \neg A \qquad \lambda x.t \equiv A^x \rightarrow \begin{array}{c} \dfrac{\Gamma}{\Gamma \wedge A^x}\,\lambda \\ \mathbf{t}\Big\Downarrow \\ C \end{array} \qquad \mu\alpha.t \equiv \begin{array}{c} \dfrac{\Gamma}{\Gamma \wedge \neg A}\,\mu \\ \mathbf{t}\Big\Downarrow \quad \vee A \\ \bot \\ \hline\hline A \end{array}$$

$$(t)u \equiv \dfrac{\begin{array}{ccc} \Gamma_t & & \Gamma_u \\ \mathbf{t}\Big\Downarrow & \wedge & \Big\Downarrow\mathbf{u} \\ A \rightarrow B & & A \end{array}}{B}\,@ \qquad (t)S \equiv \dfrac{\begin{array}{ccc} \Gamma_t & & \Delta_S \\ \mathbf{t}\Big\Downarrow & \wedge & \Big\Downarrow\mathbf{S} \\ A & & \neg A \end{array}}{\bot}\,@_{\mathbf{n}} \qquad t \circ S \equiv \dfrac{\begin{array}{ccc} \Gamma_t & & \Delta_S \\ \mathbf{t}\Big\Downarrow & \wedge & \Big\Downarrow\mathbf{S} \\ B & & \neg A \end{array}}{\neg(B \rightarrow A)}\,\circ$$

$$\langle t_1, \ldots, t_p \rangle \equiv \begin{array}{c} \Gamma_1 \\ \mathbf{t_1}\Big\Downarrow \\ B_1 \end{array} \wedge \cdots \wedge \begin{array}{c} \Gamma_n \\ \Big\Downarrow\mathbf{t_n} \\ B_n \end{array}$$

$$u^*[\vec{x_n} \leftarrow t] \equiv \begin{array}{c} \begin{array}{c} \Gamma \\ \mathbf{t}\Big\Downarrow \\ A \end{array} \quad \wedge\, \Sigma_{u^*} \\ \hline A \wedge \cdots \wedge A \;\; \Delta \\ \hline\hline A \wedge \cdots \wedge A \wedge \Sigma_{u^*} \\ \mathbf{u}^*\Big\Downarrow \\ C \end{array} \qquad u^*[\vec{\gamma_n} \leftarrow S] \equiv \begin{array}{c} \begin{array}{c} \Delta \\ \mathbf{S}\Big\Downarrow \\ \neg A \end{array} \quad \wedge\, \Sigma_{u^*} \\ \hline \neg A \wedge \cdots \wedge \neg A \;\; \Delta \\ \hline\hline \neg A \wedge \cdots \wedge \neg A \wedge \Sigma_{u^*} \\ \mathbf{u}^*\Big\Downarrow \\ C \end{array}$$

$$u^*[\vec{x_n} \twoheadleftarrow \lambda y.t^n] \equiv \begin{array}{c} A^y \rightarrow \begin{array}{c} \dfrac{\Gamma}{\Gamma \wedge A^y}\,\lambda \\ \mathbf{t^P}\Big\Downarrow \\ B_1 \wedge \cdots \wedge B_n \end{array} \quad \wedge\, \Sigma_u^* \\ \hline (A \rightarrow B_1) \wedge \cdots \wedge (A \rightarrow B_n) \;\; \mathbf{d}_\lambda \\ \hline\hline (A \rightarrow B_1) \wedge \cdots \wedge (A \rightarrow B_n) \wedge \Sigma_u^* \\ \mathbf{u}^*\Big\Downarrow \\ C \end{array}$$

$$u^*[\vec{x_n} \twoheadleftarrow \mu\alpha.t^n] \equiv \begin{array}{c} \begin{array}{c} \dfrac{\Gamma}{\Gamma \wedge \neg A}\,\mu \\ \mathbf{t^P}\Big\Downarrow \quad \vee A \\ \bot \wedge \cdots \wedge \bot \end{array} \quad \wedge\, \Sigma_u^* \\ \hline A \wedge \cdots \wedge A \;\; \mathbf{d}_\mu \\ \hline\hline A \wedge \cdots \wedge A \wedge \Sigma_u^* \\ \mathbf{u}^*\Big\Downarrow \\ C \end{array}$$

Example: Double negation elimination

$$
\dfrac{y : \neg\neg A \vdash \neg\neg A \qquad \dfrac{\dfrac{\dfrac{\dfrac{x : A \vdash A}{(x)\alpha : A \vdash \bot; A}\,@_n}{(x)\alpha : A \vdash \bot; A, \bot^{\delta}}\,w}{\mu\delta.(x)\alpha : A \vdash \bot; A}\,\mu}{\lambda x.\mu\delta.(x)\alpha : \vdash \neg A; A}\,\lambda}{\dfrac{\dfrac{\dfrac{(y)\ \lambda x.\mu\delta.(x)\alpha : \neg\neg A \vdash \bot; A}{((y)\ \lambda x.\mu\delta.(x)\alpha)\phi : \neg\neg A \vdash \bot; A, \bot}\,@_n}{\mu\alpha.((y)\ \lambda x.\mu\delta.(x)\alpha)\phi : \neg\neg A \vdash A; \bot}\,\mu}{\lambda y.\mu\alpha.((y)\ \lambda x.\mu\delta.(x)\alpha)\phi : \vdash \neg\neg A \to A; \bot}\,\lambda}\,@
$$

$$
\begin{array}{lc}
 & \top \\ \hline
 & \neg\neg A^{y} \quad (\lambda) \\ \hline
 & \neg A^{\alpha} \quad (\mu) \\ \hline
 & A^{x} \wedge \neg A^{\alpha} \quad (\lambda) \\ \hline
\neg\neg A^{y} \to \quad \neg\neg A^{y} \wedge \quad A^{x} \to & \dfrac{(A^{x} \wedge \neg A^{\alpha})}{\bot}\,@_{\mathbf{n}} \vee \bot^{\delta} \quad (\mu) \qquad \vee\ A^{\alpha} \\ \hline\hline
 & \bot^{\delta} \\ \hline\hline
 & \dfrac{\neg\neg A^{y} \wedge \neg A}{\bot}\,@_{\mathbf{n}} \\ \hline\hline
 & A^{\alpha}
\end{array}
$$

Preservation of strong normalization

For any term $T \in \Lambda\mu S$, if T is strongly normalizing then its translation $\llparenthesis\, T \,\rrparenthesis \in \Lambda\mu S_a$ is strongly normalizing.

We could show that one step of reduction in $\Lambda\mu S_a$ leads to at least one step in $\Lambda\mu S$:

$$
\begin{tikzcd}
\llparenthesis\, T \,\rrparenthesis \in \Lambda\mu S_a \arrow[d, "1"] & \rightsquigarrow & \llbracket\, \llparenthesis\, T \,\rrparenthesis \,\rrbracket = T \in \Lambda\mu S \arrow[d, "+"] \\
t' & & T' = \llbracket\, t' \,\rrbracket
\end{tikzcd}
$$

Preservation of strong normalization

Problem 1:

- Reductions in $\Lambda\mu S_a$ can be β, μ-reductions, or $\rightsquigarrow_s$
- For $t, u \in \Lambda\mu S_a$, if $t \rightsquigarrow_s u$ then $[\![\, t \,]\!] = [\![\, u \,]\!]$
- One step of $\rightsquigarrow_s$ in $\Lambda\mu S_a$ corresponds to zero steps in $\Lambda\mu S_a$

$$\begin{tikzcd}
(\!|\, T \,|\!) \in \Lambda\mu S_a \arrow[d, "1"] & & [\![\, (\!|\, T \,|\!) \,]\!] = T \in \Lambda\mu S \arrow[d, "*"] \\
t' \arrow[r, phantom, "\rightsquigarrow"] & & T' = [\![\, t' \,]\!]
\end{tikzcd}$$

Preservation of strong normalization

Solution: $\rightsquigarrow_s$ is strongly normalizing.

- If $(\!|\ T\ |\!)$ has an infinite reduction path, it will look like:

$$(\!|\ T\ |\!) \rightsquigarrow_s^* t_1 \rightarrow_{\beta,\mu}^+ t_2 \rightsquigarrow_s^* \dots \rightsquigarrow_s^* t_n \rightarrow_{\beta,\mu}^\infty \dots$$

- We can concentrate on β and μ-reductions

Preservation of strong normalization

Problem 2: infinite reductions can occur inside weakenings.

- Example: $T = (\mu\alpha.(x)\,\beta)\,\Omega$
- $(\!| T |\!) = (\mu\alpha.(x)\,\beta[\leftarrow \alpha])\,\Omega \longrightarrow_\mu t' = \mu\alpha.(x)\,\beta[\leftarrow \Omega \circ \alpha]$
- But $[\![t']\!] = \mu\alpha.(x)\,\beta$ is in normal form
- There exists an infinite reduction path from t', but not from $[\![t']\!]$

Preservation of strong normalization

An auxiliary calculus is introduced, the weakening calculus $\Lambda\mu S_w$, an extension of $\Lambda\mu S$ with explicit weakenings.

$$\begin{tikzcd}
\Lambda\mu S_a \arrow[r, "\mathsf{dupl}"] \arrow[d, "\mathsf{del},\ \mathsf{dupl}"', "s"] & \Lambda\mu S_w \arrow[r, "\mathsf{del}"] \arrow[d, "w"] & \Lambda\mu S \arrow[dl, leftrightarrow, "\mathsf{forget}"] \\
\Lambda\mu S_a\ SN \arrow[r, "\cong"] & \Lambda\mu S + \text{ wk var} &
\end{tikzcd}$$

- Find a strategy keeping infinite reductions outside of weakenings.

Other properties

- Fully-lazy sharing
- Typed calculus: strong normalization, subject reduction