Toward dependent choice: a classical sequent calculus with dependent types

Hugo Herbelin[1], Étienne Miquey[1,2]

[1]Team πr^2 (INRIA), PPS, Université Paris-Diderot
[2]Fac. de Ingeniería, Universidad de la República, Uruguay

TYPES 2015


Curry-Howard correspondence

Proof/program correspondence

Proof theory	**Functional programming**
Proposition	Type
Deduction rule	Typing rule
$A \Rightarrow B$	$A \to B$
$\dfrac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}$	$\dfrac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash (t)u : B}$

A first extension :

Classical logic [Griffin'90]

$$\texttt{call/cc} : ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$$

Extending more...

Axiom of Choice

$$AC_A : \forall x^A \exists y^B P(x,y) \to \exists f^{A \to B} \forall x^A P(x, f(x))$$

Ingredients : Martin-Löf's type theory

$$\frac{\Gamma, a : A \vdash p : B}{\Gamma \vdash \lambda a.p : A \to B} \to_I \qquad \frac{\Gamma, x : T \vdash p : A}{\Gamma \vdash \lambda x.p : \forall x^T A} \forall_I$$

$$\frac{\Gamma \vdash p : A[t/x] \quad \Gamma \vdash t : T}{\Gamma \vdash (t,p) : \exists x^T A} \exists_I \qquad \frac{\Gamma \vdash p : \exists x^T A(x)}{\Gamma \vdash \mathtt{prf}\, p : A(\mathtt{wit}\, p)} \exists_E$$

$$\begin{array}{rcl} AC_A & := & \lambda H.(\lambda x.\,\mathtt{wit}(Hx), \lambda x.\,\mathtt{prf}(Hx)) \\ & : & \forall x^A \exists y^B P(x,y) \to \exists f^{A \to B} \forall x^A P(x, f(x)) \end{array}$$

... still more ?

$$AC_A := \lambda H.(\lambda x.\, \mathtt{wit}(Hx), \lambda x.\, \mathtt{prf}(Hx))$$

- In classical logic :

$$H_0 := \mathtt{callcc}_\alpha(1, \phi(\mathtt{throw}_\alpha(2, p)) : \exists x P(x)$$

- the witness may **depend on the context, so do the proof !**
- reducing a term in different contexts, one might reach a contradiction :

$$\underbrace{(\mathtt{wit}(AC_A\ H)\ 0, \mathtt{prf}(AC_A\ H)\ 0)}_{\exists y P(0,y)} \rightsquigarrow (\mathtt{wit}\, H_0, \mathtt{prf}\, H_0) \rightsquigarrow \underbrace{(1, \overbrace{p}^{P(0,2)})}_{\cancel{\exists y P(x,y)}}$$

↬ idea : we need to *share*...

Toward a solution ?

- Restriction to countable choice :

$$AC_{\mathbb{N}} : \forall x^{\mathbb{N}} \exists y^{B} P(x, y) \to \exists f^{\mathbb{N} \to B} \forall x^{\mathbb{N}} P(x, f(x))$$

- Proof :

$$AC := \lambda H.(\lambda n.\, \mathtt{wit}(Hn), \lambda n.\, \mathtt{prf}(Hn))$$

Toward a solution ?

- Restriction to countable choice :

$$AC_{\mathbb{N}} : \forall x^{\mathbb{N}} \exists y^{B} P(x, y) \to \exists f^{\mathbb{N} \to B} \forall x^{\mathbb{N}} P(x, f(x))$$

- Proof :

$$\begin{aligned} AC := \lambda H.(\lambda n.&\texttt{if } n = 0 \texttt{ then wit } H\ 0 \texttt{ else} \\ &\texttt{if } n = 1 \texttt{ then wit } H\ 1 \texttt{ else } \dots\ , \\ \lambda n.&\texttt{if } n = 0 \texttt{ then prf } H\ 0 \texttt{ else} \\ &\texttt{if } n = 1 \texttt{ then prf } H\ 1 \texttt{ else } \dots\ ) \end{aligned}$$

Toward a solution ?

- Restriction to countable choice :

$$AC_{\mathbb{N}} : \forall x^{\mathbb{N}} \exists y^{B} P(x, y) \to \exists f^{\mathbb{N} \to B} \forall x^{\mathbb{N}} P(x, f(x))$$

- Proof :

$$\begin{aligned} AC := \lambda H.\, & \mathtt{let}\, H_0 = H\ 0\, \mathtt{in} \\ & \mathtt{let}\, H_1 = H\ 1\, \mathtt{in} \\ & \ldots \\ & (\lambda n.\mathtt{if}\ n = 0\ \mathtt{then}\ \mathtt{wit}\, H_0\ \mathtt{else} \\ & \quad \mathtt{if}\ n = 1\ \mathtt{then}\ \mathtt{wit}\, H_1\ \mathtt{else}\ \ldots\ , \\ & \lambda n.\mathtt{if}\ n = 0\ \mathtt{then}\ \mathtt{prf}\, H_0\ \mathtt{else} \\ & \quad \mathtt{if}\ n = 1\ \mathtt{then}\ \mathtt{prf}\, H_1\ \mathtt{else}\ \ldots\ ) \end{aligned}$$

Toward a solution ?

- Restriction to countable choice :

$$AC_{\mathbb{N}} : \forall x^{\mathbb{N}} \exists y^{B} P(x, y) \to \exists f^{\mathbb{N} \to B} \forall x^{\mathbb{N}} P(x, f(x))$$

- Proof :

$$AC := \lambda H.\, \mathtt{let}\, H_\infty = (H\ 0, H\ 1, \ldots, H\ n, \ldots)\, \mathtt{in}$$
$$(\lambda n.\mathtt{nth}\ n\ H_\infty, \lambda n.\mathtt{nth}\ n\ H_\infty)$$

Toward a solution ?

- Restriction to countable choice :

$$AC_{\mathbb{N}} : \forall x^{\mathbb{N}} \exists y^{B} P(x, y) \to \exists f^{\mathbb{N} \to B} \forall x^{\mathbb{N}} P(x, f(x))$$

- Proof :

$$AC := \lambda H.\, \mathtt{let}\, H_\infty = \mathtt{cofix}^0_{fn}(H\ n, f(S(n)))\, \mathtt{in}$$
$$(\lambda n.\mathtt{nth}\ n\ H_\infty, \lambda n.\mathtt{nth}\ n\ H_\infty)$$

- We can easily adapt to dependent choice
- Intuitionistic proof !

The recipe : dPA^ω

Ref : Hugo Herbelin (LICS'12)

A Constructive Proof of Dependent Choice, Compatible with Classical Logic

A formal system :

- **classical** : $p, q ::= \ldots \mid \mathtt{catch}_\alpha\, p \mid \mathtt{throw}_\alpha\, p$
- with **dependent types** :
  - formulas : $A, B ::= \ldots \mid [a : A] \to B \mid t = u$,
  - terms : $t, u ::= \ldots \mid \mathtt{wit}\, p$,
  - proofs : $p, q ::= \ldots \mid \mathtt{prf}\, p$
- a **compartimentalization** :

Definition : *Negative-elimination-free*

Values : $V ::= a \mid () \mid \lambda x.p \mid \lambda a.p \mid (t, p) \mid (V, V) \mid \iota_i(V)$

$N, M ::= \ldots \mid \lambda a.p \mid (N, N) \mid \iota_i(N) \mid \mathtt{prf}\, N \mid \mathtt{let}\, a = M\, \mathtt{in}\, N \mid \ldots$

$$\frac{\Gamma \vdash p : \exists x^T A(x) \quad p \text{ Nef}}{\Gamma \vdash \mathtt{prf}\, p : A(\mathtt{wit}\, p)} \qquad \frac{\Gamma \vdash p : \exists x^T A(x) \quad \Gamma, x : T, a : A \vdash q : B}{\Gamma \vdash \mathtt{dest}\, p\, \mathtt{as}((x, a))\, \mathtt{in}\, q : B}$$

The recipe : dPA^ω

Ref : Hugo Herbelin (LICS'12)

A Constructive Proof of Dependent Choice, Compatible with Classical Logic

A formal system :

- **classical** : $p, q ::= \ldots \mid \mathtt{catch}_\alpha\, p \mid \mathtt{throw}_\alpha\, p$
- with **dependent types** :
  - formulas : $A, B ::= \ldots \mid [a : A] \to B \mid t = u$,
  - terms : $t, u ::= \ldots \mid \mathtt{wit}\, p$,
  - proofs : $p, q ::= \ldots \mid \mathtt{prf}\, p$
- a **compartimentalization** ,
- call-by-value and **sharing** : $p, q ::= \ldots \mid \mathtt{let}\, a = q\, \mathtt{in}\, p$
- with inductive and **coinductive** constructions :
  $p, q ::= \ldots \mid \mathtt{ind}\, t\, \mathtt{of}\, [p|(x, a).q] \mid \mathtt{cofix}^t_{bn}\, p$
- **laziness** on the `cofix`

Main properties

Subject reduction

If $\Gamma \vdash p : A$ and $p \rightsquigarrow q$, then $\Gamma \vdash q : A$.

Normalisation

If $\Gamma \vdash p : A$ then p is normalisable.

Conservativity

If A is $\rightarrow$-ν-$\mathtt{wit}$-$\forall$-free, and $\vdash_{dPA^\omega} p : A$, then $\vdash_{HA^\omega} p : A$

Consistency

$$\nvdash_{dPA^\omega} \bot$$

Normalisation by CPS

- A translation $[\![\cdot]\!] : dPA^{\omega} \to \lambda^{\text{something}}$ s.t. :

$$(p \rightsquigarrow q) \qquad \Rightarrow \qquad ([\![p]\!] \to^{+} [\![q]\!])$$

- Interest : highlights on negative translation and side effects
- If we are lucky : *update-induction* [Berger'04]
- Difficulties :
  - (Continuation+State)-passing-style
  - Typing ?
  - Laziness & `cofix` ?
  - Manipulation of contexts :

$$\texttt{let}\, a = \underset{bx}{\overset{t}{\texttt{cofix}}}\, p = D[a]\, \texttt{in} \quad \triangleright \quad \ldots$$

CPS : Ariola et al (FLOPS'12)

Classical Call-by-Need Sequent Calculi : The Unity of Semantic Artifacts

Sequent calculus : $\overline{\lambda}\mu\tilde{\mu}$

Ref : Curien/Herbelin (ICFP'00)

The duality of computation,

Syntax :

$$\begin{array}{lccl} \text{(Commands)} & c & ::= & \langle p \mid e \rangle \\ \text{(Proofs)} & p, q & ::= & a \mid \lambda a.p \mid \mu\alpha.c \\ \text{(Contexts)} & e & ::= & \alpha \mid \tilde{\mu}a.c \mid p \cdot e \end{array}$$

Call-by-value reduction :

$$\begin{array}{rcl} \langle \lambda a.p \mid q \cdot e \rangle & \rightsquigarrow & \langle q \mid \tilde{\mu}a.\langle p \mid e \rangle \rangle \\ \langle V \mid \tilde{\mu}a.c \rangle & \rightsquigarrow & c[V/a] \\ \langle \mu\alpha.c \mid e \rangle & \rightsquigarrow & c[e/\alpha] \end{array}$$

Typing rules :

$$\frac{\Gamma \vdash p : A \quad \Gamma \vdash e : A^{\perp\!\!\!\perp}}{\langle p \mid e \rangle : \Gamma}\ \text{CUT} \qquad \frac{c : (\Gamma, \alpha : A^{\perp\!\!\!\perp})}{\Gamma \vdash \mu\alpha.c : A}\ \mu \qquad \frac{c : (\Gamma, a : A)}{\Gamma \vdash \tilde{\mu}a.c : A^{\perp\!\!\!\perp}}\ \tilde{\mu}$$

Adding sharing

Ref : Ariola et al (FLOPS'12)

Classical Call-by-Need Sequent Calculi : The Unity of Semantic Artifacts

Syntax :

$$\begin{array}{lrcl} \text{(Commands)} & c & ::= & \langle p \mid e \rangle \tau \\ \text{(Proofs)} & p, q & ::= & a \mid \lambda a.p \mid \mu\alpha.c \\ \text{(Contexts)} & e & ::= & \alpha \mid \tilde{\mu} a.c \mid p \cdot e \\ \text{(Environments)} & \tau & ::= & \cdot \mid [a = p]\tau \end{array}$$

Call-by-value reduction :

$$\begin{array}{rcl} \langle \lambda a.p \mid q \cdot e \rangle \tau & \rightsquigarrow & \langle q \mid \tilde{\mu} a.\langle p \mid e \rangle \rangle \tau \\ \langle \mu\alpha.c \mid e \rangle \tau & \rightsquigarrow & c[e/\alpha]\tau \\ \langle V \mid \tilde{\mu} a.c \rangle \tau & \rightsquigarrow & c[a = V]\tau \\ \langle a \mid F \rangle \tau'[a = p]\tau & \rightsquigarrow & \langle p \mid F \rangle \tau'[a = p]\tau \end{array}$$

Typing rules :

$$\ldots \quad + \quad \langle p \mid e \rangle \tau : (\Gamma) ::= \langle \tau(p) \mid \tau(e) \rangle : (\Gamma)$$

Adding cofix

Syntax :

$$
\begin{array}{lrcl}
\text{(Commands)} & c & ::= & \langle p \mid e \rangle \tau \\
\text{(Proofs)} & p, q & ::= & a \mid \lambda a.p \mid \mu\alpha.c \mid \mathtt{cofix}^t_{bx}\, p \\
\text{(Contexts)} & e & ::= & \alpha \mid \tilde{\mu} a.c \mid p \cdot e \\
\text{(Environments)} & \tau & ::= & \cdot \mid [a = p]\tau
\end{array}
$$

Call-by-value reduction :

$$
\begin{array}{rcl}
 & \vdots & \\
\langle \mathtt{cofix}^t_{bx}\, p \mid \tilde{\mu} a.c\tau' \rangle \tau & \rightsquigarrow & c\tau'[a = \mathtt{cofix}^t_{bx}\, p]\tau \\
\langle a \mid F \rangle \tau'[a = \mathtt{cofix}^t_{bx}\, p]\tau & \rightsquigarrow & \langle p[t/x][\lambda y.\, \mathtt{cofix}^y_{bx}\, p/b] \mid \tilde{\mu} a.\langle a \mid F \rangle \tau' \rangle \tau
\end{array}
$$

Typing rules :

$$
+ \quad \frac{\Gamma \vdash t : T \quad \Gamma, f : T \to \mathbb{N}, x : T, b : \forall y f(y) = 0 \vdash p : A \quad f \text{ positive in } A}{\Gamma \vdash \mathtt{cofix}^t_{bx}\, p : \nu^t_{fx} A}
$$

Adding dependent types

Syntax :

$$
\begin{array}{lrcl}
\text{(Commands)} & c & ::= & \langle p \mid e \rangle \tau \\
\textbf{(Terms)} & t, u & ::= & x \mid 0 \mid S(t) \mid \lambda x.t \mid t\ u \mid \mathtt{wit}\ p \\
\text{(Proofs)} & p, q & ::= & \ldots \mid \mathtt{cofix}^{t}_{bx}\ p \mid \lambda x.p \mid (t, p) \mid \mathtt{prf}\ p \\
\text{(Contexts)} & e & ::= & \alpha \mid \tilde{\mu} a.c \mid p \cdot e \mid t \cdot e \\
\text{(Environments)} & \tau & ::= & \cdot \mid [a = p]\tau
\end{array}
$$

Call-by-value reduction :

$$
\begin{array}{rcl}
 & \vdots & \\
\langle \mathtt{prf}\ p \mid e \rangle \tau & \rightsquigarrow & \langle p \mid \tilde{\mu} a.\langle \mathtt{prf}\ a \mid e \rangle \rangle \tau \\
\langle \mathtt{prf}\ a \mid e \rangle \tau'[a = (t, p)]\tau & \rightsquigarrow & \langle p \mid e \rangle \tau'[a = (t, p)]\tau
\end{array}
$$

Typing rules :

$$
\frac{\Gamma \vdash p : A[t/x] \quad \Gamma \vdash t : T}{\Gamma \vdash (t, p) : \exists x^{T} A}\ \exists_I
\qquad
\frac{\Gamma \vdash p : \exists x^{T} A(x) \quad p\ \mathrm{Nef}}{\Gamma \vdash \mathtt{prf}\ p : A(\mathtt{wit}\ p)}\ \mathtt{prf}
$$

Last but not least

Syntax :

$$
\begin{array}{llll}
\text{(Commands)} & c & ::= & \langle p \mid e \rangle \tau \\
\text{(Terms)} & t, u & ::= & \ldots \\
\text{(Proofs)} & p, q & ::= & \ldots \mid (a_1, a_2) \mid \iota_i(p) \mid \mathtt{subst}\ p\ q \mid \mathtt{refl} \\
\text{(Contexts)} & e & ::= & \ldots \mid \tilde{\mu}(a_1, a_2).c \mid \tilde{\mu}[a_1.c_1 | a_2.c_2] \\
\text{(Environments)} & \tau & ::= & \cdot \mid [a = p]\tau
\end{array}
$$

Call-by-value reduction :

$$
\begin{array}{c}
\vdots \\
\langle (p_1, p_2) \mid \tilde{\mu} a.c \rangle \tau \rightsquigarrow \langle p_1 \mid \tilde{\mu} a_1.\langle p_2 \mid \tilde{\mu} a_2.c[a = (a_1, a_2)] \rangle \rangle \tau
\end{array}
$$

Typing rules :

$$
\frac{\Gamma \vdash p : t = u \quad \Gamma \vdash q : B[t/x]}{\Gamma \vdash \mathtt{subst}\ p\ q : B[u/x]}\ \mathtt{subst} \qquad \frac{\Gamma \vdash t : T}{\Gamma \vdash \mathtt{refl} : t = t}\ \mathtt{refl}
$$

Does this work ?

$$\dfrac{\dfrac{\Gamma, a : A \vdash p : B[a]}{\Gamma \vdash \lambda a.p : [a : A] \to B} \quad \dfrac{\Gamma \vdash q : A \quad \Gamma \vdash e : B[q]^{\perp\!\!\!\perp} \quad q \text{ Nef}}{\Gamma \vdash q \cdot e : ([a : A] \to B)^{\perp\!\!\!\perp}}}{\langle \lambda a.p \mid q \cdot e \rangle : \Gamma}\ \text{CUT}$$

$$\rightsquigarrow$$

$$\dfrac{\Gamma \vdash q : A \qquad \dfrac{\dfrac{\Gamma, a : A \vdash p : B[a] \quad \Gamma, a : A \vdash e : B[q]^{\perp\!\!\!\perp}; \{a|q\}}{\langle p \mid e \rangle : \Gamma, a : A; \{a|q\}}\ \text{CUT}}{\Gamma \vdash \tilde{\mu}a.\langle p \mid e \rangle : A^{\perp\!\!\!\perp}; \{.|q\}}\ \tilde{\mu}}{\langle q \mid \tilde{\mu}a.\langle p \mid e \rangle \rangle : \Gamma; \{\cdot|\cdot\}}\ \text{CUT}$$

A type system with dependencies list

$$\frac{c : (\Gamma, \alpha : A^{\perp\!\!\!\perp})}{\Gamma \vdash \mu\alpha.c : A}\ \mu \qquad \frac{c : (\Gamma, a : A; \{a|p\}\varepsilon)}{\Gamma \vdash \tilde{\mu}a.c : A^{\perp\!\!\!\perp}; \{\cdot|p\}\varepsilon}\ \tilde{\mu}$$

$$\frac{\Gamma \vdash p_1 : A \quad \Gamma \vdash p_2 : B}{\Gamma \vdash (p_1, p_2) : A \wedge B}\ \wedge_I \qquad \frac{c : \Gamma, a_1 : A_1, a_2 : A_2; \{(a_1, a_2)|p\}\varepsilon}{\Gamma \vdash \tilde{\mu}(a_1, a_2).c : (A_1 \wedge A_2)^{\perp\!\!\!\perp}; \{\cdot|p\}\varepsilon}\ \wedge_E$$

$$\frac{\Gamma \vdash p : A_{\varepsilon^+} \quad \Gamma \vdash e : A^{\perp\!\!\!\perp}_{\varepsilon^-}; \{\cdot|p\}\varepsilon}{\langle p \mid e\rangle : (\Gamma; \varepsilon)}\ cut$$

where :

$$\langle p \mid e\rangle\tau : (\Gamma) ::= \langle \tau(p) \mid \tau(e)\rangle : (\Gamma)$$

$$A_{\{p|q\}\varepsilon^+} := A[p/\bullet]_{\varepsilon^+}, q \notin \text{Nef} \to \bullet, p \notin A$$

$$A_{\{p|q\}\varepsilon^-} := A[q/\bullet]_{\varepsilon^-}, q \notin \text{Nef} \to \bullet, p \notin A$$

Properties

Safe proof substitution

Assume $\Gamma, \Delta \vdash p : A$, p Nef then we have :

$$\Gamma, a : A, \Delta \vdash q : B \;\Rightarrow\; \Gamma, \Delta[p/a] \vdash q[p/a] : B[p/a]$$

(+ same property for contexts/commands/terms typing)

The same holds for terms and contexts subtitutions.

Subject reduction

If $c : \Gamma; \{\cdot | \cdot\}$ and $c \rightsquigarrow c'$, then $c : \Gamma; \{\cdot | \cdot\}$.

A translation from dPA^ω

$$
\begin{array}{rcl}
[\![\lambda a.p]\!] & := & \lambda a.[\![p]\!] \\
[\![(p)q]\!] & := & \mu\alpha.\langle [\![p]\!] \mid [\![q]\!] \cdot \alpha \rangle \\
[\![\mathtt{let}\, a = p\, \mathtt{in}\, q]\!] & := & \mu\alpha.\langle [\![p]\!] \mid \tilde{\mu} a.\langle [\![q]\!] \mid \alpha \rangle \rangle \\
[\![(p,q)]\!] & := & ([\![p]\!], [\![q]\!]) \\
[\![\mathtt{split}\, p\, \mathtt{as}\, (a_1, a_2)\, \mathtt{in}\, q]\!] & := & \mu\alpha.\langle [\![p]\!] \mid \tilde{\mu}(a_1, a_2).\langle [\![q]\!] \mid \alpha \rangle \rangle
\end{array}
$$

Typing preservation

If $\vdash_{dPA^\omega} p : A$, then $\vdash_{dLPA^\omega} p : A$

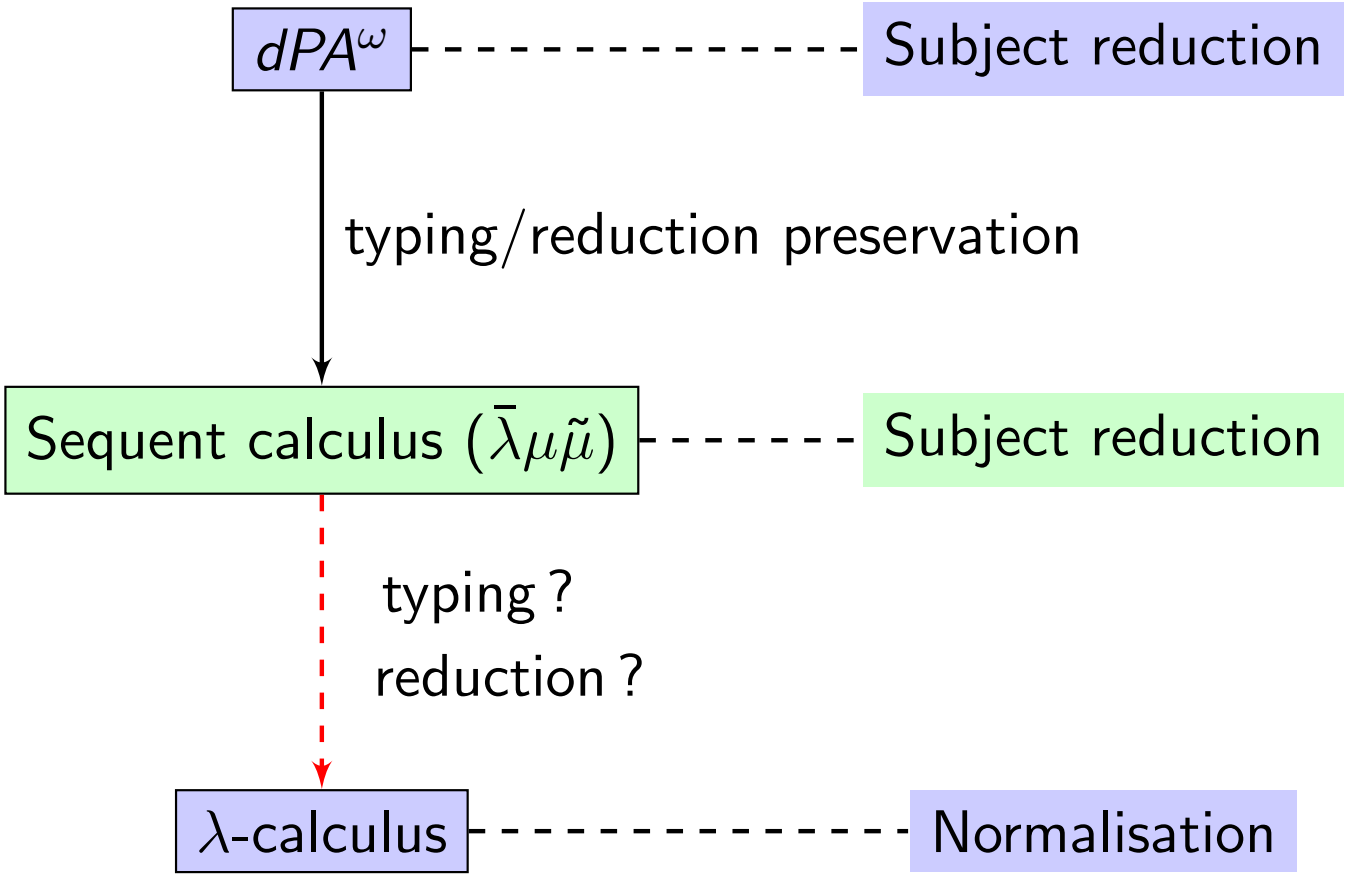
Remember :

$AC_{\mathbb{N}} := \lambda H.\, \mathtt{let}\, H_\infty = \mathtt{cofix}^0_{bn}(H\ n, b(S(n)))\, \mathtt{in}(\lambda n.\mathtt{nth}\ n\ H_\infty, \lambda n.\mathtt{nth}\ n\ H_\infty)$

Corollary

$\vdash [\![AC_{\mathbb{N}}]\!] : \forall x \exists y P(x,y) \to \exists f \forall x P(x, f(x))$

Where are we now ?



Thank you for your attention