Delimited Control and Continuation Passing Style in Pure Type Systems

Pierre Boutillier and Hugo Herbelin

Univ Paris Diderot, Paris Sorbonne Cité, PPS, UMR 7126, INRIA, F-75013 Paris, France

LI 2012, Week 3

Informal teaser

Difficulty

- With dependent types, there is computation in types.
- Contexts of computation are not the same at type level and at term level.
- What is the meaning of a term context brought by a term in a type context.

Previous Solution

- You can control classical computation by having a very explicit type system but this does not take into account dependencies. Danvy - Filinsky
- You can have a dependently typed setting and classical calculus in 2 different areas by putting restriction that forgive classical calculus to be computed in types. Barthe - Hartcliff - Sørensen

Danvy-Filinski Types

- With control operators, mixing call-by-value and call-by-name is no longer confluent. So in presence of effects, a typing system for them reflects the order of computation.
- We'll talk of

$$\Gamma \vdash u : S - A/B$$

to say that u of type S in a context answering type A will give back type B.

$$\frac{\Gamma \vdash t : \texttt{bool} - C/B \qquad \Gamma \vdash u : S - A/C \qquad \Gamma \vdash v : S - A/C}{\Gamma \vdash \mathit{if}\ t\ \mathit{then}\ u\ \mathit{else}\ v : S - A/B}\ \text{IF}$$

Danvy-Filinski Types

- With control operators, mixing call-by-value and call-by-name is no longer confluent. So in presence of effects, a typing system for them reflects the order of computation.
- We'll talk of

$$\Gamma \vdash u : S - A / B$$

to say that u of type S in a context answering type A will give back type B.

$$\text{IF}\ \frac{\Gamma \vdash t : \texttt{bool} - C / B \qquad \Gamma \vdash u : S - A / C \qquad \Gamma \vdash v : S - A / C}{\Gamma \vdash \mathit{if}\ t\ \mathit{then}\ u\ \mathit{else}\ v : S - A / B}$$

- It will become $(S \to A) \to B$ by CPS.

Barthe Hatcliff Sørensen

Setting

- A PTS with $\perp$ and μ as syntax primitives
- A special sort Prop for classical contents with restrictions over Ax and Rel.

Key point

- No classical computation is not guarded by a variable of predicate in a Type because Prop is a bottom sort.

This talk contribution

- Structure the syntax even if there is coercions between categories.
- Make coercion explicit and give it a meaning: at the time, you give an value in a term, you *drop* the computation context you had.
- This way, you have a formal notion of Call By Value and Call By Name dependently typed λ-calculus
- On which you can safely add control

Catch, Throw, Abort, Reset

How it works

$\lambda\mu\widehat{tp}$

catch$_\alpha u$	gives the name α to the present context and goes on computing u.	▶ $\mu\,\alpha.[\alpha]\,u$
throw$_\alpha u$	restores the context bound to α to compute u.	▶ $\mu\,\_.[\alpha]\,u$
# u	computes u in a "clean" context.	▶ $\mu\,\widehat{tp}.[\widehat{tp}]\,u$
abort V	drops the current context and answers V.	▶ $\mu\,\_.[\widehat{tp}]\,V$

$$\textbf{catch}_\alpha(inr\ \lambda a\colon A.\textbf{throw}_\alpha(inl\ a)) : A \vee \neg A$$

how it encodes callcc / S

$$callcc(\lambda k.u) \equiv \textbf{catch}_\alpha(u[\lambda x.\textbf{throw}_\alpha x/k])$$
$$S(\lambda k.M) \equiv \textbf{catch}_\alpha\textbf{abort}\ (M[\lambda x.\#\ \textbf{throw}_\alpha x/k])$$

A generic framework of typed λ-calculus

Setting

St	A set of **Sorts**
Ax	A set of pairs of sorts called **Axioms**
Rel	A set of triples os sorts called **Relations**

Grammar

$$u, v, S, T \quad := \quad x \mid s \mid \lambda x : S.u \mid \Pi x : S.T \mid u\ v$$

Example

$$\lambda f : (\Pi x : \texttt{Kind}.\Pi y : \texttt{Kind}.pair\ x\ y).f\ \texttt{Type}$$

A generic framework of typed λ-calculus

$$\textsc{Nil}\ \frac{}{\varepsilon_{wf}} \qquad \textsc{Cons}\ \frac{\Gamma_{wf} \qquad \Gamma \vdash S : s}{(\Gamma, x : S)_{wf}} \qquad \textsc{Var}\ \frac{x : S \in \Gamma \qquad \Gamma_{wf}}{\Gamma \vdash x : S} \qquad \textsc{Sort}\ \frac{(s_1, s_2) \in Ax \qquad \Gamma_{wf}}{\Gamma \vdash s_1 : s_2}$$

$$\textsc{Lam}\ \frac{\Gamma, x : S \vdash u : T \qquad \Gamma \vdash \Pi x : S.T : s}{\Gamma \vdash \lambda x : S.u : \Pi x : S.T} \qquad \textsc{Conv}\ \frac{\Gamma \vdash u : T \qquad \Gamma \vdash S : s \qquad T =_\beta S}{\Gamma \vdash u : S}$$

$$\textsc{Pi}\ \frac{\Gamma \vdash S : p \qquad \Gamma, x : S \vdash T : r \qquad (p, r, s) \in Rel}{\Gamma \vdash \Pi x : S.T : s}$$

$$\textsc{App}\ \frac{\Gamma \vdash u : \Pi x : S.T \qquad \Gamma \vdash v : S}{\Gamma \vdash u\ v : T[v/x]}$$

A generic framework of typed λ-calculus

$$\textsc{Nil}\ \frac{}{\varepsilon_{wf}} \qquad \textsc{Cons}\ \frac{\Gamma_{wf} \qquad \Gamma \vdash S : s}{(\Gamma, x : S)_{wf}} \qquad \textsc{Var}\ \frac{x : S \in \Gamma \qquad \Gamma_{wf}}{\Gamma \vdash x : S} \qquad \textsc{Sort}\ \frac{(s_1, s_2) \in Ax \qquad \Gamma_{wf}}{\Gamma \vdash s_1 : s_2}$$

$$\textsc{Lam}\ \frac{\Gamma, x : S \vdash u : T \qquad \Gamma \vdash \Pi x : S.T : s}{\Gamma \vdash \lambda x : S.u : \Pi x : S.T} \qquad \textsc{Conv}\ \frac{\Gamma \vdash u : T \qquad \Gamma \vdash S : s \qquad T =_\beta S}{\Gamma \vdash u : S}$$

$$\textsc{Pi}\ \frac{\Gamma \vdash S : p \qquad \Gamma, x : S \vdash T : r \qquad (p, r, s) \in Rel}{\Gamma \vdash \Pi x : S.T : s}$$

$$\textsc{App}\ \frac{\Gamma \vdash u : \Pi x : S.T \qquad \Gamma \vdash v : S}{\Gamma \vdash u\ v : T[v/x]}$$

A generic framework of typed λ-calculus

NIL
$$\frac{}{\varepsilon_{wf}}$$

CONS
$$\frac{\Gamma_{wf} \qquad \Gamma \vdash S : s}{(\Gamma, x : S)_{wf}}$$

VAR
$$\frac{x : S \in \Gamma \qquad \Gamma_{wf}}{\Gamma \vdash x : S}$$

SORT
$$\frac{(s_1, s_2) \in Ax \qquad \Gamma_{wf}}{\Gamma \vdash s_1 : s_2}$$

LAM
$$\frac{\Gamma, x : S \vdash u : T \qquad \Gamma \vdash \Pi x : S.T : s}{\Gamma \vdash \lambda x : S.u : \Pi x : S.T}$$

CONV
$$\frac{\Gamma \vdash u : T \qquad \Gamma \vdash S : s \qquad T =_\beta S}{\Gamma \vdash u : S}$$

PI
$$\frac{\Gamma \vdash S : p \qquad \Gamma, x : S \vdash T : r \qquad (p, r, s) \in Rel}{\Gamma \vdash \Pi x : S.T : s}$$

APP
$$\frac{\Gamma \vdash u : \Pi x : S.T \qquad \Gamma \vdash v : S}{\Gamma \vdash u\ v : T[v/x]}$$

What would be a Call by Value PTS ?

Syntax

t, u, v	$::=$	$\textbf{abort}\ V \mid t\ V$	term
V	$::=$	$x \mid \lambda x : S.t \mid \#\ t \mid T$	value
S, T	$::=$	$\Pi x : S.\mathcal{M}(T) \mid s \mid V$	type
$\mathcal{M}(T)$	$::=$	T	monad
Γ	$::=$	$\varepsilon \mid \Gamma, x : S$	context

Judgements

$\Gamma \vdash V : T$ $\qquad$ $\Gamma \vdash t : \mathcal{M}(T)$ $\qquad$ $\Gamma \vdash T \in s$

Reduction

$\textbf{abort}\ (\lambda x : T.u)\ V \Rightarrow_{\beta_v} u[V/x]$ $\qquad$ $\#\ \textbf{abort}\ V \Rightarrow V$

What does the monad need to provide ?

Typing rules that involve "internal types of the monad".

- Return
- Bind
- Run

The rules that deals with the state, the continuation, the...

ABORT
$$\frac{\Gamma \vdash V : T \quad \textbf{Return}}{\Gamma \vdash \textbf{abort}\ V : \mathcal{M}(T)}$$

LAM
$$\frac{\Gamma, x : S \vdash u : \mathcal{M}(T) \quad \Gamma \vdash \Pi x : S.\mathcal{M}(T) \in s}{\Gamma \vdash \lambda x : S.u : \Pi x : S.\mathcal{M}(T)}$$

NIL
$$\frac{}{\varepsilon_{wf}}$$

CONV
$$\frac{\Gamma \vdash u : \mathcal{M}(T) \quad \Gamma \vdash S \in s \quad T =_{\beta\eta} S}{\Gamma \vdash u : \mathcal{M}(S)}$$

RESET
$$\frac{\Gamma \vdash t : \mathcal{M}(T) \quad \textbf{Run}}{\Gamma \vdash \#\ t : T}$$

CONS
$$\frac{\Gamma_{wf} \quad \Gamma \vdash S \in s}{(\Gamma, x : S)_{wf}}$$

APP
$$\frac{\Gamma \vdash u : \mathcal{M}(\Pi x : S.M) \quad \Gamma \vdash v : S}{\Gamma \vdash u\ v : M[v/x]}$$

VAR
$$\frac{x : S \in \Gamma \quad \Gamma_{wf}}{\Gamma \vdash x : S}$$

SORT
$$\frac{(s_1, s_2) \in Ax \quad \Gamma_{wf}}{\Gamma \vdash s_1 \in s_2}$$

PI
$$\frac{\Gamma \vdash S \in p \qquad \Gamma, x : S \vdash \mathcal{M}(T) \in r \quad (p, r, s) \in Rel \quad \textbf{Bind}}{\Gamma \vdash \Pi x : S.\mathcal{M}(T) \in s}$$

ABORT
$$\frac{\Gamma \vdash V : T \quad \textbf{Return}}{\Gamma \vdash \textbf{abort}\ V : \mathcal{M}(T)}$$

LAM
$$\frac{\Gamma, x : S \vdash u : \mathcal{M}(T) \quad \Gamma \vdash \Pi x : S.\mathcal{M}(T) \in s}{\Gamma \vdash \lambda x : S.u : \Pi x : S.\mathcal{M}(T)}$$

NIL
$$\frac{}{\varepsilon_{wf}}$$

CONV
$$\frac{\Gamma \vdash u : \mathcal{M}(T) \quad \Gamma \vdash S \in s \quad T =_{\beta\eta} S}{\Gamma \vdash u : \mathcal{M}(S)}$$

RESET
$$\frac{\Gamma \vdash t : \mathcal{M}(T) \quad \textbf{Run}}{\Gamma \vdash \#\ t : T}$$

CONS
$$\frac{\Gamma_{wf} \quad \Gamma \vdash S \in s}{(\Gamma, x : S)_{wf}}$$

APP
$$\frac{\Gamma \vdash u : \mathcal{M}(\Pi x : S.M) \quad \Gamma \vdash v : S}{\Gamma \vdash u\ v : M[v/x]}$$

VAR
$$\frac{x : S \in \Gamma \quad \Gamma_{wf}}{\Gamma \vdash x : S}$$

SORT
$$\frac{(s_1, s_2) \in Ax \quad \Gamma_{wf}}{\Gamma \vdash s_1 \in s_2}$$

PI
$$\frac{\begin{array}{c}\Gamma \vdash S \in p \\ \Gamma, x : S \vdash \mathcal{M}(T) \in r \quad (p, r, s) \in Rel \quad \textbf{Bind}\end{array}}{\Gamma \vdash \Pi x : S.\mathcal{M}(T) \in s}$$

Classical CBV PTS

Syntax

t, u, v	$::=$	$\textbf{abort}\ V \mid t\ V \mid \textbf{catch}_{\alpha}\, t \mid \textbf{throw}_{\alpha}\, t$	term
V	$::=$	$x \mid \lambda x : S.t \mid \#\ t \mid T$	value
S, T, A	$::=$	$\mathbf{\Pi} x : S.\mathcal{M}(T) \mid s \mid V$	type
$\mathcal{M}(T)$	$::=$	$S{-}A/T$	monad
Γ	$::=$	$\varepsilon \mid \Gamma, x : S \mid \alpha : S{-}A$	context

PTS$^{\#}_{V}$ typing rules

PI
$$\frac{\Gamma \vdash S \in p \quad \Gamma, x : S \vdash T \in r \quad \Gamma, x : S \vdash A \in s_1 \quad \Gamma, x : S \vdash B \in s_2 \quad (p, r, s) \in Rel \quad (r, s_1, q) \in Rel \quad (q, s_2, o) \in Rel \quad (p, o, s) \in Rel}{\Gamma \vdash \Pi x : S . T - A / B : s}$$

ABORT
$$\frac{\Gamma \vdash t : B \quad \Gamma \vdash S \in p \quad \Gamma \vdash A \in q}{\Gamma \vdash \textbf{abort}\ t : S - A / B}$$

RESET
$$\frac{\Gamma \vdash t : T - T / S}{\Gamma \vdash \#\ t : S}$$

APP
$$\frac{\Gamma \vdash t : \Pi x : S . T - A / C - C / B \quad \Gamma \vdash w : S}{\Gamma \vdash t\ w : T[w/x] - A[w/x] / B[w/x]}$$

CATCH
$$\frac{\Gamma, \alpha : S - A \vdash t : S - A / B}{\Gamma \vdash \textbf{catch}_{\alpha} t : S - A / B}$$

THROW
$$\frac{\alpha : T - C \in \Gamma \quad \Gamma \vdash t : T - C / B \quad \Gamma \vdash S : p \quad \Gamma \vdash A : q}{\Gamma \vdash \textbf{throw}_{\alpha} t : S - A / B}$$

$\mathrm{PTS}^{\#}_{\mathrm{V}}$ typing rules

$$\textsc{Pi}\quad \frac{\Gamma \vdash S \in p \quad \Gamma, x : S \vdash T \in r \quad \Gamma, x : S \vdash A \in s_1 \quad \Gamma, x : S \vdash B \in s_2 \quad (p, r, s) \in Rel \quad (r, s_1, q) \in Rel \quad (q, s_2, o) \in Rel \quad (p, o, s) \in Rel}{\Gamma \vdash \Pi x : S . T - A / B : s}$$

$$\textsc{Abort}\quad \frac{\Gamma \vdash t : B \quad \Gamma \vdash S \in p \quad \Gamma \vdash A \in q}{\Gamma \vdash \mathbf{abort}\ t : S - A / B}$$

$$\textsc{Reset}\quad \frac{\Gamma \vdash t : T - T / S}{\Gamma \vdash \#\ t : S}$$

$$\textsc{App}\quad \frac{\Gamma \vdash t : \Pi x : S . T - A / C - C / B \quad \Gamma \vdash w : S}{\Gamma \vdash t\ w : T[w/x] - A[w/x] / B[w/x]}$$

$$\textsc{Catch}\quad \frac{\Gamma, \alpha : S - A \vdash t : S - A / B}{\Gamma \vdash \mathbf{catch}_{\alpha}\, t : S - A / B}$$

$$\textsc{Throw}\quad \frac{\alpha : T - C \in \Gamma \quad \Gamma \vdash t : T - C / B \quad \Gamma \vdash S : p \quad \Gamma \vdash A : q}{\Gamma \vdash \mathbf{throw}_{\alpha}\, t : S - A / B}$$

PTS$_V^\#$ typing rules

PI

$$\frac{\Gamma \vdash S \in p \qquad \Gamma, x:S \vdash T \in r \qquad \Gamma, x:S \vdash A \in s_1 \qquad \Gamma, x:S \vdash B \in s_2 \qquad (p,r,s) \in Rel \qquad (r,s_1,q) \in Rel \qquad (q,s_2,o) \in Rel \qquad (p,o,s) \in Rel}{\Gamma \vdash \mathbf{\Pi} x:S.T - A/B : s}$$

ABORT

$$\frac{\Gamma \vdash t : B \qquad \Gamma \vdash S \in p \qquad \Gamma \vdash A \in q}{\Gamma \vdash \mathbf{abort}\ t : S - A/B}$$

RESET

$$\frac{\Gamma \vdash t : T - T/S}{\Gamma \vdash \#\ t : S}$$

APP

$$\frac{\Gamma \vdash t : \mathbf{\Pi} x:S.T - A/C - C/B \qquad \Gamma \vdash w : S}{\Gamma \vdash t\ w : T[w/x] - A[w/x]/B[w/x]}$$

CATCH

$$\frac{\Gamma, \alpha : S - A \vdash t : S - A/B}{\Gamma \vdash \mathbf{catch}_\alpha t : S - A/B}$$

THROW

$$\frac{\alpha : T - C \in \Gamma \qquad \Gamma \vdash t : T - C/B \qquad \Gamma \vdash S : p \qquad \Gamma \vdash A : q}{\Gamma \vdash \mathbf{throw}_\alpha t : S - A/B}$$

Call By Name PTS with control

Syntax

t, u, v	$::=$	$x \mid \textbf{abort}\ V \mid t\ u \mid \textbf{catch}_\alpha t \mid \textbf{throw}_\alpha t$	term
V	$::=$	$\lambda x : \mathcal{M}(S).t \mid \#\ t \mid T$	value
S, T, A	$::=$	$\mathbf{\Pi} x : \mathcal{M}(S).\mathcal{M}(T) \mid s \mid V$	type
$\mathcal{M}(T)$	$::=$	$S - A / T$	monad
Γ	$::=$	$\varepsilon \mid \Gamma, x : \mathcal{M}(T) \mid \alpha : S - A$	context

Reduction

$$
\begin{array}{rcl}
(\textbf{abort}\ \lambda x : T.t)\ u & \Rightarrow_\beta & t[u/x] \\
\#\ (\textbf{abort}\ V) & \Rightarrow_\# & V \\
\hline
(\textbf{abort}\ V)\ (\textbf{throw}_\alpha u) & \Rightarrow & \textbf{throw}_\alpha u \\
(\textbf{throw}_\alpha u)\ v & \Rightarrow & \textbf{throw}_\alpha u \\
(\textbf{abort}\ V)\ (\textbf{catch}_\alpha u) & \Rightarrow & \textbf{catch}_\alpha((\textbf{abort}\ V) \\
 & & u[\textbf{throw}_\alpha(\textbf{abort}\ V\ u)/\textbf{throw}_\alpha u]) \\
(\textbf{catch}_\alpha u)\ v & \Rightarrow & \textbf{catch}_\alpha(u[\textbf{throw}_\alpha(t\ v)/\textbf{throw}_\alpha t]\ v) \\
\hline
\textbf{throw}_\alpha \textbf{abort}\ V & \Rightarrow & \textbf{abort}\ V \\
\textbf{catch}_\alpha \textbf{abort}\ V & \Rightarrow & \textbf{abort}\ V \\
\textbf{throw}_\alpha \textbf{catch}_\beta t & \Rightarrow & \textbf{throw}_\alpha t[\alpha/\beta] \\
\textbf{throw}_\alpha \textbf{throw}_\beta t & \Rightarrow & \textbf{throw}_\beta t \\
\textbf{catch}_\alpha \textbf{catch}_\beta t & \Rightarrow & \textbf{catch}_\alpha t[\alpha/\beta] \\
\#\ \textbf{catch}_\alpha t & \Rightarrow & \#\ t[\textbf{abort}\ \#\ t'/\textbf{throw}_\alpha t']
\end{array}
$$

Not reduction

abort $\#$ t This is the reset of monadic calculus when there is only only syntactical cateory.

catch$_\alpha$**throw**$_\beta t$ This is a generic μ at low level.

$\#$ **throw**$_\alpha t$ Is exactly what is forbidden.

(Some) PTS$^{\#}$ typing rules

Var

$$\frac{\Gamma_{wf} \qquad x : S - A/B \in \Gamma}{\Gamma \vdash x : S - A/B}$$

Reset

$$\frac{\Gamma \vdash u : S - S/T}{\Gamma \vdash u : T}$$

App

$$\frac{\Gamma \vdash u : \Pi x : S - A/B.T - C/D - C/w \qquad \Gamma \vdash v : S - A/B}{\Gamma \vdash u\ v : T[v/x] - D/w}$$

Pi

$$\frac{\begin{array}{c}\Gamma \vdash S \in s_1 \\ \Gamma \vdash A \in p_1 \quad \Gamma \vdash B \in q_1 \quad \Gamma \vdash T \in s_2 \quad \Gamma \vdash C \in p_2 \quad \Gamma \vdash D \in q_2 \quad (s_1, p_1, r_1) \in Rel \\ (r_1, q_1, o_1) \in Rel \quad (s_2, p_2, r_2) \in Rel \quad (r_2, q_2, o_2) \in Rel \quad (o_1, o_2, s) \in Rel\end{array}}{\Gamma \vdash \Pi x : S - A/B.T - C/D \in s}$$

CPS for $\mathrm{PTS}_{\mathrm{V}}^{\#}$

If $\Gamma \vdash t : T - A/B$,

$$t_* \quad = \quad \lambda k : \Pi x : T_+ . A_+ . (t)_k^{[]}$$

s_+	$\mapsto$	s
$\Pi x : A . B - C/D_+$	$\mapsto$	$\Pi x : A_+ . (B_+ \to C_+) \to D_+$
x_+	$\mapsto$	x
$(\lambda x : T . t)_+$	$\mapsto$	$\lambda x : T_+ . t_*$
$(\# \ t)_+$	$\mapsto$	$(t)_{id}^{[]}$
$(u \ v)_k^{\gamma}$	$\mapsto$	$(u)^{\gamma}_{\lambda f : \Pi x : A_+ . (B_+ \to U_+) \to V_+ . f \ v_+ \ k}$
$(\mathbf{catch}_\alpha t)_k^{\gamma}$	$\mapsto$	$(t)_k^{\gamma, \alpha \mapsto k}$
$(\mathbf{throw}_\alpha t)_k^{\gamma}$	$\mapsto$	$(t)_{\gamma(\alpha)}^{\gamma}$
$(\mathbf{abort} \ V)_k^{\gamma}$	$\mapsto$	$k \ V_+$

CPS for PTS$^{\#}$

If $\Gamma \vdash t : T - A/B$,

$$t_* \quad = \quad \lambda k : \Pi x : T_+ . A_+ . (t)_k^{[]}$$

s_+	$\mapsto$	s
$\Pi x : S - A/B . T - C/D_+$	$\mapsto$	$\Pi x : (S_+^\gamma \to A_+^\gamma) \to B_+^\gamma . (T_+^\gamma \to C_+^\gamma) \to D_+^\gamma$
$(\# \ t)_+$	$\mapsto$	$(t)_{id}^{[]}$
$\lambda x : S - A/B . t_+$	$\mapsto$	$(\lambda x : (S_+^\gamma \to A_+^\gamma) \to B_+ . t_*)$
$(x)_k^\gamma$	$\mapsto$	$x \ k$
$(\textbf{abort} \ V)_k^\gamma$	$\mapsto$	$k \ V_+$
$(u \ v)_k^\gamma$	$\mapsto$	$(u)^\gamma_{\lambda f : (\Pi x : (S_+^\gamma \to A_+^\gamma) \to B_+^\gamma . (T_+^\gamma \to C_+^\gamma) \to D_+^\gamma) . f \ v_*^\gamma \ k}$
$(\textbf{catch}_\alpha t)_k^\gamma$	$\mapsto$	$(t)_k^{\gamma, \alpha \mapsto k}$
$(\textbf{throw}_\alpha t)_k^\gamma$	$\mapsto$	$(t)_{\gamma(\alpha)}^\gamma$

Soundness

We need proofs of

1. Preservation of typing by CPS
   - For a given (St, Ax, Rel), If $\Gamma \vdash w : S$ then $\Gamma_+ \vdash w_+ : S_+$ with if $\Gamma \vdash t : S - A/B$ then $\Gamma_+, k : S_+ \to A_+ \vdash (t)_k : B_+$
   - For a given (St, Ax,Rel), If $\Gamma \vdash w : S$ then $\Gamma_+^\gamma \vdash w_+^\gamma : S_+^\gamma$ with if $\Gamma \vdash t : S - A/B$ then $\Gamma_+^\gamma, k : S_+ \to A_+ \vdash (t)_k^\gamma : B_+^\gamma$
2. Simulation
   - If $\Gamma \vdash u : S - A/B$ and $u \Rightarrow v$, then $\forall k, (u)_k \Rightarrow^* (v)_k$ with if $\Gamma \vdash w : S$ and $w \Rightarrow w'$, then $w_+ \Rightarrow *w'_+$
   - If $\Gamma \vdash u : S - A/B$ and $u \Rightarrow v$, then $\forall k, (u)_k^{\gamma\Gamma} \Rightarrow^* (v)_k^{\gamma\Gamma}$ with if $\Gamma \vdash w : S$ and $w \Rightarrow w'$, then $w_+^\gamma \Rightarrow *w'^\gamma_+$
3. Subject reduction
   - If $\Gamma \vdash u : s - A/B$ and $u \Rightarrow v$, then $\Gamma \vdash v : s - A/B$ with if $\Gamma \vdash w : S$ and $w \Rightarrow w'$, then $\Gamma \vdash w' : S$

Perspectives

- You don't have the monadic calculus because there is no way to call effect in argument that goes further than the application :

$$\textbf{let}\, x = (\textbf{throw}_\alpha\, u)\,\textbf{in}\; t \neq (\textbf{abort}\ \lambda x : \_.t)\ (\#\ \textbf{throw}_\alpha\, u)$$

Thinner control of allowed Ax and Rel in source language

- We can imagine to allow really restricted PTS and embed them in a PTS with a bigger Rel set.
- But how to link properties of the two languages.

Σ-types

- $\textbf{catch}_\alpha(\mathit{exists}\ 1, \textbf{throw}_\alpha(\mathit{exists}\ 2, \mathit{eq\_refl}\ 2)) : 1 = 2$???

Perspectives

- You don't have the monadic calculus because there is no way to call effect in argument that goes further than the application :

$$\textbf{let}\, x = (\textbf{throw}_\alpha\, u)\,\textbf{in}\, t \neq (\textbf{abort}\ \lambda x : \_.t)\ (\#\ \textbf{throw}_\alpha\, u)$$

Thinner control of allowed Ax and Rel in source language

- We can imagine to allow really restricted PTS and embed them in a PTS with a bigger Rel set.
- But how to link properties of the two languages.

Σ-types

- $\textbf{catch}_\alpha(\textit{exists}\ 1, \textbf{throw}_\alpha(\textit{exists}\ 2, \textit{eq\_refl}\ 2)) : 1 = 2$???

Thanks