Monadic translation of sequent calculus for classical logic

Luís Pinto[1]
Univ. Minho
Braga, Portugal

Theory Seminar at Inst. of Cybernetics
Tallinn, Estonia

2 December 2010

---

[1] Joint work with **José Espírito Santo**, **Ralph Matthes**, **Koji Nakazawa**

Generalities on Curry-Howard correspondence (I)

Intuitionistic implication in natural deduction and simply-typed λ-calculus are a perfect match:

- assumptions A type variables x;
- rule $\dfrac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{A \supset B}\ Intro$ types abstractions $\lambda x.t$;
- rule $\dfrac{A \supset B \quad A}{B}\ Elim$ types applications tu;
- removal of a *maximal* formula $A \supset B$, i.e. a detour

$$\dfrac{\dfrac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{A \supset B}\ Intro \qquad A}{B}\ Elim,$$

is β-reduction (*normalisation*).

Generalities on Curry-Howard correspondence (I)

Intuitionistic implication in natural deduction and simply-typed λ-calculus are a perfect match:

- assumptions A type variables x;
- rule $\dfrac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \supset B}\ Intro$ types abstractions $\lambda x.t$;
- rule $\dfrac{A \supset B \quad A}{B}\ Elim$ types applications tu;
- removal of a *maximal* formula $A \supset B$, i.e. a detour

$$\dfrac{\dfrac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \supset B}\ Intro \qquad A}{B}\ Elim,$$

is β-reduction (*normalisation*).

Generalities on Curry-Howard correspondence (I)

Intuitionistic implication in natural deduction and simply-typed λ-calculus are a perfect match:

- assumptions A type variables x;
- rule $\dfrac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \supset B}\ \mathit{Intro}$ types abstractions $\lambda x.t$;
- rule $\dfrac{A \supset B \quad A}{B}\ \mathit{Elim}$ types applications tu;
- removal of a *maximal* formula $A \supset B$, i.e. a detour

$$\dfrac{\dfrac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \supset B}\ \mathit{Intro} \qquad A}{B}\ \mathit{Elim},$$

is β-reduction (*normalisation*).

Generalities on Curry-Howard correspondence (II)

How about extensions to sequent calculus and to classical logic?

Intuitionistic sequent calculus:

- typical rules:

$$\frac{\Gamma \vdash A \quad \Gamma, B \vdash C}{\Gamma, A \supset B \vdash C}\ Left \qquad \frac{\Gamma \vdash C \quad \Gamma, C \vdash A}{\Gamma \vdash A}\ Cut$$

- issue: many proofs are essentially the same (differ up to permutation of inferences)
- $\overline{\lambda}$-calculus of Herbelin addresses this issue: there are two forms of sequents, one, $\Gamma | l : A \vdash B$, has a selected formula on LHS and types *lists* $l := [] | u :: l$:

$$\frac{\Gamma \vdash u : A \quad \Gamma | l : B \vdash C}{\Gamma | u :: l : A \supset B \vdash C}\ Left \qquad \frac{}{\Gamma | [] : A \vdash A}\ Ax$$

Generalities on Curry-Howard correspondence (II)

How about extensions to sequent calculus and to classical logic?

Intuitionistic sequent calculus:

- typical rules:

$$\frac{\Gamma \vdash A \quad \Gamma, B \vdash C}{\Gamma, A \supset B \vdash C}\ \mathit{Left} \qquad \frac{\Gamma \vdash C \quad \Gamma, C \vdash A}{\Gamma \vdash A}\ \mathit{Cut}$$

- issue: many proofs are essentially the same (differ up to permutation of inferences)
- $\overline{\lambda}$-calculus of Herbelin addresses this issue: there are two forms of sequents, one, $\Gamma | l : A \vdash B$, has a selected formula on LHS and types *lists* $l := [] | u :: l$:

$$\frac{\Gamma \vdash u : A \quad \Gamma | l : B \vdash C}{\Gamma | u :: l : A \supset B \vdash C}\ \mathit{Left} \qquad \frac{}{\Gamma | [] : A \vdash A}\ \mathit{Ax}$$

Generalities on Curry-Howard correspondence (II)

How about extensions to sequent calculus and to classical logic?

Intuitionistic sequent calculus:

- typical rules:

$$\frac{\Gamma \vdash A \quad \Gamma, B \vdash C}{\Gamma, A \supset B \vdash C}\ \mathit{Left} \qquad \frac{\Gamma \vdash C \quad \Gamma, C \vdash A}{\Gamma \vdash A}\ \mathit{Cut}$$

- issue: many proofs are essentially the same (differ up to permutation of inferences)
- $\overline{\lambda}$-calculus of Herbelin addresses this issue: there are two forms of sequents, one, $\Gamma | l : A \vdash B$, has a selected formula on LHS and types *lists* $l := [] | u :: l$:

$$\frac{\Gamma \vdash u : A \quad \Gamma | l : B \vdash C}{\Gamma | u :: l : A \supset B \vdash C}\ \mathit{Left} \qquad \frac{}{\Gamma | [] : A \vdash A}\ \mathit{Ax}$$

Generalities on Curry-Howard correspondence (II)

How about extensions to sequent calculus and to classical logic?

Intuitionistic sequent calculus:

- typical rules:

$$\frac{\Gamma \vdash A \quad \Gamma, B \vdash C}{\Gamma, A \supset B \vdash C}\ \mathit{Left} \qquad \frac{\Gamma \vdash C \quad \Gamma, C \vdash A}{\Gamma \vdash A}\ \mathit{Cut}$$

- issue: many proofs are essentially the same (differ up to permutation of inferences)
- $\overline{\lambda}$-calculus of Herbelin addresses this issue: there are two forms of sequents, one, $\Gamma | l : A \vdash B$, has a selected formula on LHS and types *lists* $l := [] | u :: l$:

$$\frac{\Gamma \vdash u : A \quad \Gamma | l : B \vdash C}{\Gamma | u :: l : A \supset B \vdash C}\ \mathit{Left} \qquad \frac{}{\Gamma | [] : A \vdash A}\ \mathit{Ax}$$

Generalities on Curry-Howard correspondence (III)

Classical natural deduction:

- one option is adding $\dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ RAA$
- other option is multiple-conclusion sequents as in Parigot's $\lambda\mu$:
  - expressions: $\begin{array}{rcll} t, u & ::= & x \mid \lambda x.t \mid tu \mid \mu a.c & (\mathit{terms}) \\ c & ::= & at & (\mathit{commands}) \end{array}$

    (a is called *name* but also *co-variable/continuation variable*)
  - sequents: $\Gamma \vdash t : A | \Delta$ and $c : (\Gamma \vdash \Delta)$
    (Γ resp Δ consist of declarations $x : A$ resp $a : A$)
  - typing: $\dfrac{\Gamma \vdash t : A | a : A, \Delta}{at : (\Gamma \vdash a : A, \Delta)}\ \mathit{Pass} \qquad \dfrac{c : (\Gamma \vdash a : A, \Delta)}{\Gamma \vdash \mu a.c : A | \Delta}\ \mathit{Act}$

Classical sequent calculus:

- Unrestricts intuitionistic sequent calculus, by allowing sequents with multiple conclusions
- Curien-Herbelin proposed the elegant calculus $\overline{\lambda}\mu\tilde{\mu}$ (to be detailed ahead), where dualites like cbn/cbv emerge.

Generalities on Curry-Howard correspondence (III)

Classical natural deduction:

- one option is adding $\dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ RAA$
- other option is multiple-conclusion sequents as in Parigot's $\lambda\mu$:
  - expressions:
    $$\begin{array}{rcll} t, u & ::= & x \mid \lambda x.t \mid tu \mid \mu a.c & (\textit{terms}) \\ c & ::= & at & (\textit{commands}) \end{array}$$
    (a is called *name* but also *co-variable/continuation variable*)
  - sequents: $\Gamma \vdash t : A|\Delta$ and $c : (\Gamma \vdash \Delta)$
    (Γ resp Δ consist of declarations $x : A$ resp $a : A$)
  - typing: $\dfrac{\Gamma \vdash t : A|a : A, \Delta}{at : (\Gamma \vdash a : A, \Delta)}\ Pass \qquad \dfrac{c : (\Gamma \vdash a : A, \Delta)}{\Gamma \vdash \mu a.c : A|\Delta}\ Act$

Classical sequent calculus:

- Unrestricts intuitionistic sequent calculus, by allowing sequents with multiple conclusions
- Curien-Herbelin proposed the elegant calculus $\bar{\lambda}\mu\tilde{\mu}$ (to be detailed ahead), where dualites like cbn/cbv emerge.

Generalities on Curry-Howard correspondence (III)

Classical natural deduction:

- one option is adding $\dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ RAA$
- other option is multiple-conclusion sequents as in Parigot's $\lambda\mu$:
  - expressions: $\begin{array}{rcll} t, u & ::= & x \mid \lambda x.t \mid tu \mid \mu a.c & (terms) \\ c & ::= & at & (commands) \end{array}$

    (a is called *name* but also *co-variable/continuation variable*)
  - sequents: $\Gamma \vdash t : A | \Delta$ and $c : (\Gamma \vdash \Delta)$
    (Γ resp Δ consist of declarations $x : A$ resp $a : A$)
  - typing: $\dfrac{\Gamma \vdash t : A | a : A, \Delta}{at : (\Gamma \vdash a : A, \Delta)}\ Pass \qquad \dfrac{c : (\Gamma \vdash a : A, \Delta)}{\Gamma \vdash \mu a.c : A | \Delta}\ Act$

Classical sequent calculus:

- Unrestricts intuitionistic sequent calculus, by allowing sequents with multiple conclusions
- Curien-Herbelin proposed the elegant calculus $\bar{\lambda}\mu\tilde{\mu}$ (to be detailed ahead), where dualites like cbn/cbv emerge.

Generalities on Curry-Howard correspondence (III)

Classical natural deduction:

- one option is adding $\dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ RAA$
- other option is multiple-conclusion sequents as in Parigot's $\lambda\mu$:
  - expressions: $$\begin{array}{rcll} t, u & ::= & x \mid \lambda x.t \mid tu \mid \mu a.c & (terms) \\ c & ::= & at & (commands) \end{array}$$
    (a is called *name* but also *co-variable/continuation variable*)
  - sequents: $\Gamma \vdash t : A|\Delta$ and $c : (\Gamma \vdash \Delta)$
    (Γ resp Δ consist of declarations $x : A$ resp $a : A$)
  - typing: $\dfrac{\Gamma \vdash t : A|a : A, \Delta}{at : (\Gamma \vdash a : A, \Delta)}\ Pass \qquad \dfrac{c : (\Gamma \vdash a : A, \Delta)}{\Gamma \vdash \mu a.c : A|\Delta}\ Act$

Classical sequent calculus:

- Unrestricts intuitionistic sequent calculus, by allowing sequents with multiple conclusions
- Curien-Herbelin proposed the elegant calculus $\overline{\lambda}\mu\tilde{\mu}$ (to be detailed ahead), where dualites like cbn/cbv emerge.

Generalities on Curry-Howard correspondence (III)

Classical natural deduction:

- one option is adding $\dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ RAA$
- other option is multiple-conclusion sequents as in Parigot's $\lambda\mu$:
  - expressions: $\begin{array}{lcll} t, u & ::= & x \mid \lambda x.t \mid tu \mid \mu a.c & (terms) \\ c & ::= & at & (commands) \end{array}$

    (a is called *name* but also *co-variable/continuation variable*)
  - sequents: $\Gamma \vdash t : A|\Delta$ and $c : (\Gamma \vdash \Delta)$
    (Γ resp Δ consist of declarations $x : A$ resp $a : A$)
  - typing: $\dfrac{\Gamma \vdash t : A|a : A, \Delta}{at : (\Gamma \vdash a : A, \Delta)}\ Pass \qquad \dfrac{c : (\Gamma \vdash a : A, \Delta)}{\Gamma \vdash \mu a.c : A|\Delta}\ Act$

Classical sequent calculus:

- Unrestricts intuitionistic sequent calculus, by allowing sequents with multiple conclusions
- Curien-Herbelin proposed the elegant calculus $\overline{\lambda}\mu\tilde{\mu}$ (to be detailed ahead), where dualites like cbn/cbv emerge.

Generalities on Curry-Howard correspondence (III)

Classical natural deduction:

- one option is adding $\dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ RAA$
- other option is multiple-conclusion sequents as in Parigot's $\lambda\mu$:
  - expressions:
    $$\begin{array}{lcll} t, u & ::= & x \mid \lambda x.t \mid tu \mid \mu a.c & (\mathit{terms}) \\ c & ::= & at & (\mathit{commands}) \end{array}$$
    (a is called *name* but also *co-variable/continuation variable*)
  - sequents: $\Gamma \vdash t : A|\Delta$ and $c : (\Gamma \vdash \Delta)$
    (Γ resp Δ consist of declarations $x : A$ resp $a : A$)
  - typing: $\dfrac{\Gamma \vdash t : A|a : A, \Delta}{at : (\Gamma \vdash a : A, \Delta)}\ Pass \qquad \dfrac{c : (\Gamma \vdash a : A, \Delta)}{\Gamma \vdash \mu a.c : A|\Delta}\ Act$

Classical sequent calculus:

- Unrestricts intuitionistic sequent calculus, by allowing sequents with multiple conclusions
- Curien-Herbelin proposed the elegant calculus $\overline{\lambda}\mu\tilde{\mu}$ (to be detailed ahead), where dualites like cbn/cbv emerge.

cps translation of λ-calculus

The cbn case:

- Terms: $$\begin{aligned} \overline{x} &= x \\ \overline{\lambda x.t} &= \lambda k.k(\lambda x.\overline{t}) \\ \overline{tu} &= \lambda k.\overline{t}(\lambda f.f\overline{u}k) \end{aligned}$$
- Types: $\overline{A} = \neg\neg A^*$, and $X^* = X$, $(A \supset B)^* = \overline{A} \supset \overline{B}$
- Preservation of typing: $\dfrac{\Gamma \vdash t : A}{\overline{\Gamma} \vdash \overline{t} : \overline{A}}$ is admissible.

Hatcliff-Danvy decomposition of cps's:

cps translation of λ-calculus

The cbn case:

- Terms:
$$\begin{array}{rcl} \overline{x} & = & x \\ \overline{\lambda x.t} & = & \lambda k.k(\lambda x.\overline{t}) \\ \overline{tu} & = & \lambda k.\overline{t}(\lambda f.f\overline{u}k) \end{array}$$

- Types: $\overline{A} = \neg\neg A^*$, and $X^* = X$, $(A \supset B)^* = \overline{A} \supset \overline{B}$

- Preservation of typing: $\dfrac{\Gamma \vdash t : A}{\overline{\Gamma} \vdash \overline{t} : \overline{A}}$ is admissible.

Hatcliff-Danvy decomposition of cps's:

cps translation of λ-calculus

The cbn case:

- Terms:
$$\begin{array}{rcl} \overline{x} & = & x \\ \overline{\lambda x.t} & = & \lambda k.k(\lambda x.\overline{t}) \\ \overline{tu} & = & \lambda k.\overline{t}(\lambda f.f\overline{u}k) \end{array}$$
- Types: $\overline{A} = \neg\neg A^*$, and $X^* = X$, $(A \supset B)^* = \overline{A} \supset \overline{B}$
- Preservation of typing: $\dfrac{\Gamma \vdash t : A}{\overline{\Gamma} \vdash \overline{t} : \overline{A}}$ is admissible.

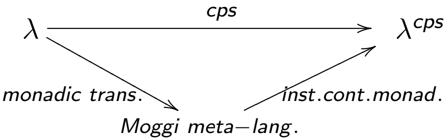
Hatcliff-Danvy decomposition of cps's:

cps translation of λ-calculus

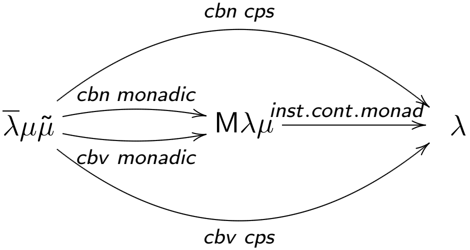
The cbn case:

- Terms:
$$\begin{array}{rcl} \overline{x} & = & x \\ \overline{\lambda x.t} & = & \lambda k.k(\lambda x.\overline{t}) \\ \overline{tu} & = & \lambda k.\overline{t}(\lambda f.f\overline{u}k) \end{array}$$
- Types: $\overline{A} = \neg\neg A^*$, and $X^* = X$, $(A \supset B)^* = \overline{A} \supset \overline{B}$
- Preservation of typing: $\dfrac{\Gamma \vdash t : A}{\overline{\Gamma} \vdash \overline{t} : \overline{A}}$ is admissible.

Hatcliff-Danvy decomposition of cps's:

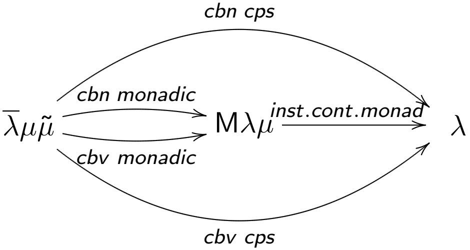


Overview of what we achieve



- $\mathrm{M}\lambda\mu$ is a new monadic language
- all maps strictly preserve reduction and allow inheritance of SN

Overview of what we achieve



- $\mathrm{M}\lambda\mu$ is a new monadic language
- all maps strictly preserve reduction and allow inheritance of SN

$\overline{\lambda}\mu\tilde{\mu}$-calculus of Curien-Herbelin

Types: $A, B \quad ::= \quad X \mid A \supset B$

Expressions:

$$t \quad ::= \quad \underbrace{x \mid \lambda x.t}_{values} \mid \mu a.c \qquad (terms)$$

$$e \quad ::= \quad \underbrace{a \mid u :: e}_{co-values} \mid \tilde{\mu} x.c \qquad (co-terms)$$

$$c \quad ::= \quad \langle t | e \rangle \qquad (commands)$$

Typing judgements:

$$\Gamma \vdash t : A | \Delta \qquad \Gamma | e : A \vdash \Delta \qquad c : (\Gamma \vdash \Delta)$$

Γ: type context for variables (x)
Δ: type context for co-variables (a)

$\overline{\lambda}\mu\tilde{\mu}$-calculus of Curien-Herbelin

Types: $A, B \quad ::= \quad X \mid A \supset B$

Expressions:

$$t \quad ::= \quad \underbrace{x \mid \lambda x.t}_{values} \mid \mu a.c \qquad (terms)$$

$$e \quad ::= \quad \underbrace{a \mid u :: e}_{co-values} \mid \tilde{\mu} x.c \quad (co-terms)$$

$$c \quad ::= \quad \langle t | e \rangle \qquad (commands)$$

Typing judgements:

$$\Gamma \vdash t : A | \Delta \qquad \Gamma | e : A \vdash \Delta \qquad c : (\Gamma \vdash \Delta)$$

Γ: type context for variables (x)
Δ: type context for co-variables (a)

$\overline{\lambda}\mu\tilde{\mu}$-calculus of Curien-Herbelin

Types: $A, B \quad ::= \quad X \mid A \supset B$

Expressions:

$$
\begin{array}{llll}
t & ::= & \underbrace{x \mid \lambda x.t}_{values} \mid \mu a.c & (terms) \\
e & ::= & \underbrace{a \mid u :: e}_{co-values} \mid \tilde{\mu} x.c & (co-terms) \\
c & ::= & \langle t | e \rangle & (commands)
\end{array}
$$

Typing judgements:

$$\Gamma \vdash t : A | \Delta \qquad \Gamma | e : A \vdash \Delta \qquad c : (\Gamma \vdash \Delta)$$

Γ: type context for variables (x)
Δ: type context for co-variables (a)

Typing rules of $\overline{\lambda}\mu\tilde{\mu}$

$$\frac{}{\Gamma, x : A \vdash x : A | \Delta} \; R - Ax \qquad \frac{}{\Gamma | a : A \vdash a : A, \Delta} \; L - Ax$$

$$\frac{\Gamma, x : A \vdash t : B | \Delta}{\Gamma \vdash \lambda x.t : A \supset B | \Delta} \; R- \supset \qquad \frac{\Gamma \vdash u : A | \Delta \quad \Gamma | e : B \vdash \Delta}{\Gamma | u :: e : A \supset B \vdash \Delta} \; L- \supset$$

$$\frac{c : (\Gamma \vdash a : A, \Delta)}{\Gamma \vdash \mu a.c : A | \Delta} \; R - Sel \qquad \frac{c : (\Gamma, x : A \vdash \Delta)}{\Gamma | \tilde{\mu} x.c : A \vdash \Delta} \; L - Sel$$

$$\frac{\Gamma \vdash t : A | \Delta \quad \Gamma | e : A \vdash \Delta}{\langle t | e \rangle : (\Gamma \vdash \Delta)} \; Cut$$

Reduction rules of $\overline{\lambda}\mu\tilde{\mu}$

$$
\begin{array}{rrcl}
(\beta) & \langle \lambda x.t | u :: e \rangle & \to & \langle u | \tilde{\mu} x.\langle t|e \rangle \rangle \\
(\sigma) & \langle t | \tilde{\mu} x.c \rangle & \to & [t/x]c \\
(\pi) & \langle \mu a.c | e \rangle & \to & [e/a]c \\
(\eta_{\tilde{\mu}}) & \tilde{\mu} x.\langle x | e \rangle & \to & e, \text{ if } x \notin e \\
(\eta_{\mu}) & \mu a.\langle t | a \rangle & \to & t, \text{ if } a \notin t
\end{array}
$$

The set of rules is SN (for typed terms), but not confluent due to the critical pair:

Two confluent fragments emerge:

- call-by-value $\overline{\lambda}\mu\tilde{\mu}$: in the σ-rule t must be a value.
- call-by-name $\overline{\lambda}\mu\tilde{\mu}$: in the π-rule e must be a co-value.

Reduction rules of $\overline{\lambda}\mu\tilde{\mu}$

$$
\begin{array}{rrcl}
(\beta) & \langle \lambda x.t | u :: e \rangle & \rightarrow & \langle u | \tilde{\mu} x.\langle t | e \rangle \rangle \\
(\sigma) & \langle t | \tilde{\mu} x.c \rangle & \rightarrow & [t/x]c \\
(\pi) & \langle \mu a.c | e \rangle & \rightarrow & [e/a]c \\
(\eta_{\tilde{\mu}}) & \tilde{\mu} x.\langle x | e \rangle & \rightarrow & e, \text{ if } x \notin e \\
(\eta_{\mu}) & \mu a.\langle t | a \rangle & \rightarrow & t, \text{ if } a \notin t
\end{array}
$$

The set of rules is SN (for typed terms), but not confluent due to the critical pair:

$$
[\tilde{\mu} x.c'/a]c \;\overset{\pi}{\longleftarrow}\; \langle \mu a.c | \tilde{\mu} x.c' \rangle \;\overset{\sigma}{\longrightarrow}\; [\mu a.c/x]c'
$$

Two confluent fragments emerge:

- call-by-value $\overline{\lambda}\mu\tilde{\mu}$: in the σ-rule t must be a value.
- call-by-name $\overline{\lambda}\mu\tilde{\mu}$: in the π-rule e must be a co-value.

Monadic meta-language of Moggi

The meta-language adds to simply typed lambda-calculus:

- Types: $A, B ::= ...|MA$ (monadic types)
- Expressions: $t, u ::= ...|\eta t|\text{bind}(t, x.u)$
- Typing rules:

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \eta t : MA} \qquad \frac{\Gamma \vdash t : MA \quad \Gamma, x : A \vdash u : MB}{\Gamma \vdash \text{bind}(t, x.u) : MB}$$

- Reduction rules (equations in Moggi):

$$\begin{array}{lrcl} (\sigma) & \text{bind}(\eta t, x.u) & \to & [t/x]u \\ (\text{assoc}) & \text{bind}(\text{bind}(t, x.u), y.s) & \to & \text{bind}(t, x.\text{bind}(u, y.s)) \\ (\eta_{\text{bind}}) & \text{bind}(t, x.\eta x) & \to & t \end{array}$$

The reduction system is confluent and SN.

Monadic meta-language of Moggi

The meta-language adds to simply typed lambda-calculus:

- Types: $A, B ::= ...|MA$ (monadic types)
- Expressions: $t, u ::= ...|\eta t|\text{bind}(t, x.u)$
- Typing rules:

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \eta t : MA} \qquad \frac{\Gamma \vdash t : MA \quad \Gamma, x : A \vdash u : MB}{\Gamma \vdash \text{bind}(t, x.u) : MB}$$

- Reduction rules (equations in Moggi):

$$\begin{array}{rrcl} (\sigma) & \text{bind}(\eta t, x.u) & \rightarrow & [t/x]u \\ (\text{assoc}) & \text{bind}(\text{bind}(t, x.u), y.s) & \rightarrow & \text{bind}(t, x.\text{bind}(u, y.s)) \\ (\eta_{\text{bind}}) & \text{bind}(t, x.\eta x) & \rightarrow & t \end{array}$$

The reduction system is confluent and SN.

Monadic $\lambda\mu$-calculus M$\lambda\mu$

Types: $A, B ::= X \mid A \supset B \mid MA$

Expressions:
$$\begin{array}{lcll} t, u & ::= & x \mid \lambda x.t \mid tu \mid \mu a.c \mid \eta t & (\textit{terms}) \\ c & ::= & at \mid \mathrm{bind}(t, x.c) & (\textit{commands}) \end{array}$$

Typing judgements: $\Gamma \vdash t : A|\Delta$ and $c : (\Gamma \vdash \Delta)$.

Δ consists of declarations $a : MA$ (just monadic types).

Some typing rules:

$$\frac{\Gamma \vdash t : MA|a : MA, \Delta}{at : (\Gamma \vdash a : MA, \Delta)}\ \mathit{Pass} \qquad \frac{c : (\Gamma \vdash a : MA, \Delta)}{\Gamma \vdash \mu a.c : MA|\Delta}\ \mathit{Act}$$

$$\frac{\Gamma \vdash s : A|\Delta}{\Gamma \vdash \eta s : MA|\Delta} \qquad \frac{\Gamma \vdash r : MA|\Delta \quad c : (\Gamma, x : A \vdash \Delta)}{\mathrm{bind}(r, x.c) : (\Gamma \vdash \Delta)}$$

Contexts:

$$C ::= a[\,] \mid \mathrm{bind}([\,], x.c) \mid \mathrm{bind}(\eta[\,], x.c)$$

$C[t]$ means fill the hole of C with t.

$[C/a]c$ means substitution in c of au by $C[u]$.

Monadic $\lambda\mu$-calculus M$\lambda\mu$

Types: $A, B ::= X \mid A \supset B \mid MA$

Expressions:
$$\begin{array}{lcll} t, u & ::= & x \mid \lambda x.t \mid tu \mid \mu a.c \mid \eta t & (\textit{terms}) \\ c & ::= & at \mid \mathrm{bind}(t, x.c) & (\textit{commands}) \end{array}$$

Typing judgements: $\Gamma \vdash t : A|\Delta$ and $c : (\Gamma \vdash \Delta)$.

Δ consists of declarations $a : MA$ (just monadic types).

Some typing rules:

$$\frac{\Gamma \vdash t : MA|a : MA, \Delta}{at : (\Gamma \vdash a : MA, \Delta)}\ \textit{Pass} \qquad \frac{c : (\Gamma \vdash a : MA, \Delta)}{\Gamma \vdash \mu a.c : MA|\Delta}\ \textit{Act}$$

$$\frac{\Gamma \vdash s : A|\Delta}{\Gamma \vdash \eta s : MA|\Delta} \qquad \frac{\Gamma \vdash r : MA|\Delta \quad c : (\Gamma, x : A \vdash \Delta)}{\mathrm{bind}(r, x.c) : (\Gamma \vdash \Delta)}$$

Contexts:

$$C \ ::= \ a[\,] \mid \mathrm{bind}([\,], x.c) \mid \mathrm{bind}(\eta[\,], x.c)$$

$C[t]$ means fill the hole of C with t.

$[C/a]c$ means substitution in c of au by $C[u]$.

Monadic $\lambda\mu$-calculus M$\lambda\mu$

Types: $A, B ::= X \,|\, A \supset B \,|\, MA$

Expressions:

$$\begin{array}{lcll} t, u & ::= & x \,|\, \lambda x.t \,|\, tu \,|\, \mu a.c \,|\, \eta t & (\textit{terms}) \\ c & ::= & at \,|\, \mathrm{bind}(t, x.c) & (\textit{commands}) \end{array}$$

Typing judgements: $\Gamma \vdash t : A|\Delta$ and $c : (\Gamma \vdash \Delta)$.

Δ consists of declarations $a : MA$ (just monadic types).

Some typing rules:

$$\frac{\Gamma \vdash t : MA|a : MA, \Delta}{at : (\Gamma \vdash a : MA, \Delta)}\ Pass \qquad \frac{c : (\Gamma \vdash a : MA, \Delta)}{\Gamma \vdash \mu a.c : MA|\Delta}\ Act$$

$$\frac{\Gamma \vdash s : A|\Delta}{\Gamma \vdash \eta s : MA|\Delta} \qquad \frac{\Gamma \vdash r : MA|\Delta \quad c : (\Gamma, x : A \vdash \Delta)}{\mathrm{bind}(r, x.c) : (\Gamma \vdash \Delta)}$$

Contexts:

$$C \quad ::= \quad a[\,] \,|\, \mathrm{bind}([\,], x.c) \,|\, \mathrm{bind}(\eta[\,], x.c)$$

$C[t]$ means fill the hole of C with t.

$[C/a]c$ means substitution in c of au by $C[u]$.

Monadic $\lambda\mu$-calculus M$\lambda\mu$

Types: $A, B ::= X \,|\, A \supset B \,|\, MA$

Expressions:

$$
\begin{array}{rcll}
t, u & ::= & x \,|\, \lambda x.t \,|\, tu \,|\, \mu a.c \,|\, \eta t & (\textit{terms}) \\
c & ::= & at \,|\, \mathrm{bind}(t, x.c) & (\textit{commands})
\end{array}
$$

Typing judgements: $\Gamma \vdash t : A|\Delta$ and $c : (\Gamma \vdash \Delta)$.

Δ consists of declarations $a : MA$ (just monadic types).

Some typing rules:

$$
\frac{\Gamma \vdash t : MA | a : MA, \Delta}{at : (\Gamma \vdash a : MA, \Delta)}\ Pass \qquad \frac{c : (\Gamma \vdash a : MA, \Delta)}{\Gamma \vdash \mu a.c : MA|\Delta}\ Act
$$

$$
\frac{\Gamma \vdash s : A|\Delta}{\Gamma \vdash \eta s : MA|\Delta} \qquad \frac{\Gamma \vdash r : MA|\Delta \quad c : (\Gamma, x : A \vdash \Delta)}{\mathrm{bind}(r, x.c) : (\Gamma \vdash \Delta)}
$$

Contexts:

$$
C \quad ::= \quad a[\,] \,|\, \mathrm{bind}([\,], x.c) \,|\, \mathrm{bind}(\eta[\,], x.c)
$$

$C[t]$ means fill the hole of C with t.

$[C/a]c$ means substitution in c of au by $C[u]$.

Monadic $\lambda\mu$-calculus M$\lambda\mu$

Types: $A, B ::= X \,|\, A \supset B \,|\, MA$

Expressions:

$$\begin{array}{lll} t, u & ::= & x \,|\, \lambda x.t \,|\, tu \,|\, \mu a.c \,|\, \eta t \quad (\textit{terms}) \\ c & ::= & at \,|\, \mathrm{bind}(t, x.c) \quad (\textit{commands}) \end{array}$$

Typing judgements: $\Gamma \vdash t : A|\Delta$ and $c : (\Gamma \vdash \Delta)$.

Δ consists of declarations $a : MA$ (just monadic types).

Some typing rules:

$$\frac{\Gamma \vdash t : MA|a : MA, \Delta}{at : (\Gamma \vdash a : MA, \Delta)}\; Pass \qquad \frac{c : (\Gamma \vdash a : MA, \Delta)}{\Gamma \vdash \mu a.c : MA|\Delta}\; Act$$

$$\frac{\Gamma \vdash s : A|\Delta}{\Gamma \vdash \eta s : MA|\Delta} \qquad \frac{\Gamma \vdash r : MA|\Delta \quad c : (\Gamma, x : A \vdash \Delta)}{\mathrm{bind}(r, x.c) : (\Gamma \vdash \Delta)}$$

Contexts:

$$C \;::=\; a[\,] \,|\, \mathrm{bind}([\,], x.c) \,|\, \mathrm{bind}(\eta[\,], x.c)$$

$C[t]$ means fill the hole of C with t.

$[C/a]c$ means substitution in c of au by $C[u]$.

Monadic $\lambda\mu$-calculus M$\lambda\mu$

Types: $A, B ::= X \,|\, A \supset B \,|\, MA$

Expressions:
$$t, u ::= x \,|\, \lambda x.t \,|\, tu \,|\, \mu a.c \,|\, \eta t \quad (terms)$$
$$c ::= at \,|\, \mathrm{bind}(t, x.c) \quad (commands)$$

Typing judgements: $\Gamma \vdash t : A | \Delta$ and $c : (\Gamma \vdash \Delta)$.

Δ consists of declarations $a : MA$ (just monadic types).

Some typing rules:

$$\frac{\Gamma \vdash t : MA | a : MA, \Delta}{at : (\Gamma \vdash a : MA, \Delta)}\ Pass \qquad \frac{c : (\Gamma \vdash a : MA, \Delta)}{\Gamma \vdash \mu a.c : MA | \Delta}\ Act$$

$$\frac{\Gamma \vdash s : A | \Delta}{\Gamma \vdash \eta s : MA | \Delta} \qquad \frac{\Gamma \vdash r : MA | \Delta \quad c : (\Gamma, x : A \vdash \Delta)}{\mathrm{bind}(r, x.c) : (\Gamma \vdash \Delta)}$$

Contexts:
$$C ::= a[\,] \,|\, \mathrm{bind}([\,], x.c) \,|\, \mathrm{bind}(\eta[\,], x.c)$$

$C[t]$ means fill the hole of C with t.

$[C/a]c$ means substitution in c of au by $C[u]$.

Monadic $\lambda\mu$-calculus M$\lambda\mu$

Reduction rules:

$$
\begin{array}{rrcl}
(\beta) & (\lambda x.t)u & \rightarrow & [u/x]t \\
(\sigma) & \mathsf{bind}(\eta t, x.c) & \rightarrow & [t/x]c \\
(\pi_{\mathsf{bind}}) & \mathsf{bind}(\mu a.c, x.c') & \rightarrow & [\mathsf{bind}([\,], x.c')/a]c \\
(\pi_{\mathsf{covar}}) & b(\mu a.c) & \rightarrow & [b[\,]/a]c \\
(\eta_\mu) & \mu a.at & \rightarrow & t \quad (a \notin t) \\
(\eta_{\mathsf{bind}}) & \mathsf{bind}(t, x.a(\eta x)) & \rightarrow & at
\end{array}
$$

The reduction system is confluent and SN.

Relationship with Moggi's meta-language:

- The intuitionistic fragment of M$\lambda\mu$ arises by allowing only one co-variable.
- This fragment gives a variant of Moggi's meta-language where π_{bind} corresponds to an eager version of assoc.

Monadic $\lambda\mu$-calculus M$\lambda\mu$

Reduction rules:

$$
\begin{array}{rrcl}
(\beta) & (\lambda x.t)u & \to & [u/x]t \\
(\sigma) & \mathsf{bind}(\eta t, x.c) & \to & [t/x]c \\
(\pi_{\mathsf{bind}}) & \mathsf{bind}(\mu a.c, x.c') & \to & [\mathsf{bind}([\,], x.c')/a]c \\
(\pi_{\mathsf{covar}}) & b(\mu a.c) & \to & [b[\,]/a]c \\
(\eta_\mu) & \mu a.at & \to & t \quad (a \notin t) \\
(\eta_{\mathsf{bind}}) & \mathsf{bind}(t, x.a(\eta x)) & \to & at
\end{array}
$$

The reduction system is confluent and SN.

Relationship with Moggi's meta-language:

- The intuitionistic fragment of M$\lambda\mu$ arises by allowing only one co-variable.
- This fragment gives a variant of Moggi's meta-language where π_{bind} corresponds to an eager version of assoc.

Monadic $\lambda\mu$-calculus M$\lambda\mu$

Reduction rules:

$$
\begin{array}{rrcl}
(\beta) & (\lambda x.t)u & \rightarrow & [u/x]t \\
(\sigma) & \mathsf{bind}(\eta t, x.c) & \rightarrow & [t/x]c \\
(\pi_{\mathsf{bind}}) & \mathsf{bind}(\mu a.c, x.c') & \rightarrow & [\mathsf{bind}([\,], x.c')/a]c \\
(\pi_{\mathsf{covar}}) & b(\mu a.c) & \rightarrow & [b[\,]/a]c \\
(\eta_\mu) & \mu a.at & \rightarrow & t \quad (a \notin t) \\
(\eta_{\mathsf{bind}}) & \mathsf{bind}(t, x.a(\eta x)) & \rightarrow & at
\end{array}
$$

The reduction system is confluent and SN.

Relationship with Moggi's meta-language:

- The intuitionistic fragment of M$\lambda\mu$ arises by allowing only one co-variable.
- This fragment gives a variant of Moggi's meta-language where π_{bind} corresponds to an eager version of assoc.

Monadic translation $\overline{(.)} : \overline{\lambda}\mu\tilde{\mu} \to \mathsf{M}\lambda\mu$ (the cbn case)

Types: $\overline{A} = MA_*$, and $X_* = X$, $(A \supset B)_* = \overline{A} \supset \overline{B}$ (cf. $A_* \supset \overline{B}$ in cbv).

Expressions:

$$\begin{array}{rclrcl} \overline{y} & = & y & \overline{a} & = & a[\,] \\ \overline{\lambda y.t} & = & \eta(\lambda y.\overline{t}) & \overline{u :: e} & = & \mathsf{bind}([\,], f.\mathsf{bind}(\eta(\overline{u}), z.\overline{e}[fz])) \\ \overline{\mu a.c} & = & \mu a.\overline{c} & \overline{\tilde{\mu} y.c} & = & \mathsf{bind}(\eta([\,]), y.\overline{c}) \end{array}$$

$$\overline{\langle t|e\rangle} = \overline{e}[\overline{t}]$$

Preservation of typing:

$$\frac{\Gamma \vdash t : A|\Delta}{\overline{\Gamma} \vdash \overline{t} : \overline{A}|\overline{\Delta}} \quad \frac{\Gamma|e : A \vdash \Delta}{\overline{e}[y] : (\overline{\Gamma}, y : \overline{A} \vdash \overline{\Delta})} \quad \frac{c : (\Gamma \vdash \Delta)}{\overline{c} : (\overline{\Gamma} \vdash \overline{\Delta})}$$ are admissible.

Strict simulation of reduction:

If $t \to u$ in cbn $\overline{\lambda}\mu\tilde{\mu}$, then $\overline{t} \to^+ \overline{u}$ in $\mathsf{M}\lambda\mu$.

(Simulation is almost 1-1: only β in the source needs 2 steps in the target.)

Monadic translation $\overline{(.)} : \overline{\lambda}\mu\tilde{\mu} \to \mathsf{M}\lambda\mu$ (the cbn case)

Types: $\overline{A} = MA_*$, and $X_* = X$, $(A \supset B)_* = \overline{A} \supset \overline{B}$ (cf. $A_* \supset \overline{B}$ in cbv).

Expressions:

$$
\begin{array}{rclrcl}
\overline{y} & = & y & \overline{a} & = & a[\,] \\
\overline{\lambda y.t} & = & \eta(\lambda y.\overline{t}) & \overline{u :: e} & = & \mathrm{bind}([\,], f.\mathrm{bind}(\eta(\overline{u}), z.\overline{e}[fz])) \\
\overline{\mu a.c} & = & \mu a.\overline{c} & \overline{\tilde{\mu} y.c} & = & \mathrm{bind}(\eta([\,]), y.\overline{c})
\end{array}
$$

$$\overline{\langle t|e\rangle} = \overline{e}[\overline{t}]$$

Preservation of typing:

$$\frac{\Gamma \vdash t : A|\Delta}{\overline{\Gamma} \vdash \overline{t} : \overline{A}|\overline{\Delta}} \quad \frac{\Gamma|e : A \vdash \Delta}{\overline{e}[y] : (\overline{\Gamma}, y : \overline{A} \vdash \overline{\Delta})} \quad \frac{c : (\Gamma \vdash \Delta)}{\overline{c} : (\overline{\Gamma} \vdash \overline{\Delta})} \quad \text{are admissible.}$$

Strict simulation of reduction:

If $t \to u$ in cbn $\overline{\lambda}\mu\tilde{\mu}$, then $\overline{t} \to^+ \overline{u}$ in $\mathsf{M}\lambda\mu$.

(Simulation is almost 1-1: only β in the source needs 2 steps in the target.)

Monadic translation $\overline{(.)} : \overline{\lambda}\mu\tilde{\mu} \to \mathrm{M}\lambda\mu$ (the cbn case)

Types: $\overline{A} = MA_*$, and $X_* = X$, $(A \supset B)_* = \overline{A} \supset \overline{B}$ (cf. $A_* \supset \overline{B}$ in cbv).

Expressions:

$$\begin{array}{rclrcl} \overline{y} & = & y & \overline{a} & = & a[\,] \\ \overline{\lambda y.t} & = & \eta(\lambda y.\overline{t}) & \overline{u :: e} & = & \mathrm{bind}([\,], f.\mathrm{bind}(\eta(\overline{u}), z.\overline{e}[fz])) \\ \overline{\mu a.c} & = & \mu a.\overline{c} & \overline{\tilde{\mu}y.c} & = & \mathrm{bind}(\eta([\,]), y.\overline{c}) \end{array}$$

$$\overline{\langle t|e\rangle} = \overline{e}[\overline{t}]$$

Preservation of typing:

$$\frac{\Gamma \vdash t : A|\Delta}{\overline{\Gamma} \vdash \overline{t} : \overline{A}|\overline{\Delta}} \quad \frac{\Gamma|e : A \vdash \Delta}{\overline{e}[y] : (\overline{\Gamma}, y : \overline{A} \vdash \overline{\Delta})} \quad \frac{c : (\Gamma \vdash \Delta)}{\overline{c} : (\overline{\Gamma} \vdash \overline{\Delta})}$$ are admissible.

Strict simulation of reduction:

If $t \to u$ in cbn $\overline{\lambda}\mu\tilde{\mu}$, then $\overline{t} \to^+ \overline{u}$ in $\mathrm{M}\lambda\mu$.

(Simulation is almost 1-1: only β in the source needs 2 steps in the target.)

Monadic translation $\overline{(.)} : \overline{\lambda}\mu\tilde{\mu} \to \mathrm{M}\lambda\mu$ (the cbn case)

Types: $\overline{A} = MA_*$, and $X_* = X$, $(A \supset B)_* = \overline{A} \supset \overline{B}$ (cf. $A_* \supset \overline{B}$ in cbv).

Expressions:

$$\begin{array}{rcl} \overline{y} & = & y \\ \overline{\lambda y.t} & = & \eta(\lambda y.\overline{t}) \\ \overline{\mu a.c} & = & \mu a.\overline{c} \end{array} \qquad \begin{array}{rcl} \overline{a} & = & a[\,] \\ \overline{u :: e} & = & \mathrm{bind}([\,], f.\mathrm{bind}(\eta(\overline{u}), z.\overline{e}[fz])) \\ \overline{\tilde{\mu}y.c} & = & \mathrm{bind}(\eta([\,]), y.\overline{c}) \end{array}$$

$$\overline{\langle t|e\rangle} = \overline{e}[\overline{t}]$$

Preservation of typing:

$$\frac{\Gamma \vdash t : A|\Delta}{\overline{\Gamma} \vdash \overline{t} : \overline{A}|\overline{\Delta}} \quad \frac{\Gamma|e : A \vdash \Delta}{\overline{e}[y] : (\overline{\Gamma}, y : \overline{A} \vdash \overline{\Delta})} \quad \frac{c : (\Gamma \vdash \Delta)}{\overline{c} : (\overline{\Gamma} \vdash \overline{\Delta})}$$ are admissible.

Strict simulation of reduction:

If $t \to u$ in cbn $\overline{\lambda}\mu\tilde{\mu}$, then $\overline{t} \to^+ \overline{u}$ in $\mathrm{M}\lambda\mu$.

(Simulation is almost 1-1: only β in the source needs 2 steps in the target.)

Continuations-monad instantiation $(.)^\bullet : \mathrm{M}\lambda\mu \to \lambda$

Follows the usual term representation of the continuations-monad:

$$MA := \neg\neg A, \qquad \eta t := \lambda k.kt, \qquad \mathrm{bind}(t, x.u) := \lambda k.t(\lambda x.uk).$$

Expressions:

$$\begin{array}{rcl} x^\bullet & = & x \\ (\lambda x.t)^\bullet & = & \lambda x.t^\bullet \\ (tu)^\bullet & = & t^\bullet u^\bullet \\ (\mu a.c)^\bullet & = & \lambda k_a.c^\bullet \\ (\eta t)^\bullet & = & \lambda k.kt^\bullet \end{array} \qquad \begin{array}{rcl} (\mathrm{bind}(t, x.c))^\bullet & = & t^\bullet(\lambda x.c^\bullet) \\ (at)^\bullet & = & t^\bullet k_a \end{array}$$

(each a has a corresponding cont. var k_a)

Preservation of typing:

$$\frac{\Gamma \vdash t : A|\Delta}{\Gamma^\bullet, \neg\Delta^{-\bullet} \vdash t^\bullet : A^\bullet} \qquad \neg\Delta^{-\bullet} = \{k_a : \neg A^\bullet | a : MA \in \Delta\}$$

Strict simulation of reduction: If $t \to u$ in $\mathrm{M}\lambda\mu$,then $t^\bullet \to^+ u^\bullet$ in $\lambda[\beta\eta]$.
(η needed for simulating η_μ and η_{bind})

Corollary: $\mathrm{M}\lambda\mu$ is SN for typed terms.

Continuations-monad instantiation $(.)^\bullet : \mathsf{M}\lambda\mu \to \lambda$

Follows the usual term representation of the continuations-monad:

$$MA := \neg\neg A, \qquad \eta t := \lambda k.kt, \qquad \mathsf{bind}(t, x.u) := \lambda k.t(\lambda x.uk).$$

Expressions:

$$\begin{array}{rcl} x^\bullet & = & x \\ (\lambda x.t)^\bullet & = & \lambda x.t^\bullet \\ (tu)^\bullet & = & t^\bullet u^\bullet \\ (\mu a.c)^\bullet & = & \lambda k_a.c^\bullet \\ (\eta t)^\bullet & = & \lambda k.kt^\bullet \end{array} \qquad \begin{array}{rcl} (\mathsf{bind}(t, x.c))^\bullet & = & t^\bullet(\lambda x.c^\bullet) \\ (at)^\bullet & = & t^\bullet k_a \end{array}$$

(each a has a corresponding cont. var k_a)

Preservation of typing:

$$\frac{\Gamma \vdash t : A|\Delta}{\Gamma^\bullet, \neg\Delta^{-\bullet} \vdash t^\bullet : A^\bullet} \qquad \neg\Delta^{-\bullet} = \{k_a : \neg A^\bullet | a : MA \in \Delta\}$$

Strict simulation of reduction: If $t \to u$ in $\mathsf{M}\lambda\mu$,then $t^\bullet \to^+ u^\bullet$ in $\lambda[\beta\eta]$.

(η needed for simulating η_μ and η_{bind})

Corollary: $\mathsf{M}\lambda\mu$ is SN for typed terms.

cps translation of $\overline{\lambda}\mu\tilde{\mu}$

cps-translations of $\overline{\lambda}\mu\tilde{\mu}$ are obtained by composing the monadic translations with the continuations-monad instantiation:

$$\overline{\overline{(.)}} : \overline{\lambda}\mu\tilde{\mu} \stackrel{\overline{(.)}}{\to} \mathsf{M}\lambda\mu \stackrel{(.)^\bullet}{\to} \lambda$$

A closer look actually shows that simulation of the image of $\overline{(.)}$ needs no η-steps, and so $\lambda[\beta]$ is enough for strict simulation of $\overline{\lambda}\mu\tilde{\mu}$ via $\overline{\overline{(.)}}$.

Corollary: The cbn and cbv fragments of $\overline{\lambda}\mu\tilde{\mu}$ are SN for typed terms.

The cbn case

Types: $\overline{\overline{A}} = \neg\neg A^*$, $X^* = X$, $(A \supset B)^* = \overline{\overline{A}} \supset \overline{\overline{B}}$.

Expressions:

$$\begin{array}{rclcrcl} \overline{\overline{y}} & = & y & & \overline{\overline{a}} & = & [\,]k_a \\ \overline{\overline{\lambda y.t}} & = & \mathrm{Eta}(\lambda y.\overline{\overline{t}}) & & \overline{\overline{u :: e}} & = & [\,](\lambda f.\mathrm{Eta}(\overline{\overline{u}})(\lambda z.\overline{\overline{e}}[fz])) \\ \overline{\overline{\mu a.c}} & = & \lambda k_a.\overline{\overline{c}} & & \overline{\overline{\tilde{\mu} y.c}} & = & \mathrm{Eta}([\,])(\lambda y.\overline{\overline{c}}) \end{array}$$

$$\overline{\overline{\langle t|e\rangle}} = \overline{\overline{e}}[\overline{\overline{t}}] \qquad (\mathrm{Eta}(t) = \lambda k.kt)$$

cps translation of $\overline{\lambda}\mu\tilde{\mu}$

cps-translations of $\overline{\lambda}\mu\tilde{\mu}$ are obtained by composing the monadic translations with the continuations-monad instantiation:

$$\overline{\overline{(.)}} : \overline{\lambda}\mu\tilde{\mu} \stackrel{\overline{(.)}}{\rightarrow} \mathsf{M}\lambda\mu \stackrel{(.)^\bullet}{\rightarrow} \lambda$$

A closer look actually shows that simulation of the image of $\overline{(.)}$ needs no η-steps, and so $\lambda[\beta]$ is enough for strict simulation of $\overline{\lambda}\mu\tilde{\mu}$ via $\overline{\overline{(.)}}$.

Corollary: The cbn and cbv fragments of $\overline{\lambda}\mu\tilde{\mu}$ are SN for typed terms.

The cbn case

Types: $\overline{\overline{A}} = \neg\neg A^*$, $\quad X^* = X$, $\quad (A \supset B)^* = \overline{\overline{A}} \supset \overline{\overline{B}}$.

Expressions:

$$\begin{array}{rclrcl}
\overline{\overline{y}} & = & y & \overline{\overline{a}} & = & [\,]k_a \\
\overline{\overline{\lambda y.t}} & = & \mathsf{Eta}(\lambda y.\overline{\overline{t}}) & \overline{\overline{u :: e}} & = & [\,](\lambda f.\mathsf{Eta}(\overline{\overline{u}})(\lambda z.\overline{\overline{e}}[fz])) \\
\overline{\overline{\mu a.c}} & = & \lambda k_a.\overline{\overline{c}} & \overline{\overline{\tilde{\mu} y.c}} & = & \mathsf{Eta}([\,])(\lambda y.\overline{\overline{c}})
\end{array}$$

$$\overline{\overline{\langle t|e\rangle}} = \overline{\overline{e}}[\overline{\overline{t}}] \qquad (\mathsf{Eta}(t) = \lambda k.kt)$$

cps translation of $\overline{\lambda}\mu\tilde{\mu}$

cps-translations of $\overline{\lambda}\mu\tilde{\mu}$ are obtained by composing the monadic translations with the continuations-monad instantiation:

$$\overline{\overline{(.)}} : \overline{\lambda}\mu\tilde{\mu} \stackrel{\overline{(.)}}{\rightarrow} \mathsf{M}\lambda\mu \stackrel{(.)^\bullet}{\rightarrow} \lambda$$

A closer look actually shows that simulation of the image of $\overline{(.)}$ needs no η-steps, and so $\lambda[\beta]$ is enough for strict simulation of $\overline{\lambda}\mu\tilde{\mu}$ via $\overline{\overline{(.)}}$.

Corollary: The cbn and cbv fragments of $\overline{\lambda}\mu\tilde{\mu}$ are SN for typed terms.

The cbn case

Types: $\overline{\overline{A}} = \neg\neg A^*$, $\quad X^* = X$, $\quad (A \supset B)^* = \overline{\overline{A}} \supset \overline{\overline{B}}$.

Expressions:

$$\begin{array}{rcl@{\qquad}rcl}
\overline{\overline{y}} & = & y & \overline{\overline{a}} & = & [\,]k_a \\
\overline{\overline{\lambda y.t}} & = & \mathrm{Eta}(\lambda y.\overline{\overline{t}}) & \overline{\overline{u :: e}} & = & [\,](\lambda f.\mathrm{Eta}(\overline{\overline{u}})(\lambda z.\overline{\overline{e}}[fz])) \\
\overline{\overline{\mu a.c}} & = & \lambda k_a.\overline{\overline{c}} & \overline{\overline{\tilde{\mu} y.c}} & = & \mathrm{Eta}([\,])(\lambda y.\overline{\overline{c}})
\end{array}$$

$$\overline{\overline{\langle t|e\rangle}} = \overline{\overline{e}}[\overline{\overline{t}}] \qquad (\mathrm{Eta}(t) = \lambda k.kt)$$

cps translation of $\overline{\lambda}\mu\tilde{\mu}$

cps-translations of $\overline{\lambda}\mu\tilde{\mu}$ are obtained by composing the monadic translations with the continuations-monad instantiation:

$$\overline{\overline{(.)}} : \overline{\lambda}\mu\tilde{\mu} \stackrel{\overline{(.)}}{\to} \mathsf{M}\lambda\mu \stackrel{(.)^{\bullet}}{\to} \lambda$$

A closer look actually shows that simulation of the image of $\overline{(.)}$ needs no η-steps, and so $\lambda[\beta]$ is enough for strict simulation of $\overline{\lambda}\mu\tilde{\mu}$ via $\overline{\overline{(.)}}$.

Corollary: The cbn and cbv fragments of $\overline{\lambda}\mu\tilde{\mu}$ are SN for typed terms.

The cbn case

Types: $\overline{\overline{A}} = \neg\neg A^*, \qquad X^* = X, \qquad (A \supset B)^* = \overline{\overline{A}} \supset \overline{\overline{B}}.$

Expressions:

$$\begin{array}{rcl@{\qquad}rcl}
\overline{\overline{y}} & = & y & \overline{\overline{a}} & = & [\,]k_a \\
\overline{\overline{\lambda y.t}} & = & \mathsf{Eta}(\lambda y.\overline{\overline{t}}) & \overline{\overline{u :: e}} & = & [\,](\lambda f.\mathsf{Eta}(\overline{\overline{u}})(\lambda z.\overline{\overline{e}}[fz])) \\
\overline{\overline{\mu a.c}} & = & \lambda k_a.\overline{\overline{c}} & \overline{\overline{\tilde{\mu} y.c}} & = & \mathsf{Eta}([\,])(\lambda y.\overline{\overline{c}})
\end{array}$$

$$\overline{\overline{\langle t|e\rangle}} = \overline{\overline{e}}[\overline{\overline{t}}] \qquad\qquad (\mathsf{Eta}(t) = \lambda k.kt)$$

Extension to 2nd-order

The ideas before apply also to 2nd-order extensions of $\overline{\lambda}\mu\tilde{\mu}$, M$\lambda\mu$ and λ.

In particular, we find new SN results for the cbn and cbv fragments of 2nd-order $\overline{\lambda}\mu\tilde{\mu}$, inheriting SN of $\lambda 2$ via cps with strict simulation.

2nd-order $\overline{\lambda}\mu\tilde{\mu}$:

$$\frac{\Gamma \vdash t : B|\Delta}{\Gamma \vdash \Lambda X.t : \forall X.B|\Delta}\ (X \notin \Gamma,\Delta) \qquad \frac{\Gamma|e : [A/X]B \vdash \Delta}{\Gamma|A :: e : \forall X.B \vdash \Delta}$$

$$(\beta 2) \quad \langle \Lambda X.t|A :: e\rangle \rightarrow \langle [A/X]t|e\rangle$$

2nd-order M$\lambda\mu$:

$$\frac{\Gamma \vdash t : B|\Delta}{\Gamma \vdash \Lambda X.t : \forall X.B|\Delta}\ (X \notin \Gamma,\Delta) \qquad \frac{\Gamma \vdash t : \forall X.B|\Delta}{\Gamma \vdash tA : [A/X]B|\Delta}$$

$$(\beta 2) \quad (\Lambda X.t)A \rightarrow [A/X]t$$

Cbn cps-translation:

$$(\forall X.A)^* = \forall X.\overline{\overline{A}}, \qquad \overline{\Lambda X.t} = \mathrm{Eta}(\Lambda X.\overline{t}) \qquad \overline{A :: e} = [\,](\lambda z.\overline{e}[zA^*])$$

Extension to 2nd-order

The ideas before apply also to 2nd-order extensions of $\overline{\lambda}\mu\tilde{\mu}$, M$\lambda\mu$ and λ.

In particular, we find new SN results for the cbn and cbv fragments of 2nd-order $\overline{\lambda}\mu\tilde{\mu}$, inheriting SN of $\lambda 2$ via cps with strict simulation.

2nd-order $\overline{\lambda}\mu\tilde{\mu}$:

$$\frac{\Gamma \vdash t : B|\Delta}{\Gamma \vdash \Lambda X.t : \forall X.B|\Delta}\ (X \notin \Gamma, \Delta) \qquad \frac{\Gamma|e : [A/X]B \vdash \Delta}{\Gamma|A :: e : \forall X.B \vdash \Delta}$$

$$(\beta 2) \quad \langle \Lambda X.t|A :: e\rangle \rightarrow \langle [A/X]t|e\rangle$$

2nd-order M$\lambda\mu$:

$$\frac{\Gamma \vdash t : B|\Delta}{\Gamma \vdash \Lambda X.t : \forall X.B|\Delta}\ (X \notin \Gamma, \Delta) \qquad \frac{\Gamma \vdash t : \forall X.B|\Delta}{\Gamma \vdash tA : [A/X]B|\Delta}$$

$$(\beta 2) \quad (\Lambda X.t)A \rightarrow [A/X]t$$

Cbn cps-translation:

$$(\forall X.A)^* = \forall X.\overline{\overline{A}}, \qquad \overline{\overline{\Lambda X.t}} = \mathrm{Eta}(\Lambda X.\overline{\overline{t}}) \qquad \overline{\overline{A :: e}} = [\,](\lambda z.\overline{\overline{e}}[zA^*])$$

Extension to 2nd-order

The ideas before apply also to 2nd-order extensions of $\overline{\lambda}\mu\tilde{\mu}$, M$\lambda\mu$ and λ.

In particular, we find new SN results for the cbn and cbv fragments of 2nd-order $\overline{\lambda}\mu\tilde{\mu}$, inheriting SN of $\lambda 2$ via cps with strict simulation.

2nd-order $\overline{\lambda}\mu\tilde{\mu}$:

$$\frac{\Gamma \vdash t : B|\Delta}{\Gamma \vdash \Lambda X.t : \forall X.B|\Delta}\ (X \notin \Gamma,\Delta) \qquad \frac{\Gamma|e : [A/X]B \vdash \Delta}{\Gamma|A :: e : \forall X.B \vdash \Delta}$$

$$(\beta 2) \quad \langle \Lambda X.t|A :: e\rangle \rightarrow \langle [A/X]t|e\rangle$$

2nd-order M$\lambda\mu$:

$$\frac{\Gamma \vdash t : B|\Delta}{\Gamma \vdash \Lambda X.t : \forall X.B|\Delta}\ (X \notin \Gamma,\Delta) \qquad \frac{\Gamma \vdash t : \forall X.B|\Delta}{\Gamma \vdash tA : [A/X]B|\Delta}$$

$$(\beta 2) \quad (\Lambda X.t)A \rightarrow [A/X]t$$

Cbn cps-translation:

$$(\forall X.A)^* = \forall X.\overline{\overline{A}}, \qquad \overline{\overline{\Lambda X.t}} = \mathrm{Eta}(\Lambda X.\overline{\overline{t}}) \qquad \overline{\overline{A :: e}} = [\,](\lambda z.\overline{\overline{e}}[zA^*])$$

Extension to 2nd-order

The ideas before apply also to 2nd-order extensions of $\overline{\lambda}\mu\tilde{\mu}$, M$\lambda\mu$ and λ.

In particular, we find new SN results for the cbn and cbv fragments of 2nd-order $\overline{\lambda}\mu\tilde{\mu}$, inheriting SN of $\lambda 2$ via cps with strict simulation.

2nd-order $\overline{\lambda}\mu\tilde{\mu}$:

$$\frac{\Gamma \vdash t : B|\Delta}{\Gamma \vdash \Lambda X.t : \forall X.B|\Delta}\ (X \notin \Gamma,\Delta) \qquad \frac{\Gamma|e : [A/X]B \vdash \Delta}{\Gamma|A :: e : \forall X.B \vdash \Delta}$$

$$(\beta 2) \quad \langle \Lambda X.t|A :: e\rangle \longrightarrow \langle [A/X]t|e\rangle$$

2nd-order M$\lambda\mu$:

$$\frac{\Gamma \vdash t : B|\Delta}{\Gamma \vdash \Lambda X.t : \forall X.B|\Delta}\ (X \notin \Gamma,\Delta) \qquad \frac{\Gamma \vdash t : \forall X.B|\Delta}{\Gamma \vdash tA : [A/X]B|\Delta}$$

$$(\beta 2) \quad (\Lambda X.t)A \rightarrow [A/X]t$$

Cbn cps-translation:

$$(\forall X.A)^* = \forall X.\overline{\overline{A}}, \qquad \overline{\overline{\Lambda X.t}} = \mathrm{Eta}(\Lambda X.\overline{\overline{t}}) \qquad \overline{\overline{A :: e}} = [\,](\lambda z.\overline{\overline{e}}[zA^*])$$

Final remarks

- An elementary proof of SN for cbn/cbv $\overline{\lambda}\mu\tilde{\mu}$ via cps-translations is achieved.
- The cps-translations factor through a new classical monadic language.
- The technique easily extends to 2nd-order.
- Big improvement of our earlier results on intuitionistic sequent calculus (TYPES'08).
- Extend results, e.g. to other connectives, or to dependent types.
- Further study $\mathsf{M}\lambda\mu$ and ways to combine classical logic with monads.