Curry-Howard Correspondence for Classical Logic

Stéphane Graham-Lengrand
CNRS, Laboratoire d’Informatique de l’X

Stephane.Lengrand@Polytechnique.edu
Lecture II
Confluence?
Review from the previous lecture

Easy enough to introduce proof-terms to represent classical proofs
symmetry of classical logic = symmetry between programs and continuations
use of classical reasoning = control = programs can capture their continuations
Curien-Herbelin-Wadler - typing

\[
\begin{align*}
\Gamma, x : A & \vdash x : A \ ; \Delta \\
\Gamma, x : A & \vdash t : B \ ; \Delta \\
\Gamma & \vdash \lambda x . t : A \to B \ ; \Delta \\
\Gamma & \vdash t_1 : A_1 \ ; \Delta \ \Gamma & \vdash t_2 : A_2 \ ; \Delta \\
\Gamma & \vdash \langle t_1, t_2 \rangle : A_1 \land A_2 \ ; \Delta \\
\Gamma & \vdash t : A_i \ ; \Delta \\
\Gamma & \vdash \text{inj}_i (t) : A_1 \lor A_2 \ ; \Delta \\
\Gamma, x : A & \vdash c : \Delta \\
\Gamma & \vdash \mu \alpha . c : A \ ; \Delta \\
\Gamma & \vdash t : A \ ; \Delta \ \Gamma & \vdash e : A \vdash \Delta \\
\langle t \cdot e \rangle & : (\Gamma \vdash \Delta)
\end{align*}
\]
I. Reduction
Curien-Herbelin-Wadler - reduction

**Reduction**

\[(\rightarrow) \quad \langle \lambda x.t_1 \cdot t_2 :: e \rangle \rightarrow \langle t_2 \cdot \mu x.(t_1 \cdot e) \rangle\]

\[(\land) \quad \langle \langle t_1, t_2 \rangle \cdot \text{inj}_i(e) \rangle \rightarrow \langle t_i \cdot e \rangle\]

\[(\lor) \quad \langle \text{inj}_i(t) \cdot \langle e_1, e_2 \rangle \rangle \rightarrow \langle t \cdot e_i \rangle\]

\[\langle \mu \beta.c \cdot e \rangle \rightarrow \{e_\beta\} c\]

\[\langle t \cdot \mu x.c \rangle \rightarrow \{t_x\} c\]

**Theorem**: Subject Reduction

**OK**

**Theorem**: Progress?

**OK**

Cuts remaining in normal forms are of the form \(\langle x \cdot e \rangle\) and \(\langle t \cdot \alpha \rangle\), i.e. they represent contraction-left and contraction-right

**Theorem**: Normalisation?

**OK**

(Barbanera-Berardi’s symmetric reducibility candidates, see next lecture)

Symmetry of LK = Symmetry of terms vs. continuations. Now in the very syntax.
II. (Non-)confluence
Lafont’s example

\[
\text{Γ} \vdash \Delta, A, \text{Γ}' \vdash \Delta', \quad \text{Γ}', A \vdash \Delta' \\
\text{Γ, Γ'} \vdash \Delta, \Delta'
\]

Two ways to eliminate the cut:

\[
\text{Γ} \vdash \Delta \\
\text{Γ, Γ'} \vdash \Delta, \Delta'
\]

or

\[
\text{Γ}' \vdash \Delta' \\
\text{Γ, Γ'} \vdash \Delta, \Delta'
\]

but we could have the mix rule:

\[
\text{Γ} \vdash \Delta, \text{Γ}' \vdash \Delta' \\
\text{Γ, Γ'} \vdash \Delta, \Delta'
\]

Do we want this derivation as a normal proof?
Lafont’s example (in an additive world)

\[
\begin{align*}
\pi & \quad \pi' \\
\Gamma \vdash \Delta & \quad \Gamma \vdash \Delta \\
\Gamma \vdash \Delta, A & \quad \Gamma, A \vdash \Delta \\
\Gamma \vdash \Delta & \\
\end{align*}
\]

Two ways to eliminate the cut: \(\pi\) or \(\pi'\)

\[
\begin{align*}
\Gamma \vdash \Delta & \quad \Gamma \vdash \Delta \\
\end{align*}
\]

but we could have:

\[
\begin{align*}
\Gamma \vdash \Delta & \quad \Gamma \vdash \Delta \\
\Gamma \vdash \Delta & \\
\end{align*}
\]

Do we want this derivation as a normal proof?
More problematic example

\[
\begin{align*}
\vdash \pi & \\
\Gamma & \vdash \Delta, A, A & \Gamma, A, A & \vdash \Delta \\
\hline
\Gamma & \vdash \Delta, A & \Gamma, A & \vdash \Delta \\
\hline
\Gamma & \vdash \Delta
\end{align*}
\]

\[
\begin{align*}
\vdash \pi' & \\
\bullet & \bullet & \vdash \pi'(\cong) & \Gamma & \vdash \Delta \\
\vdash \pi' & \\
\bullet & \vdash \pi & \Gamma & \vdash \Delta \\
\hline
\text{or}
\end{align*}
\]

e.g.:
\[
(A \rightarrow B) \rightarrow A \vdash A \quad A, A \rightarrow C, A \rightarrow D \vdash C \land D
\]

\[
\hline
(A \rightarrow B) \rightarrow A, A \rightarrow C, A \rightarrow D \vdash C \land D
\]
In concrete terms

In Curien-Herbelin-Wadler’s calculus Curien and Herbelin [2000]; Wadler [2003], both examples appear as:

\[
\begin{align*}
\frac{c : (\Gamma \vdash \alpha : A, \Delta)}{\Gamma \vdash \mu \alpha . c : A ; \Delta} & \quad \frac{c' : (\Gamma, x : A \vdash \Delta)}{\Gamma ; \mu x . c' : A \vdash \Delta} \\
\frac{}{\langle \mu \alpha . c \bullet \mu x . c' \rangle : (\Gamma \vdash \Delta)}
\end{align*}
\]

\(\alpha\) (resp. \(x\)) could be used 0 (weakening), 1, or several (contraction) times in \(c\) (resp. \(c'\))

\[
\begin{align*}
\frac{c' : (\Gamma, x : A \vdash \Delta)}{c : (\Gamma \vdash \alpha : A, \Delta)} & \quad \frac{c : (\Gamma \vdash \alpha : A, \Delta)}{c' : (\Gamma, x : A \vdash \Delta)} \\
\frac{}{\{\mu x . c' / \alpha\} c : (\Gamma \vdash \Delta)} & \quad \frac{}{\{\mu \alpha . c / x\} c' : (\Gamma \vdash \Delta)}
\end{align*}
\]

(Dotted lines not inference rules, but properties of typing system)
In conclusion

Easy enough to give rewrite system on proof-terms to represent cut-elimination, system follows the intuitions of continuations and control.

Gives non-confluent calculus because cut-elimination is non-confluent in classical logic because programs and continuations fight for the control of computation.

This makes it very hard to give a semantics of classical proofs / typed proof-terms.

Today’s challenge: Find a way to construct a denotational semantics.
Back to the main issue

Remember that a CCC with $\neg \neg A \simeq A$ collapses.

3 ways to get away:
1. Break the symmetry between $\land$ and $\lor$
2. Break the cartesian product (Dosen-Petric et al.)
3. Break the curryfication (Strassburger et al.)

In this course: Break the symmetry between $\land$ and $\lor$

Why? Only one among the three for which computational interpretations of cut-elimination are reasonably well-understood
Breaking the \( \land \lor \) symmetry by the CBN/CBV approach

\[
\begin{align*}
\vdash \pi & \quad \vdash \pi' \\
\Gamma \vdash \Delta, A & \quad \Gamma, A \vdash \Delta \\
\hline \\
\Gamma \vdash \Delta
\end{align*}
\]

Give systematic priority to

- the right (push \( \pi \) into \( \pi' \))
- or to the left (push \( \pi' \) into \( \pi \))

1. Both solutions make the calculus confluent.

2. Suggests to construct 2 denotational semantics \([c]_N\) and \([c]_V\) with the hope that:

\[
\begin{align*}
[c_0]_N & = [c_1]_N \quad \text{iff} \quad "c_0 \leftrightarrow^* c_1 \text{ with systematic priority to the right}"
\\
[c_0]_V & = [c_1]_V \quad \text{iff} \quad "c_0 \leftrightarrow^* c_1 \text{ with systematic priority to the left}" \\
\end{align*}
\]

3. Relates to the notions of \textit{Call-by-name} and \textit{Call-by-value}

- Plotkin \cite{Plotkin1975} CBV/CBN
- Moggi \cite{Moggi1989} monadic \( \lambda \)-calculus
III. From programming languages to rewriting theory
Call-by-name and call-by-value

proc MyFavoriteFunction(x) {
  ... x ... 
}

... 

MyFavoriteFunction(A)

Should A be evaluated before entering the function (CBV) or when it is used (CBN)?

... depends on the compiler

... may depend on datatype (base types may have different behaviour)

In presence of side-effects, knowing which of the two the compiler implements, is vital
How to evaluate a functional program?

Evaluation should produce values.  

In functional programming, *functions are values* (e.g. can be given as arguments)

⇒ No need to reduce them.
\(\lambda\)-calculus: a core functional language vs. a theory of functions

...equipped with an operational semantics (close to implementation)

...which can be expressed by an evaluation strategy that selects a unique \(\beta\)-redex to reduce:

- Never reduce a \(\lambda\)-abstraction, as it is a “value” (this is called weak reduction)
- Always reduce \(M\) first in an application \(MN\). Then:
  - If \(M\) is an abstraction: reduce the \(\beta\)-redex first (CBN) reduce \(N\) first (CBV)
  - Otherwise, reduce \(N\) (never happens with closed terms)

Strategies denoted \(\rightarrow_{\text{CBN}}\) and \(\rightarrow_{\text{CBV}}\)

Neither is “better” than the other -cf. Haskell (CBN) vs. Caml (CBV)
**λ-calculus: a core functional language vs. a theory of functions**

...equipped with a **denotational semantics** (close to mathematical functions)

...where equalities are congruences (e.g. if $M = N$ then $\lambda x. M = \lambda x. N$)

and reductions are congruences (this is called **strong reduction**)

Formally, in λ-calculus:

**values:** $\lambda x. M$ and $x$  (denoted $V$...)

**not values:** $MN$

Why? because by evaluating $MN$, you may get something completely different

In this view,

“Call-by-name” = general $\beta$-reduction

$$(\lambda x. M) N \rightarrow_\beta \{N/x\} M$$

“Call-by-value” = restriction to arguments being values

$$(\lambda x. M) V \rightarrow_{\beta_v} \{V/x\} M$$
Question:
CBN: Is there a relation between $\rightarrow_{\text{CBN}}$ and $\rightarrow_{\beta}$?
CBV: Is there a relation between $\rightarrow_{\text{CBV}}$ and $\rightarrow_{\beta_v}$?

Answer:
Clearly, $\rightarrow_{\text{CBN}} \subseteq \rightarrow_{\beta}$ and $\rightarrow_{\text{CBV}} \subseteq \rightarrow_{\beta_v}$

What about the other way round?
\textbf{\textit{\lambda}-calculus: a core functional language vs. a theory of functions}

Bridge between weak and strong reductions = Plotkin’s result Plotkin [1975]:

CBN: $\longrightarrow^* \beta$ is the closure of $\longrightarrow^*_{\text{CBN}}$ under

$$
\frac{M_1 \longrightarrow^*_{\text{CBN}} C[M_2]}{M_1 \longrightarrow C[M_3]}
$$

CBV: $\longrightarrow^* \beta_v$ is the closure of $\longrightarrow^*_{\text{CBV}}$ under

$$
\frac{M_1 \longrightarrow^*_{\text{CBV}} C[M_2]}{M_1 \longrightarrow C[M_3]}
$$

What’s the point?

This result allows us to call CBN and CBV

not some operational semantics of some functional programming language

but some rewriting theories in \textit{\lambda}-calculus.
IV. The comeback of continuations
Compiling with continuations

CBN/CBV= question of compilation

\(\lambda\)-calculus can be compiled into (a fragment of) itself!
called Continuation Passing Style (CPS)-translation

\[\begin{align*}
    x &:= \lambda k.x \ k \\
    \lambda x.M &:= \lambda k.(k (\lambda x.M)) \\
    M N &:= \lambda k.M (\lambda y.y N k)
\end{align*}\]

\[\begin{align*}
    \overline{x} &:= \lambda k.k \ x \\
    \overline{\lambda x.M} &:= \lambda k.(k (\lambda x.\lambda k'.\overline{M} \ k')) \\
    \overline{M N} &:= \lambda k.\overline{M} (\lambda y.\overline{N} (\lambda z.y z k))
\end{align*}\]

What’s the point? Look, arguments are always values!

\(\Rightarrow\) CPS-evaluation (i.e. evaluation of the CPS-translated term) is strategy-indifferent

(\(\rightarrow_\beta = \rightarrow_{\beta_v}\) for the translated terms)
The CPS-translations preserve reductions

**Theorem** (Simulations - soundness)

CBN If $M \rightarrow_{\beta} N$ then $\overline{M} \rightarrow^{*}_{\beta} \overline{N}$

CBV If $M \rightarrow_{\beta_v} N$ then $\overline{M} \rightarrow^{*}_{\beta} \overline{N}$

**Theorem** (Simulations - completeness)

CBN If $\overline{M} \leftrightarrow^{*}_{\beta} \overline{N}$ then $M \leftrightarrow^{*}_{\beta} N$

CBV Not the case for CBV! (unless extended - Moggi Moggi [1989])
The CPS-translations preserve types

We deal here with simple types

\[ A, B, \ldots ::= \alpha \mid A \rightarrow B \]

Assume \( \Gamma \vdash M : A \). Do we have:

\( \Gamma' \vdash M : A' \) (for some \( \Gamma', A' \))? \( \Gamma'' \vdash \overline{M} : A'' \) (for some \( \Gamma'', A'' \))?

CPS-translations reveal 2 classes of terms in the target: *values* & *continuations* (like \( k \))

The types of values and continuations in the translated terms depend on CBN or CBV:
We choose or we add a particular atomic type \( R \), an abstract type of *responses*, then

CBN

\[
\begin{align*}
\alpha & ::= \alpha \\
A \rightarrow B & ::= ((A \rightarrow R) \rightarrow R) \rightarrow (B \rightarrow R) \rightarrow R
\end{align*}
\]

CBV

\[
\begin{align*}
\bar{\alpha} & ::= \alpha \\
\bar{A} \rightarrow \bar{B} & ::= \bar{A} \rightarrow (\bar{B} \rightarrow R) \rightarrow R
\end{align*}
\]

**Theorem** : Preservation of typing

If \( \Gamma \vdash M : A \) then

\( (\Gamma \rightarrow R) \rightarrow R \vdash M : (A \rightarrow R) \rightarrow R \)

If \( \Gamma \vdash M : A \) then

\( \bar{\Gamma} \vdash \bar{M} : (\bar{A} \rightarrow R) \rightarrow R \)
Variants

Fischer's translation for CBV Fischer [1972]

\[
\begin{align*}
\overline{x} & := \lambda k. k \ x \\
\overline{\lambda x.M} & := \lambda k. (k \ (\lambda k'. \overline{\lambda x.M} \ k')) \\
\overline{M \ N} & := \lambda k. \overline{M} \ (\lambda y. \overline{N} \ (\lambda z. y \ k \ z))
\end{align*}
\]

Hofmann & Streicher's translation for CBN Hofmann and Streicher [1997]. using product types

\[
\begin{align*}
\overline{x} & := \lambda k. x \ k \\
\overline{\lambda x.M} & := \lambda (x, k). \overline{M} \ k \\
\overline{M \ N} & := \lambda k. \overline{M} \ (\lambda y. \overline{N} \ k)
\end{align*}
\]

\[
\begin{align*}
\overline{\alpha} & := \alpha \\
\overline{A \rightarrow B} & := (\overline{B} \rightarrow \overline{R}) \rightarrow (\overline{A} \rightarrow \overline{R}) \\
\overline{\alpha \rightarrow R} & := \alpha \rightarrow \overline{R} \\
\overline{A \rightarrow B} & := (\overline{A} \rightarrow \overline{R}) \times \overline{B}
\end{align*}
\]

Theorem: If \( \Gamma \vdash M : A \) then \( \overline{\Gamma} \rightarrow \overline{R} \vdash \overline{M} : \overline{A} \rightarrow \overline{R} \)
CPS-translations and categorical semantics

Remember: simple-typed $\lambda$-terms have a semantics in a Cartesian Closed Category

CPS-translations compile the simply-typed $\lambda$-calculus into itself in a semantically meaningful way:

We can now assign to a simply-typed $\lambda$-term $M$, the semantics (in a CCC) of $M$ or $\overline{M}$ (semantics now depends on CBN/CBV).

By the simulation theorem, reductions are sound w.r.t. that semantics.

CPS-Fragment $\Rightarrow$ we need less than a CCC:

Exponentials just of the form $R^A \Rightarrow \textit{Response Category}$.

Sub-cat of the objects of that form: $\textit{Continuation category} = \text{CCC} + \text{rich structure}$ also called $\textit{Control Category}$ (Selinger Selinger [2001])

Useful for classical logic.
V. Classical logic and CBN/CBV
Translating classical logic into intuitionistic logic

Turning $P$ into $P'$ by adding (enough) double negations, you get

If $\vdash_c P$ then $\vdash_i P'$.

Obviously, $\vdash_c P \leftrightarrow P'$.

$\neg\neg$-translation, Goedel’s A-translation, . . .

$$\begin{align*}
\alpha^* & := \alpha \\
(A \rightarrow B)^* & := (((A^* \rightarrow \bot) \rightarrow \bot) \rightarrow (((B^* \rightarrow \bot) \rightarrow \bot))
\end{align*}$$

$$\begin{align*}
\alpha^* & := \alpha \\
(A \rightarrow B)^* & := A^* \rightarrow (B^* \rightarrow \bot) \rightarrow \bot
\end{align*}$$

Having selected a response type $\bot$, a continuation is a proof of negation
Why such a fuss about intuitionistic vs. classical, then?

If it suffices to add negations in a classical provable formula, are the two logics really different?

Yes. Adding negations breaks nice properties of intuitionistic logic:

In intuitionistic logic:
If \( \vdash A_1 \lor A_2 \) then either \( \vdash A_1 \) or \( \vdash A_2 \).
If \( \vdash \exists x A \) then there is \( t \) such that \( \vdash \{t/x\} A \)

Getting \( t \) from the proof of \( \vdash \exists x A \) = Witness extraction
Also true in some theories, like arithmetics (Heyting arithmetics):
If \( HA \vdash \exists x A \) then there is \( t \) such that \( HA \vdash \{t/x\} A \)

Cannot say anything when \( \vdash \neg\neg(A_1 \lor A_2) \) or \( \vdash \neg\neg \exists x A \)

What to do with a classical proof of \( \vdash \exists x A \)?

If \( A \) is nice enough, Classical witness extraction.
Reminder: classical proof-terms Curien and Herbelin [2000]; Wadler [2003]

terms \[ t ::= x \mid \mu \beta. c \mid \lambda x.t \mid \langle t_1, t_2 \rangle \mid \text{inj}_i(t) \]

continuations \[ e ::= \alpha \mid \mu x.c \mid t :: e \mid \langle e_1, e_2 \rangle \mid \text{inj}_i(e) \]

commands \[ c ::= \langle t \cdot e \rangle \]

\[
\begin{align*}
(\rightarrow) & \quad \langle \lambda x.t_1 \cdot t_2 :: e \rangle \to \langle t_2 \cdot \mu x. \langle t_1 \cdot e \rangle \rangle \\
(\land) & \quad \langle \langle t_1, t_2 \rangle \cdot \text{inj}_i(e) \rangle \to \langle t_i \cdot e \rangle \\
(\lor) & \quad \langle \text{inj}_i(t) \cdot \langle e_1, e_2 \rangle \rangle \to \langle t \cdot e_i \rangle \\
\langle \mu \beta . c \cdot e \rangle & \to \{^e_\beta \} c \\
\langle t \cdot \mu x . c \rangle & \to \{^t_x \} c
\end{align*}
\]
CBN and CBV for classical proof-terms Curien and Herbelin [2000]; Wadler [2003]

term values

\[ V ::= x \mid \lambda x.t \mid \langle V_1, V_2 \rangle \mid \text{inj}_i(V) \]

continuation values

\[ E ::= \alpha \mid t :: E \mid \langle E_1, E_2 \rangle \mid \text{inj}_i(E) \]

\[
\begin{align*}
\langle \lambda x.t_1 \bullet t_2 :: E \rangle & \to \langle t_2 \bullet \mu x.\langle t_1 \bullet E \rangle \rangle \\
\langle \langle t_1, t_2 \rangle \bullet \text{inj}_i(E) \rangle & \to \langle t_i \bullet E \rangle \\
\langle \text{inj}_i(t) \bullet \langle E_1, E_2 \rangle \rangle & \to \langle t \bullet E_i \rangle \\
\langle \mu \beta.c \bullet E \rangle & \to \{ E/\beta \} c \\
\langle t \bullet \mu x.c \rangle & \to \{ V/\alpha \} c
\end{align*}
\]

CBV

\[
\begin{align*}
\langle \lambda x.t \bullet V :: e \rangle & \to \langle V \bullet \mu x.\langle t \bullet e \rangle \rangle \\
\langle \langle V_1, V_2 \rangle \bullet \text{inj}_i(e) \rangle & \to \langle V_i \bullet e \rangle \\
\langle \text{inj}_i(V) \bullet \langle e_1, e_2 \rangle \rangle & \to \langle V \bullet e_i \rangle \\
\langle \mu \beta.c \bullet e \rangle & \to \{ e/\beta \} c \\
\langle V \bullet \mu x.c \rangle & \to \{ V/\alpha \} c
\end{align*}
\]

plus some focussing rules to ensure progress.

The two reduction relations now denoted \( \rightarrow \text{CBN} \) and \( \rightarrow \text{CBV} \).

Theorem \( \rightarrow \text{CBN} \) and \( \rightarrow \text{CBV} \) are confluent
CPS-translations of classical proof-terms Curien and Herbelin [2000];
Wadler [2003]

It is possible to define CPS-translations of terms, continuations, and commands:

CBN  \[ \frac{t \quad e \quad c}{t} \]  
CBV  \[ \frac{\bar{t} \quad \bar{e} \quad \bar{c}}{t} \]

**Theorem** (Preservation of reduction)

CBN If \( c_1 \xrightarrow{\text{CBN}} c_2 \) then \( c_1 \xrightarrow{\beta}^\ast c_2 \)

CBV If \( c_1 \xrightarrow{\text{CBV}} c_2 \) then \( \bar{c_1} \xrightarrow{\beta}^\ast \bar{c_2} \)

\[ \Gamma \vdash t : A ; \Delta \]

**Theorem** (Preservation of typing) If \( \Gamma ; e : A \vdash \Delta \) then \( c : (\Gamma \vdash \Delta) \)

\[ \bar{\Gamma} \rightsquigarrow R, \bar{\Delta} \vdash \bar{t} : \bar{A} \rightarrow R \]
\[ \bar{\Gamma} \rightsquigarrow R, \bar{\Delta} \vdash \bar{e} : (\bar{A} \rightarrow R) \rightarrow R \]
\[ \bar{\Gamma} \rightsquigarrow R, \bar{\Delta} \vdash \bar{c} : R \]

Using Hofman-Streicher Hofmann and Streicher [1997]

Using Fischer Fischer [1972]
Categorical semantics

Define $[c]_v := [\overline{c}]$ and $[c]_N := [c]$

where $[t]$ is the semantics, in a response category, of a $\lambda$-term $t$ in the CPS-fragment

Assume $c : (x_1 : A_1, \ldots, x_n : A_n \vdash \alpha_1 : B_1, \ldots, \alpha_m : B_m)$

Remember that a control category is the sub-category of a response category $C$ whose objects are in $\{ R^A | A \in C \}$

CBN Write $K_A$ for the object corresponding to $\overline{A}$, and $C_A$ for $R^{K_A}$,

$$[c]_N : \begin{cases} \left( C_{A_1} \times \ldots \times C_{A_n} \times K_{B_1} \times \ldots \times K_{B_m} \right) \to R & \text{in a response category} \\ C_{A_1} \times \ldots \times C_{A_n} \to C_{B_1} \& \ldots \& C_{B_m} & \text{in a control category} \end{cases}$$

CBV Write $V_A$ for the object corresponding to $\overline{A}$, $K_A$ for $R^{V_A}$ and $C_A$ for $R^{K_A}$,

$$[c]_v : \begin{cases} \left( V_{A_1} \times \ldots \times V_{A_n} \times K_{B_1} \times \ldots \times K_{B_m} \right) \to R & \text{in a response category} \\ K_{B_1} \times \ldots \times K_{B_m} \to K_{A_1} \& \ldots \& K_{A_n} & \text{in a control category} \\ K_{A_1} \otimes \ldots \otimes K_{A_n} \to K_{B_1} + \ldots + K_{B_m} & \text{in a co-control category} \end{cases}$$

(Where $\otimes$ is the dual of $\&$)
Semantics for classical proofs: the historical point of view

Here we see that

$\text{CBN} = \text{Control categories},$

$\text{CBV} = \text{Co-Control categories}.$

The semantics validate the reductions

If $c \longrightarrow_{\text{CBN}} c'$ then $[c]_N = [c']_N$

If $c \longrightarrow_{\text{CBV}} c'$ then $[c]_N = [c']_V$

Today’s goal is achieved

…by breaking the symmetry between $\land$ and $\lor$:

$\neg$ is not the dual of $\times!!$

(equivalently, $\lor$ is not the dual of $\otimes$)

Due to Selinger Selinger [2001].

Comes from preliminary works:

- De Groote, Barbanera, Berardi, Ong,…
- Hofmann, Streicher, Reus Hofmann and Streicher [1997]; Streicher and Reus [1998]

Semantics of continuations.

Question of Duality CBV/CBN (in $\lambda\mu$) is conjectured.
Many variants have been studied

- variants of Parigot's $\lambda\mu$ have different properties with respect to separation. **Delimited control** (Saurin, Herbelin, etc)
- Lots of open issues on extensionality, observational equivalence and separation, $\eta$-conversion, etc...
- Classical calculi and focusing: Zeilberger, Herbelin, Munch, Houtmann.
Questions?
References


P. Selinger. Control categories and duality: on the categorical semantics of the $\lambda\mu$-calculus. 
