How delimited continuations can be used to define dependently typed CPS

Étienne Miquey

Équipe Gallinette, INRIA
LS2N, Université de Nantes

TYPES 2018

Dependent types

Lot of features:

- for *programmers*
- for *logicians*
- for *proof-assisted* people
- for *proof-assisting* people

Ingredients:

- dependent product:

$$\frac{\Gamma, a : A \vdash p : B}{\Gamma \vdash \lambda a.p : \Pi(x : A).B} \qquad \frac{\Gamma \vdash p : \Pi(a : A).B[a] \quad \Gamma \vdash t : A}{\Gamma \vdash pt : B[t]}$$

- dependent sum:

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash p : B[t/x]}{\Gamma \vdash (t,p) : \Sigma(x : A).B} \qquad \frac{\Gamma \vdash p : \Sigma(x : A).B}{\Gamma \vdash \mathtt{wit}\ p : A} \qquad \frac{\Gamma \vdash p : \Sigma(x : A).B}{\Gamma \vdash \mathtt{prf}\ p : B(\mathtt{wit}\ p)}$$

CPS, classical logic & sequent calculi

CPS:

	Call-by-name	Call-by-value
$[\![tu]\!]$	$\lambda k.[\![t]\!]\,[\![u]\!]\,k$	$\lambda k.[\![u]\!]\,(\lambda v.[\![t]\!]\,v\,k)$
$[\![A \to B]\!]$	$(\neg\neg A) \to \neg\neg B$	$A \to \neg\neg B$

Provides semantics for control operators:

$$[\![\mathtt{call/cc}_\alpha\ t]\!] \triangleq \lambda\alpha.[\![t]\!]\alpha \qquad\qquad [\![\mathtt{throw}_\alpha\ t]\!] \triangleq \lambda\_.[\![t]\!]\alpha$$

Factorizes through sequent calculus:

$$\begin{tikzcd}
\boxed{\lambda\text{-calculus CbN}} \arrow[r, "\text{embed.}"] & \boxed{\lambda\mu\tilde{\mu}\text{-calculus CbN}} \arrow[r, "\text{CPS}"] & \boxed{\lambda\text{-calculus}} \\
\boxed{\lambda\text{-calculus CbV}} \arrow[r, "\text{embed.}"] & \boxed{\lambda\mu\tilde{\mu}\text{-calculus CbV}} \arrow[r, "\text{CPS}"] & \boxed{\lambda\text{-calculus}}
\end{tikzcd}$$

CPS, classical logic & sequent calculi

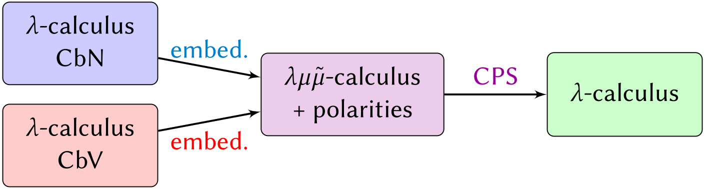
CPS:

	Call-by-name	Call-by-value
$[\![tu]\!]$	$\lambda k.[\![t]\!]\,[\![u]\!]\,k$	$\lambda k.[\![u]\!]\,(\lambda v.[\![t]\!]\,v\,k)$
$[\![A \to B]\!]$	$(\neg\neg A) \to \neg\neg B$	$A \to \neg\neg B$

Provides semantics for control operators:

$$[\![\mathtt{call/cc}_\alpha\, t]\!] \triangleq \lambda\alpha.[\![t]\!]\alpha \qquad\qquad [\![\mathtt{throw}_\alpha\, t]\!] \triangleq \lambda\_.[\![t]\!]\alpha$$

Factorizes through sequent calculus:



Dependent Types & Classical Logic

On the Degeneracy of Σ-Types in Presence of Computational Classical Logic

H. Herbelin, TLCA 2005

Abstract. We show that a minimal dependent type theory based on Σ-types and equality is degenerated in presence of computational classical logic. By computational classical logic is meant a classical logic derived from a control operator equipped with reduction rules similar to the ones of Felleisen's $\mathcal{C}$ or Parigot's μ operators. As a consequence, formalisms such as Martin-Löf's type theory or the (`Set`-predicative variant of the) Calculus of Inductive Constructions are inconsistent in presence of computational classical logic. Besides, an analysis of the role of the η-rule for control operators through a set-theoretic model of computational classical logic is given.

Dependent Types & Classical Logic

Computational classical logic:

- $\texttt{call/cc}_\alpha$ captures the *current continuation*
- $\texttt{throw}_\alpha$ replaces the *current continuation* by α

Paradox:

One can define:

$$H_0 := \texttt{call/cc}_\alpha(1, \texttt{throw}_\alpha(0, \texttt{refl})) : \Sigma(x:\mathbb{N}).x = 0$$

and reach a contradiction:

$$(\texttt{wit}\, H_0, \texttt{prf}\, H_0) \;\to\; \underbrace{(1, \overbrace{\texttt{refl}}^{0=0})}_{\cancel{\Sigma(x:\mathbb{N}).x=0}}$$

Morality:

↪ need to **restrict** dependencies to "*not too effectful*" computations

Dependent types & CPS

CPS Translating Inductive and Coinductive Types

G. Barthe & T. Uustalu, PEPM 2002

ABSTRACT

We investigate CPS translatability of typed λ-calculi with inductive and coinductive types. We show that tenable Plotkin-style call-by-name CPS translations exist for simply typed λ-calculi with a natural number type and stream types and, more generally, with arbitrary positive inductive and coinductive types. These translations also work in the presence of control operators and generalize for dependently typed calculi where case-like eliminations are only allowed in non-dependent forms. No translation is possible along the same lines for small Σ-types and sum types with dependent case.

Dependent types & CPS

Questions:

- Is it *really* impossible?
- Is the problem limited to *call-by-name*?
- Is the problem limited to Σ*-types*?
- What about *value restriction*?
- Can't we get a *dependent sequent calculus?*

Dependent types & CPS

Questions:

- Is it *really* impossible?
- Is the problem limited to *call-by-name*?
- Is the problem limited to Σ*-types*?
- What about *value restriction*?
- Can't we get a *dependent sequent calculus?*

No it's not:

- Bowman *et al.* [POPL 2018]:

parametric answer-types + *extensional type theory*

- M. [ESOP 2017]:

parametric and dependent answer-types + *delimited continuations*

- Cong, Asai [ICFP 2018]

Dependently typed CPS

Sequent calculus | Call-by-value | Π-type

Can this work?

$$\dfrac{\dfrac{\begin{array}{c}\Pi_p\\ \vdots\\ \Gamma, a : A \vdash p : B[a] \mid \Delta\end{array}}{\Gamma \vdash \lambda a.p : \Pi(a : A).B \mid \Delta}\,(\to_r) \qquad \dfrac{\begin{array}{c}\Pi_q\\ \vdots\\ \Gamma \vdash q : A \mid \Delta\end{array} \quad \begin{array}{c}\Pi_e\\ \vdots\\ \Gamma \mid e : B[q] \vdash \Delta\end{array} \quad q \in V}{\Gamma \mid q \cdot e : \Pi(a : A).B \vdash \Delta}\,(\to_l)}{\langle \lambda a.p \,\|\, q \cdot e \rangle : (\Gamma \vdash \Delta)}\,(\textsc{Cut})$$

$$\longrightarrow$$

Sequent calculus | Call-by-value | Π-type

Can this work?

$$\dfrac{\dfrac{\begin{matrix}\Pi_p\\ \vdots\\ \Gamma, a : A \vdash p : B[a] \mid \Delta\end{matrix}}{\Gamma \vdash \lambda a.p : \Pi(a:A).B \mid \Delta}\,(\to_r) \qquad \dfrac{\begin{matrix}\Pi_q\\ \vdots\\ \Gamma \vdash q : A \mid \Delta\end{matrix} \quad \begin{matrix}\Pi_e\\ \vdots\\ \Gamma \mid e : B[q] \vdash \Delta\end{matrix} \quad q \in V}{\Gamma \mid q \cdot e : \Pi(a:A).B \vdash \Delta}\,(\to_l)}{\color{blue}{\langle \lambda a.p \parallel q \cdot e\rangle} : (\Gamma \vdash \Delta)}\,(\textsc{Cut})$$

$$\color{blue}{\longrightarrow}$$

$$\dfrac{\begin{matrix}\Pi_q\\ \vdots\\ \Gamma \vdash q : A \mid \Delta\end{matrix} \qquad \dfrac{\dfrac{\Gamma, a : A \vdash p : \cancel{B[a]} \mid \Delta \qquad \Gamma, a : A \mid e : \cancel{B[q]} \vdash \Delta}{\langle p \parallel e\rangle : (\Gamma, a : A \vdash \Delta)}\ \mathit{Mismatch}}{\Gamma \mid \tilde\mu a.\langle p \parallel e\rangle : A \vdash \Delta}\,(\tilde\mu)}{\color{blue}{\langle q \parallel \tilde\mu a.\langle p \parallel e\rangle\rangle} : (\Gamma \vdash \Delta)}\,(\textsc{Cut})$$

Sequent calculus | Call-by-value | Π-type

Can this work? ✓

$$\dfrac{\dfrac{\overset{\Pi_p}{\vdots}}{\dfrac{\Gamma, a : A \vdash p : B[a] \mid \Delta}{\Gamma \vdash \lambda a.p : \Pi(a:A).B \mid \Delta}}(\rightarrow_r) \qquad \dfrac{\overset{\Pi_q}{\vdots}\quad\overset{\Pi_e}{\vdots}}{\dfrac{\Gamma \vdash q : A \mid \Delta \quad \Gamma \mid e : B[q] \vdash \Delta \quad q \in V}{\Gamma \mid q \cdot e : \Pi(a:A).B \vdash \Delta}}(\rightarrow_l)}{\langle \lambda a.p \parallel q \cdot e \rangle : (\Gamma \vdash \Delta)}(\textsc{Cut})$$

$$\longrightarrow$$

$$\dfrac{\dfrac{\Pi_q}{\vdots}\;\; \Gamma \vdash q : A \mid \Delta \qquad \dfrac{\dfrac{\Gamma, a : A \vdash p : B[a] \mid \Delta \quad \Gamma, a : A \mid e : B[q] \vdash \Delta; \{\cdot|p\}\{a|q\}}{\langle p \parallel e \rangle : \Gamma, a : A \vdash \Delta; \{a|q\}}(\textsc{Cut})}{\Gamma \mid \tilde{\mu}a.\langle p \parallel e\rangle : A \vdash \Delta; \{.|q\}}(\tilde{\mu})}{\langle q \parallel \tilde{\mu}a.\langle p \parallel e \rangle\rangle : (\Gamma \vdash \Delta); \{\cdot|\cdot\}}(\textsc{Cut})$$

CPS | Call-by-value | Π-type (1/2)

Is it enough?

- subject reduction / normalization / consistency as a logic ✓
- suitable for CPS translation ✗

$$[\![q]\!]\,[\![\tilde{\mu}a.\langle p \parallel e\rangle]\!] = \underbrace{[\![q]\!]}_{\neg\neg A}\,(\lambda a.\,\underbrace{[\![p]\!]}_{\neg\neg B(a)}\,\underbrace{[\![e]\!]}_{\neg B(q)})$$

This is quite normal:

- we observed a desynchronization
- we compensated only within the type system

↪ *we need to do this within the calculus!*

Intuition: $[\![p]\!]$ *shouldn't be applied to* $[\![e]\!]$ *before* $[\![q]\!]$ *has reduced*

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e \rangle]\!] \xrightarrow{?} ([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

1. Is any q compatible with such a reduction ?
2. Is this typable ?

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

1. Is any q compatible with such a reduction ?

- If q eventually gives a value V:

$$([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda a.[\![p]\!])[\![V]\!])[\![e]\!] \to [\![p]\!][[\![V]\!]/a][\![e]\!] = [\![p[V/a]]\!][\![e]\!] \quad ✓$$

- If $[\![q]\!] \to \lambda\_.t$ and drops its continuation (meaning $t : \bot$):

$$([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda\_.t)\lambda a.[\![p]\!])[\![e]\!] \to t[\![e]\!] \quad ✗$$

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!]$$

Questions:

1. Is any q compatible with such a reduction ? $\rightsquigarrow q \in \mathrm{NEF}$

- If q eventually gives a value V:

$$([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda a.[\![p]\!])[\![V]\!])[\![e]\!] \to [\![p]\!][[\![V]\!]/a][\![e]\!] = [\![p[V/a]]\!][\![e]\!] \quad \checkmark$$

- If $[\![q]\!] \to \lambda\_.t$ and drops its continuation (meaning $t : \bot$):

$$([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!] \to ((\lambda\_.t)\lambda a.[\![p]\!])[\![e]\!] \to t[\![e]\!] \quad ✗$$

Negative-elimination free (Herbelin'12)

Values + one continuation variable + no application

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\,(\lambda a.[\![p]\!]))\,[\![e]\!]$$

Questions:

1. Is any q compatible with such a reduction ? $\rightsquigarrow q \in \mathrm{NEF}$
2. Is this typable ?

Naive attempt:

$$\underbrace{(\underbrace{[\![q]\!]}_{(A\to\bot)\to\bot}\;(\underbrace{\lambda a.[\![p]\!]}_{\Pi_{(a:A)}\neg\neg B(a)}))}_{✗}\;\underbrace{[\![e]\!]}_{\neg B(q)}$$

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\ (\lambda a.[\![p]\!]))\,[\![e]\!]$$

Questions:

1. Is any q compatible with such a reduction ? $\rightsquigarrow q \in \mathrm{NEF}$
2. Is this typable ?

Friedman's trick:

$$\underbrace{(\underbrace{[\![q]\!]}_{\forall R.(A\to R?)\to R?}\ (\underbrace{\lambda a.[\![p]\!]}_{\Pi_{(a:A)}\neg\neg B(a)}))}_{\neg\neg B}\ \underbrace{[\![e]\!]}_{\neg B(q)}$$

CPS | Call-by-value | Π-type (2/2)

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \xrightarrow{?} ([\![q]\!]\,(\lambda a.[\![p]\!]))\,[\![e]\!]$$

Questions:

1. Is any q compatible with such a reduction ? $\rightsquigarrow q \in \text{NEF}$
2. Is this typable ? $\rightsquigarrow$ *parametric return-type*

Better:

$$\underbrace{(\underbrace{[\![q]\!]}_{\forall R.(\Pi_{(a:A)}R(a))\to R(q)}\;(\underbrace{\lambda a.[\![p]\!]}_{\Pi_{(a:A)}\neg\neg B(a)}))}_{\neg\neg B(q)}\;\underbrace{[\![e]\!]}_{\neg B(q)}$$

(*Remark: not possible without* $q \in$ NEF)

Delimited continuations

$$[\![\langle \lambda a.p \parallel q \cdot e\rangle]\!] \quad\longrightarrow\quad ([\![q]\!]\,(\lambda a.[\![p]\!]))[\![e]\!]$$

So, we're looking for:

$$\langle \lambda a.p \parallel q \cdot e\rangle \quad\xrightarrow{q\in\mathsf{NEF}}\quad \Big\langle \mu\,?\,.\langle q \parallel \tilde\mu a.\langle p \parallel \,?\,\rangle\rangle \,\Big\|\, e\Big\rangle$$

such that we first reduce $\langle q \parallel \tilde\mu a.\langle p \parallel \,?\,\rangle\rangle$.

Delimited continuations:

$$\begin{array}{rcll} \langle \mu\hat{\mathrm{tp}}.c \parallel e\rangle & \longrightarrow & \langle \mu\hat{\mathrm{tp}}.c' \parallel e\rangle & (\text{if } c \to c') \\ \langle \mu\hat{\mathrm{tp}}.\langle p \parallel \hat{\mathrm{tp}}\rangle \parallel e\rangle & \longrightarrow & \langle p \parallel e\rangle & \end{array}$$

In other words:

$$q \cdot e \triangleq \tilde\mu b.\Big\langle \mu\hat{\mathrm{tp}}.\langle q \parallel \tilde\mu v.\langle p \parallel v \cdot \hat{\mathrm{tp}}\rangle\rangle \,\Big\|\, e\Big\rangle \qquad (q \in \mathsf{NEF})$$

Delimited continuations

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \quad \longrightarrow \quad ([\![q]\!]\,(\lambda a.[\![p]\!]))[\![e]\!]$$

So, we're looking for:

$$\langle \lambda a.p \,\|\, q \cdot e\rangle \quad \xrightarrow{q \in \mathrm{NEF}} \quad \Big\langle \mu\, ?\, .\langle q \,\|\, \tilde{\mu} a.\langle p \,\|\, ?\, \rangle\rangle \,\Big\|\, e \Big\rangle$$

such that we first reduce $\langle q \,\|\, \tilde{\mu} a.\langle p \,\|\, ?\, \rangle\rangle$.

Delimited continuations:

$$\begin{array}{rcll} \langle \mu \hat{\mathrm{tp}}.c \,\|\, e\rangle & \longrightarrow & \langle \mu \hat{\mathrm{tp}}.c' \,\|\, e\rangle & \quad (\text{if } c \to c') \\ \langle \mu \hat{\mathrm{tp}}.\langle p \,\|\, \hat{\mathrm{tp}}\rangle \,\|\, e\rangle & \longrightarrow & \langle p \,\|\, e\rangle & \end{array}$$

In other words:

$$q \cdot e \triangleq \tilde{\mu} b.\Big\langle \mu \hat{\mathrm{tp}}.\langle q \,\|\, \tilde{\mu} v.\langle p \,\|\, v \cdot \hat{\mathrm{tp}}\rangle\rangle \,\Big\|\, e \Big\rangle \qquad (q \in \mathrm{NEF})$$

Delimited continuations

$$[\![\langle \lambda a.p \,\|\, q \cdot e\rangle]\!] \quad \longrightarrow \quad ([\![q]\!]\ (\lambda a.[\![p]\!]))[\![e]\!]$$

So, we're looking for:

$$\langle \lambda a.p \,\|\, q \cdot e\rangle \quad \xrightarrow{q\in\mathrm{NEF}} \quad \Big\langle \mu\hat{\mathrm{tp}}.\langle q \,\|\, \tilde{\mu}a.\langle p \,\|\, \hat{\mathrm{tp}}\rangle\rangle \,\Big\|\, e\Big\rangle$$

such that we first reduce $\langle q \,\|\, \tilde{\mu}a.\langle p \,\|\, \hat{\mathrm{tp}}\rangle\rangle$.

Delimited continuations:

$$\begin{array}{rcll} \langle \mu\hat{\mathrm{tp}}.c \,\|\, e\rangle & \longrightarrow & \langle \mu\hat{\mathrm{tp}}.c' \,\|\, e\rangle & \qquad (\text{if } c \to c') \\ \langle \mu\hat{\mathrm{tp}}.\langle p \,\|\, \hat{\mathrm{tp}}\rangle \,\|\, e\rangle & \longrightarrow & \langle p \,\|\, e\rangle & \end{array}$$

In other words:

$$q \cdot e \triangleq \tilde{\mu}b.\Big\langle \mu\hat{\mathrm{tp}}.\langle q \,\|\, \tilde{\mu}v.\langle p \,\|\, v \cdot \hat{\mathrm{tp}}\rangle\rangle \,\Big\|\, e\Big\rangle \qquad (q \in \mathrm{NEF})$$

Call-by-value | Σ-type

Exact same story:

$$\dfrac{\dfrac{\dfrac{x:T\vdash x:T \qquad a:\cancel{A[t]}\vdash a:\cancel{A[x]}}{x:T,a:A[t]\vdash (x,a):\Sigma(x:T).A}\,\textit{Mism.}}{\vdash p:A[t] \qquad \vdots}}{}$$

$$\dfrac{\dfrac{\vdash t:T \quad \vdash p:A[t]}{\vdash (t,p):\Sigma(x:T).A} \quad \dots}{\langle (t,p)\,\|\, e\rangle} \xrightarrow{?} \dfrac{\vdots}{\Big\langle t \,\Big\|\, \tilde\mu x.\langle p \,\|\, \tilde\mu a.\langle (x,a)\,\|\, e\rangle\rangle \Big\rangle}$$

CPS:

$$[\![(t,p)]\!]_p\, k \triangleq [\![t]\!]_t\, (\lambda x.\, \underbrace{[\![p]\!]_p}_{\neg\neg A[t]}\, \underbrace{(\lambda a.k\,(x,a))}_{\neg A[x]})$$

$[\![p]\!]$ shouldn't be applied to its continuation before $[\![t]\!]$ has reduced

Call-by-value | Σ-type

$[\![p]\!]$ shouldn't be applied to its continuation before $[\![t]\!]$ has reduced

CPS:

$$[\![(t,p)]\!]_p\, k \triangleq \overbrace{[\![p]\!]_p}^{\neg\neg A[t]}\ \overbrace{\Big(\underbrace{[\![t]\!]_t}_{\forall R.\Pi(x:T).R[x]\to R[t]}\ \underbrace{(\lambda xa.k\,(x,a))}_{\Pi(x:T).\neg A[x]}\Big)}^{\neg A[t]}$$

Co-delimited continuations:

$$\langle (t,p) \,\|\, e\rangle \longrightarrow \Big\langle p \,\Big\|\, \tilde\mu \check{\mathrm{tp}}.\langle t \,\|\, \tilde\mu x.\langle \check{\mathrm{tp}} \,\|\, \tilde\mu a.\langle (x,a) \,\|\, e\rangle\rangle\rangle \Big\rangle$$

Conclusion

Here comes duality again!

$$\begin{tikzcd}
\text{MLTT CbV} \arrow[r, "\text{embed.}"] & d\lambda\mu\tilde{\mu}\text{-calculus CbV} \arrow[r, "\text{CPS}"] & \text{MLTT} \\
\text{MLTT CbN} \arrow[r, "\text{embed.}"] & d\lambda\mu\tilde{\mu}\text{-calculus CbN} \arrow[r, "\text{CPS}"] & \text{MLTT}
\end{tikzcd}$$

Call-by-name:

- $(q \cdot e) : \Pi(a : A).B$ is not problematic (q is to be substituted directly)
- $(t,p) : \Sigma(a : A).B$ poses the exact same problem

↬ same ideas allow to soundly define reduction & CPS

In each case:

- problem of *synchronizing the evaluation* of a *pair* (t,p) / $(q \cdot e)$
- (co-)*delimited continuations* solve the problem
- *dependent and parametric* return-type in the CPS
- requires a restriction to "*nice*" terms

Here comes duality again!

$$\begin{tikzcd} \text{MLTT CbV} \arrow[r, "\text{embed.}"] & d\lambda\mu\tilde{\mu}\text{-calculus CbV} \arrow[r, "\text{CPS}"] & \text{MLTT} \\ \text{MLTT CbN} \arrow[r, "\text{embed.}"] & d\lambda\mu\tilde{\mu}\text{-calculus CbN} \arrow[r, "\text{CPS}"] & \text{MLTT} \end{tikzcd}$$

Call-by-name:

- $(q \cdot e) : \Pi(a : A).B$ is not problematic (q is to be substituted directly)
- $(t,p) : \Sigma(a : A).B$ poses the exact same problem

↪ same ideas allow to soundly define reduction & CPS

In each case:

- problem of *synchronizing the evaluation* of a *pair* (t,p) / $(q \cdot e)$
- (co-)*delimited continuations* solve the problem
- *dependent and parametric* return-type in the CPS
- requires a restriction to "*nice*" terms

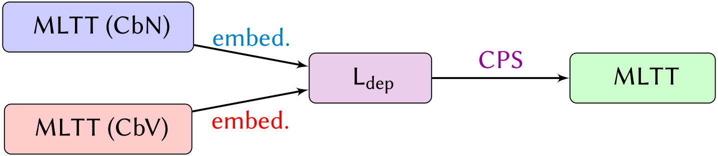
Occam's razor

Observation:

> the solutions are unrelated to the evaluation strategy

Couldn't we:

1. consider a core calculus with pairs/delimited continuations?
2. use polarities to define evaluation order?



Conjecture: ($\mathsf{L}_{\mathrm{dep}}$)

$$\begin{array}{lcl} V, V' & ::= & x \mid (V, V') \mid \mu(\kappa_1 \cdot \kappa_2).c \mid \mu x^{-}.c \\ t, u & ::= & V \mid \mu x^{+}.c \mid \hat{\mu}.c \mid \hat{} \end{array} \qquad \begin{array}{lcl} c & ::= & \langle t \,\|\, t' \rangle^{-} \\ \varepsilon & ::= & - \mid + \end{array}$$

Future work

(*starring Guillaume Munch-Maccagnoni*)

Study $\mathbf{L_{dep}}$, sequent calculus presentation for a dependent CBPV

- "*nice*" $\stackrel{?}{=}$ thunkable
- compatible with different kinds of effect?

Connections with Vákár's dCBPV:

- thunkable terms as the category of values?
- $V ::= \ldots \mid t^\bullet$ with t thunkable?

Connections with Pédrot-Tabareau's Baclofen TT:

- translations does not account for classical logic
- what about a sequent calculus view of the effects they handled?

Thank you for you attention.