Two ways to the focused sequent calculi

Matthias Puech

Parsifal working group,
LIX, Ecole Polytechnique

November 10, 2015

In this talk

My two encounters with focusing while studying the Curry-Howard isomorphism

In this talk

My two encounters with focusing while studying the Curry-Howard isomorphism

1. From Natural deduction to LJT
2. From the CPS transformation to LJQ

In this talk

My two encounters with focusing while studying the Curry-Howard isomorphism

1. From Natural deduction to LJT
2. From the CPS transformation to LJQ

Goal

Trigger discussions on term assignments for LKF

1. From Natural Deduction to LJT

Proofs, upside down [Puech, 2013]

2. From the CPS transformation to LJQ

Typeful CPS transformations [Danvy & Puech, 201?]

Warning: raw material

Continuation-passing styles

A CPS transformation is

- a programming technique
- an intermediate language in compilers
  (complex language $\rightarrow$ simpler language)
- a semantic artifact
  ($\simeq$ operational/denotational/process/...semantics)
- a proof transformation
  (classical $\rightarrow$ intuitionistic)
- ...

Continuation-passing styles

A CPS transformation is

- a programming technique
- an intermediate language in compilers
  (complex language $\to$ simpler language)
- a semantic artifact
  ($\simeq$ operational/denotational/process/…semantics)
- a proof transformation
  (classical $\to$ intuitionistic)
- …

Many variants, long, long history

My interrogations

- How can it be so many things at the same time?
- What does it correspond to as a proof system?
- What is this thing anyway?

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\, (\lambda x.\, [\![M]\!]) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, k)) \\
[\![\mathbf{let}\, x = M\, \mathbf{in}\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, k)
\end{aligned}
$$

From the CPS transformation to LJQ

The motivation

Understand CPS through syntax and typing

From the CPS transformation to LJQ

The motivation

Understand CPS through syntax and typing

The tools

We are going to engineer a *one-pass*, *β-normal* CPS thanks to:

- gradual analysis and optimization
- tight typed syntax (OCaml's GADTs)

From the CPS transformation to LJQ

The motivation

Understand CPS through syntax and typing

The tools

We are going to engineer a *one-pass*, *β-normal* CPS thanks to:

- gradual analysis and optimization
- tight typed syntax (OCaml's GADTs)

The result (SPOILER)

A type system for the β-normal forms of CPS terms

From the CPS transformation to LJQ

The motivation

Understand CPS through syntax and typing

The tools

We are going to engineer a *one-pass*, *β-normal* CPS thanks to:

- gradual analysis and optimization
- tight typed syntax (OCaml's GADTs)

The result (SPOILER)

A type system for the β-normal forms of CPS terms

$= \text{ANF!}$

From the CPS transformation to LJQ

The motivation

Understand CPS through syntax and typing

The tools

We are going to engineer a *one-pass*, *β-normal* CPS thanks to:

- gradual analysis and optimization
- tight typed syntax (OCaml's GADTs)

The result (SPOILER)

A type system for the β-normal forms of CPS terms

= ANF!

= LJQ!

From the CPS transformation to LJQ

The motivation

Understand CPS through syntax and typing

The tools

We are going to engineer a *one-pass*, *β-normal* CPS thanks to:

- gradual analysis and optimization
- tight typed syntax (OCaml's GADTs)

The result (SPOILER)

A type system for the β-normal forms of CPS terms

= ANF!

= LJQ!

here: *call-by-value* (exercise: *call-by-name*)

Outline

1. Fischer & Plotkin's original CPS transformation
2. *One-pass CPS* (through Control-Flow Analysis)
3. The syntax of CPS terms (through syntax aggregation)
4. Proper transformation of β*-redexes*

Fischer & Plotkin's original transformation

$$M ::= \lambda x.\, M \mid M\ M \mid x \mid \textbf{let}\ x = M\ \textbf{in}\ M \qquad \in \mathit{Exp}$$

Fischer & Plotkin's original transformation

$$M ::= \lambda x.\, M \mid M\; M \mid x \mid \mathbf{let}\; x = M \;\mathbf{in}\; M \qquad \in \mathit{Exp}$$

$$[\![\cdot]\!] \;:\; M \to M$$

$$\begin{aligned}
[\![x]\!] &= \lambda k.\, k\; x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\; (\lambda x.\, [\![M]\!]) \\
[\![M\; N]\!] &= \lambda k.\, [\![M]\!]\; (\lambda m.\, [\![N]\!]\; (\lambda n.\, m\; n\; k)) \\
[\![\mathbf{let}\; x = M \;\mathbf{in}\; N]\!] &= \lambda k.\, [\![M]\!]\; (\lambda x.\, [\![N]\!]\; k)
\end{aligned}$$

Fischer & Plotkin's original transformation

$$M ::= \lambda x.\, M \mid M\ M \mid x \mid \mathbf{let}\ x = M\ \mathbf{in}\ M \qquad \in Exp$$

$$[\![\cdot]\!] \;:\; M \to M$$

$$\begin{aligned}
[\![x]\!] &= \lambda k.\, k\ x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\ (\lambda x.\, [\![M]\!]) \\
[\![M\ N]\!] &= \lambda k.\, [\![M]\!]\ (\lambda m.\, [\![N]\!]\ (\lambda n.\, m\ n\ k)) \\
[\![\mathbf{let}\ x = M\ \mathbf{in}\ N]\!] &= \lambda k.\, [\![M]\!]\ (\lambda x.\, [\![N]\!]\ k)
\end{aligned}$$

Properties

Simulation $[\![eval_v(M)]\!] \simeq eval_v([\![M]\!]\ (\lambda x.\, x))$

Fischer & Plotkin's original transformation

$$M ::= \lambda x.\, M \mid M\; M \mid x \mid \mathbf{let}\; x = M \;\mathbf{in}\; M \qquad \in Exp$$

$$[\![\cdot]\!] \;:\; M \to M$$

$$\begin{aligned}
[\![x]\!] &= \lambda k.\, k\; x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\; (\lambda x.\, [\![M]\!]) \\
[\![M\; N]\!] &= \lambda k.\, [\![M]\!]\; (\lambda m.\, [\![N]\!]\; (\lambda n.\, m\; n\; k)) \\
[\![\mathbf{let}\; x = M \;\mathbf{in}\; N]\!] &= \lambda k.\, [\![M]\!]\; (\lambda x.\, [\![N]\!]\; k)
\end{aligned}$$

Properties

Simulation $[\![eval_v(M)]\!] \simeq eval_v([\![M]\!]\; (\lambda x.x))$

Indifference $eval_v([\![M]\!](\lambda x.x)) \simeq eval_n([\![M]\!]\; (\lambda x.x))$

Fischer & Plotkin's original transformation

$$M ::= \lambda x.M \mid M\ M \mid x \mid \textbf{let}\ x = M\ \textbf{in}\ M \qquad \in Exp$$

$$[\![\cdot]\!] \ : \ M \to M$$

$$\begin{aligned}
[\![x]\!] &= \lambda k.\, k\ x \\
[\![\lambda x.M]\!] &= \lambda k.\, k\ (\lambda x.\, [\![M]\!]) \\
[\![M\ N]\!] &= \lambda k.\, [\![M]\!]\ (\lambda m.\, [\![N]\!]\ (\lambda n.\, m\ n\ k)) \\
[\![\textbf{let}\ x = M\ \textbf{in}\ N]\!] &= \lambda k.\, [\![M]\!]\ (\lambda x.\, [\![N]\!]\ k)
\end{aligned}$$

$$\begin{aligned}
[\![A]\!] &= (\lfloor\!\lfloor A \rfloor\!\rfloor \to o) \to o \\
\lfloor\!\lfloor A \to B \rfloor\!\rfloor &= \lfloor\!\lfloor A \rfloor\!\rfloor \to [\![B]\!]
\end{aligned}$$

Properties

Simulation $[\![eval_v(M)]\!] \simeq eval_v([\![M]\!]\ (\lambda x.x))$

Indifference $eval_v([\![M]\!](\lambda x.x)) \simeq eval_n([\![M]\!]\ (\lambda x.x))$

Preservation of typing If $\Gamma \vdash M : A$ then $\Gamma \vdash [\![M]\!] : [\![A]\!]$

Problem *"administrative redexes"*

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\, (\lambda x.\, [\![M]\!]) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, k)) \\
[\![\mathbf{let}\ x = M\ \mathbf{in}\ N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, k)
\end{aligned}
$$

Examples

- $[\![\lambda x.\, x]\!] = \lambda k.\, k\, (\lambda xk.\, k\, x)$

Problem *"administrative redexes"*

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\, (\lambda x.\, [\![M]\!]) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, k)) \\
[\![\mathbf{let}\ x = M\ \mathbf{in}\ N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, k)
\end{aligned}
$$

Examples

- $[\![\lambda x.x]\!] = \lambda k.\, k\, (\lambda xk.\, k\, x)$
- $[\![\lambda x.x\, x]\!] = \lambda k.\, k\, (\lambda xk.\, (\lambda k.\, k\, x)\, (\lambda m.\, (\lambda k.\, k\, x)\, (\lambda n.\, m\, n\, k)))$

Problem *“administrative redexes”*

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x. M]\!] &= \lambda k.\, k\, (\lambda x. [\![M]\!]) \\
[\![M\, N]\!] &= \lambda k. [\![M]\!]\, (\lambda m. [\![N]\!]\, (\lambda n. m\, n\, k)) \\
[\![\mathbf{let}\, x = M\, \mathbf{in}\, N]\!] &= \lambda k. [\![M]\!]\, (\lambda x. [\![N]\!]\, k)
\end{aligned}
$$

Examples

- $[\![\lambda x. x]\!] = \lambda k.\, k\, (\lambda xk.\, k x)$
- $[\![\lambda x. x\, x]\!] = \lambda k.\, k\, (\lambda xk. (\lambda k.\, k\, x)\, (\lambda m. (\lambda k.\, k\, x)\, (\lambda n. m\, n\, k)))$

Proposition

Translate, then reduce administrative redexes (two passes).

Problem *“administrative redexes”*

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\, (\lambda x.\, [\![M]\!]) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, k)) \\
[\![\mathbf{let}\, x = M\, \mathbf{in}\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, k)
\end{aligned}
$$

Examples

- $[\![\lambda x.\, x]\!] = \lambda k.\, k\, (\lambda x k.\, k x)$
- $[\![\lambda x.\, x\, x]\!] = \lambda k.\, k\, (\lambda x k.\, (\lambda k.\, k\, x)\, (\lambda m.\, (\lambda k.\, k\, x)\, (\lambda n.\, m\, n\, k)))$

Proposition

Translate, then reduce administrative redexes (two passes).
But how to distinguish administrative/source redexes?

Analysis Control flow in the CPS

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\, (\lambda x.\, [\![M]\!]) \\
[\![M\ N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\ n\ k)) \\
[\![\mathbf{let}\ x = M\ \mathbf{in}\ N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\ k)
\end{aligned}
$$

Analysis Control flow in the CPS

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\; x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\; (\lambda x.\, [\![M]\!]) \\
[\![M\; N]\!] &= \lambda k.\, [\![M]\!]\; (\lambda m.\, [\![N]\!]\; (\lambda n.\, m\; n\; k)) \\
[\![\mathbf{let}\; x = M\; \mathbf{in}\; N]\!] &= \lambda k.\, [\![M]\!]\; (\lambda x.\, [\![N]\!]\; k)
\end{aligned}
$$

1. where can the $\lambda k.$ occur in the residual term?

Analysis Control flow in the CPS

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\, (\lambda x.\, [\![M]\!]) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, k)) \\
[\![\mathbf{let}\; x = M\; \mathbf{in}\; N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, k)
\end{aligned}
$$

1. where can the $\lambda k.$ occur in the residual term?

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x. M]\!] &= \lambda k.\, k\, (\lambda x k.\, [\![M]\!]\, k) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, k)) \\
[\![\textbf{let}\, x = M\ \textbf{in}\ N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, k)
\end{aligned}
$$

1. where can the $\lambda k.$ occur in the residual term?

Analysis Control flow in the CPS

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.M]\!] &= \lambda k.\, k\, (\lambda x k.\, [\![M]\!]\, k) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, k)) \\
[\![\textbf{let}\ x = M\ \textbf{in}\ N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, k)
\end{aligned}
$$

1. where can the $\lambda k.$ occur in the residual term?
2. which terms can be denoted by the k?

Analysis Control flow in the CPS

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.M]\!] &= \lambda k.\, k\, (\lambda x k.\, [\![M]\!]\, (\lambda m.\, k\, m)) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, k)) \\
[\![\mathbf{let}\; x = M\; \mathbf{in}\; N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, (\lambda n.\, k\, n))
\end{aligned}
$$

1. where can the $\lambda k.$ occur in the residual term?
2. which terms can be denoted by the k?

Analysis Control flow in the CPS

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.\, M]\!] &= \lambda k.\, k\, (\lambda x k.\, [\![M]\!]\, (\lambda m.\, k\, m)) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, k)) \\
[\![\textbf{let}\; x = M\; \textbf{in}\; N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, (\lambda n.\, k\, n))
\end{aligned}
$$

1. where can the $\lambda k.$ occur in the residual term?
2. which terms can be denoted by the k?
3. where do these k occur?

Analysis Control flow in the CPS

$$
\begin{aligned}
[\![x]\!] &= \lambda k.\, k\, x \\
[\![\lambda x.M]\!] &= \lambda k.\, k\, (\lambda x k.\, [\![M]\!]\, (\lambda m.\, k\, m)) \\
[\![M\, N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda m.\, [\![N]\!]\, (\lambda n.\, m\, n\, (\lambda v.\, k\, v))) \\
[\![\mathbf{let}\; x = M\; \mathbf{in}\; N]\!] &= \lambda k.\, [\![M]\!]\, (\lambda x.\, [\![N]\!]\, (\lambda n.\, k\, n))
\end{aligned}
$$

1. where can the $\lambda k.$ occur in the residual term?
2. which terms can be denoted by the k?
3. where do these k occur?

Analysis Control flow in the CPS

$$
\begin{aligned}
[\![x]\!] &= \lambda K.\, K[x] \\
[\![\lambda x.\, M]\!] &= \lambda K.\, K[\lambda x k.\, [\![M]\!][\lambda M.\, k\; M]] \\
[\![M\; N]\!] &= \lambda K.\, [\![M]\!][\lambda M.\, [\![N]\!][\lambda N.\, M\; N\; (\lambda v.\, K[v])]] \\
[\![\mathbf{let}\; x = M\; \mathbf{in}\; N]\!] &= \lambda K.\, [\![M]\!][\lambda X.\, [\![N]\!]\; [\lambda N.\, K[N]]]
\end{aligned}
$$

1. where can the $\lambda k.$ occur in the residual term?
2. which terms can be denoted by the k?
3. where do these k occur?
4. what are the static abs. $\lambda X.\, T$ and app. $T[U]$?

Analysis Control flow in the CPS

$$
\begin{aligned}
[\![x]\!] &= \lambda K.\, K[x] \\
[\![\lambda x.\, M]\!] &= \lambda K.\, K[\lambda x k.\, [\![M]\!][\lambda M.\, k\, M]] \\
[\![M\ N]\!] &= \lambda K.\, [\![M]\!][\lambda M.\, [\![N]\!][\lambda N.\, M\ N\ (\lambda v.\, K[v])]] \\
[\![\mathbf{let}\, x = M\ \mathbf{in}\ N]\!] &= \lambda K.\, [\![M]\!][\lambda M.\, \mathbf{let}\, x = M\ \mathbf{in}\ [\![N]\!]\ [\lambda N.\, K[N]]]
\end{aligned}
$$

1. where can the $\lambda k.$ occur in the residual term?
2. which terms can be denoted by the k?
3. where do these k occur?
4. what are the static abs. $\lambda X.\, T$ and app. $T[U]$?
5. are there variable mismatches?

Result The *one-pass* CPS transform

(Danvy & Filinski, *Representing Control*, 1991)

$$
\begin{aligned}
[\![x]\!][K] &= K[x] \\
[\![\lambda x.\,M]\!][K] &= K[\lambda xk.\,[\![M]\!][\lambda M.\,k\;M]] \\
[\![M\;N]\!][K] &= [\![M]\!]\;[\lambda M.\,[\![N]\!]\;(\lambda N.\,M\;N\;(\lambda v.\,K[v]))] \\
[\![\mathbf{let}\;x = M\;\mathbf{in}\;N]\!][K] &= [\![M]\!]\;[\lambda M.\,\mathbf{let}\;x = M\;\mathbf{in}\;[\![N]\!][\lambda N.\,K[N]]]
\end{aligned}
$$

Result The *one-pass* CPS transform

(Danvy & Filinski, *Representing Control*, 1991)

$$[\![\cdot]\!][\cdot] \;:\; M \to (M \to M) \to M$$

$$\begin{aligned}
[\![x]\!][K] &= K[x] \\
[\![\lambda x.\, M]\!][K] &= K[\lambda xk.\, [\![M]\!][\lambda M.\, k\ M]] \\
[\![M\ N]\!][K] &= [\![M]\!]\ [\lambda M.\, [\![N]\!]\ (\lambda N.\, M\ N\ (\lambda v.\, K[v]))] \\
[\![\mathbf{let}\ x = M\ \mathbf{in}\ N]\!][K] &= [\![M]\!]\ [\lambda M.\, \mathbf{let}\ x = M\ \mathbf{in}\ [\![N]\!][\lambda N.\, K[N]]]
\end{aligned}$$

Result The *one-pass* CPS transform

(Danvy & Filinski, *Representing Control*, 1991)

$$[\![\cdot]\!][\cdot] \;:\; M \to (M \to M) \to M$$

$$\begin{aligned}
[\![x]\!][K] &= K[x] \\
[\![\lambda x.\, M]\!][K] &= K[\lambda x k.\, [\![M]\!][\lambda M.\, k\; M]] \\
[\![M\; N]\!][K] &= [\![M]\!]\; [\lambda M.\, [\![N]\!]\; (\lambda N.\, M\; N\; (\lambda v.\, K[v]))] \\
[\![\mathbf{let}\; x = M \;\mathbf{in}\; N]\!][K] &= [\![M]\!]\; [\lambda M.\, \mathbf{let}\; x = M \;\mathbf{in}\; [\![N]\!][\lambda N.\, K[N]]]
\end{aligned}$$

$$[\![\cdot]\!] \;:\; M \to M$$

$$[\![M]\!] = [\![M]\!][?]$$

Result The *one-pass* CPS transform

(Danvy & Filinski, *Representing Control*, 1991)

$$[\![\cdot]\!][\cdot] \ : \ M \to (M \to M) \to M$$

$$\begin{aligned}
[\![x]\!][K] &= K[x] \\
[\![\lambda x.\, M]\!][K] &= K[\lambda x k.\, [\![M]\!][\lambda M.\, k\, M]] \\
[\![M\ N]\!][K] &= [\![M]\!]\ [\lambda M.\, [\![N]\!]\ (\lambda N.\, M\ N\ (\lambda v.\, K[v]))] \\
[\![\mathbf{let}\ x = M\ \mathbf{in}\ N]\!][K] &= [\![M]\!]\ [\lambda M.\, \mathbf{let}\ x = M\ \mathbf{in}\ [\![N]\!][\lambda N.\, K[N]]]
\end{aligned}$$

$$[\![\cdot]\!] \ : \ M \to M$$

$$[\![M]\!] = \lambda k.\, [\![M]\!][k]$$

Result The *one-pass* CPS transform

(Danvy & Filinski, *Representing Control*, 1991)

$$[\![\cdot]\!][\cdot] \;:\; M \to (M \to M) \to M$$

$$\begin{aligned}
[\![x]\!][K] &= K[x] \\
[\![\lambda x.\, M]\!][K] &= K[\lambda x k.\, [\![M]\!][\lambda M.\, k\; M]] \\
[\![M\; N]\!][K] &= [\![M]\!]\; [\lambda M.\, [\![N]\!]\; (\lambda N.\, M\; N\; (\lambda v.\, K[v]))] \\
[\![\mathbf{let}\; x = M\; \mathbf{in}\; N]\!][K] &= [\![M]\!]\; [\lambda M.\, \mathbf{let}\; x = M\; \mathbf{in}\; [\![N]\!][\lambda N.\, K[N]]]
\end{aligned}$$

$$[\![\cdot]\!] \;:\; M \to M$$

$$[\![M]\!] = \lambda k.\, [\![M]\!][\lambda M.\, k\; M]$$

Result The *one-pass* CPS transform

(Danvy & Filinski, *Representing Control*, 1991)

$$[\![\cdot]\!][\cdot] \; : \; M \to (M \to M) \to M$$

$$\begin{aligned}
[\![x]\!][K] &= K[x] \\
[\![\lambda x.\, M]\!][K] &= K[\lambda xk.\, [\![M]\!][\lambda M.\, k\; M]] \\
[\![M\; N]\!][K] &= [\![M]\!]\; [\lambda M.\, [\![N]\!]\; (\lambda N.\, M\; N\; (\lambda v.\, K[v]))] \\
[\![\textbf{let}\; x = M\; \textbf{in}\; N]\!][K] &= [\![M]\!]\; [\lambda M.\, \textbf{let}\; x = M\; \textbf{in}\; [\![N]\!][\lambda N.\, K[N]]]
\end{aligned}$$

$$[\![\cdot]\!] \; : \; M \to M$$

$$[\![M]\!] = \lambda k.\, [\![M]\!][\lambda M.\, k\; M]$$

Examples

- $[\![\lambda x.\, x]\!] = \lambda k.\, k\; (\lambda xk.\, k\; x)$

Result The *one-pass* CPS transform

(Danvy & Filinski, *Representing Control*, 1991)

$$[\![\cdot]\!][\cdot] \;:\; M \to (M \to M) \to M$$

$$\begin{aligned}
[\![x]\!][K] &= K[x] \\
[\![\lambda x.\,M]\!][K] &= K[\lambda xk.\,[\![M]\!][\lambda M.\,k\,M]] \\
[\![M\;N]\!][K] &= [\![M]\!]\;[\lambda M.\,[\![N]\!]\;(\lambda N.\,M\;N\;(\lambda v.\,K[v]))] \\
[\![\mathbf{let}\;x = M\;\mathbf{in}\;N]\!][K] &= [\![M]\!]\;[\lambda M.\,\mathbf{let}\;x = M\;\mathbf{in}\;[\![N]\!][\lambda N.\,K[N]]]
\end{aligned}$$

$$[\![\cdot]\!] \;:\; M \to M$$

$$[\![M]\!] = \lambda k.\,[\![M]\!][\lambda M.\,k\,M]$$

Examples

- $[\![\lambda x.\,x]\!] = \lambda k.\,k\;(\lambda xk.\,k\;x)$
- $[\![\lambda x.\,x\;x]\!] = \lambda k.\,k\;(\lambda xk.\,x\;x\;(\lambda v.\,k\;v))$

Result The *one-pass* CPS transform

(Danvy & Filinski, *Representing Control*, 1991)

$$[\![\cdot]\!][\cdot] \;:\; M^{A} \to (M^{A} \to M^{[\![A]\!]}) \to M^{[\![A]\!]}$$

$$\begin{aligned}
[\![x]\!][K] &= K[x] \\
[\![\lambda x.\, M]\!][K] &= K[\lambda xk.\, [\![M]\!][\lambda M.\, k\, M]] \\
[\![M\; N]\!][K] &= [\![M]\!]\; [\lambda M.\, [\![N]\!]\; (\lambda N.\, M\; N\; (\lambda v.\, K[v]))] \\
[\![\mathbf{let}\; x = M \;\mathbf{in}\; N]\!][K] &= [\![M]\!]\; [\lambda M.\, \mathbf{let}\; x = M \;\mathbf{in}\; [\![N]\!][\lambda N.\, K[N]]]
\end{aligned}$$

$$[\![\cdot]\!] \;:\; M^{A} \to M^{[\![A]\!]}$$

$$[\![M]\!] = \lambda k.\, [\![M]\!][\lambda M.\, k\, M]$$

Examples

- $[\![\lambda x.\, x]\!] = \lambda k.\, k\; (\lambda xk.\, k\; x)$
- $[\![\lambda x.\, x\; x]\!] = \lambda k.\, k\; (\lambda xk.\, x\; x\; (\lambda v.\, k\; v))$

Problem What is the structure of CPS terms?

$$\llbracket x \rrbracket\, K = K[x]$$
$$\llbracket \lambda x. M \rrbracket\, K = K[\lambda xk. \llbracket M \rrbracket [\lambda M. k\, M]]$$
$$\llbracket M\, N \rrbracket\, K = \llbracket M \rrbracket\, [\lambda M. \llbracket N \rrbracket\, (\lambda N. M\, N\, (\lambda v. K[v]))]$$
$$\llbracket \textbf{let}\ x = M\ \textbf{in}\ N \rrbracket\, K = \llbracket M \rrbracket\, [\lambda M. \textbf{let}\ x = M\ \textbf{in}\ \llbracket N \rrbracket [\lambda N. K[N]]]$$

$$\llbracket M \rrbracket = \lambda k. \llbracket M \rrbracket\, [\lambda M. k\, M]$$

Quiz

Is there M s.t. $\llbracket M \rrbracket = \lambda k. k\, (\lambda xk. x)$?

Problem What is the structure of CPS terms?

$$\llbracket x \rrbracket\, K = K[x]$$
$$\llbracket \lambda x.M \rrbracket\, K = K[\lambda xk.\llbracket M \rrbracket[\lambda M.k\, M]]$$
$$\llbracket M\, N \rrbracket\, K = \llbracket M \rrbracket\, [\lambda M.\llbracket N \rrbracket\, (\lambda N.M\, N\, (\lambda v.K[v]))]$$
$$\llbracket \mathbf{let}\ x = M\ \mathbf{in}\ N \rrbracket\, K = \llbracket M \rrbracket\, [\lambda M.\mathbf{let}\ x = M\ \mathbf{in}\ \llbracket N \rrbracket[\lambda N.K[N]]]$$

$$\llbracket M \rrbracket = \lambda k.\llbracket M \rrbracket\, [\lambda M.k\, M]$$

Quiz

Is there M s.t. $\llbracket M \rrbracket = \lambda k.k\, (\lambda xk.x)$?
What is the image of the one-pass CPS transform?

Problem What is the structure of CPS terms?

$$
\begin{aligned}
[\![x]\!]\, K &= K[x] \\
[\![\lambda x. M]\!]\, K &= K[\lambda x k. [\![M]\!][\lambda M. k\, M]] \\
[\![M\, N]\!]\, K &= [\![M]\!]\, [\lambda M. [\![N]\!]\, (\lambda N. M\, N\, (\lambda v. K[v]))] \\
[\![\mathbf{let}\, x = M\, \mathbf{in}\, N]\!]\, K &= [\![M]\!]\, [\lambda M. \mathbf{let}\, x = M\, \mathbf{in}\, [\![N]\!][\lambda N. K[N]]]
\end{aligned}
$$

$$
[\![M]\!] = \lambda k. [\![M]\!]\, [\lambda M. k\, M]
$$

Quiz

Is there M s.t. $[\![M]\!] = \lambda k. k\, (\lambda x k. x)$?
What is the image of the one-pass CPS transform?

Motivation

A precise syntax for CPS terms?

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\,\cdot \;&:\; M \to (M \to M) \to M \\
[\![x]\!]\,K &= K[x] \\
[\![\lambda x.M]\!]\,K &= K[\lambda xk.\,[\![M]\!][\lambda M.\,k\,M]] \\
[\![M\,N]\!]\,K &= [\![M]\!]\,[\lambda M.\,[\![N]\!]\,(\lambda N.\,M\,N\,(\lambda v.\,K[v]))] \\
[\![\textbf{let}\;x = M\;\textbf{in}\;N]\!]\,K &= [\![M]\!]\,[\lambda M.\,\textbf{let}\;x = M\;\textbf{in}\;[\![N]\!][\lambda N.\,K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!] \;&:\; M \to M \\
[\![M]\!] &= \lambda k.\,[\![M]\!][\lambda M.\,k\,M]
\end{aligned}
$$

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\cdot \;&:\; M \to (T \to S) \to U \\
[\![x]\!]\,K &= K[x] \\
[\![\lambda x.M]\!]\,K &= K[\lambda xk.\,[\![M]\!][\lambda M.k\,M]] \\
[\![M\,N]\!]\,K &= [\![M]\!]\,[\lambda M.\,[\![N]\!]\,(\lambda N.M\,N\,(\lambda v.K[v]))] \\
[\![\mathbf{let}\,x = M\,\mathbf{in}\,N]\!]\,K &= [\![M]\!]\,[\lambda M.\,\mathbf{let}\,x = M\,\mathbf{in}\,[\![N]\!][\lambda N.K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!] \;&:\; M \to P \\
[\![M]\!] &= \lambda k.\,[\![M]\!][\lambda M.k\,M]
\end{aligned}
$$

$S ::=$

$T ::=$

$P ::=$

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\,\cdot\ &:\ M \rightarrow (T \rightarrow S) \rightarrow U \\
[\![x]\!]\,K &= K[x] \\
[\![\lambda x.M]\!]\,K &= K[\lambda xk.\,[\![M]\!][\lambda M.\,k\,M]] \\
[\![M\,N]\!]\,K &= [\![M]\!]\,[\lambda M.\,[\![N]\!]\,(\lambda N.\,M\,N\,(\lambda v.\,K[v]))] \\
[\![\mathbf{let}\ x = M\ \mathbf{in}\ N]\!]\,K &= [\![M]\!]\,[\lambda M.\,\mathbf{let}\ x = M\ \mathbf{in}\ [\![N]\!][\lambda N.\,K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!]\ &:\ M \rightarrow P \\
[\![M]\!] &= \lambda k.\,[\![M]\!][\lambda M.\,k\,M]
\end{aligned}
$$

$S ::=$

$T ::=$

$P ::=$

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\,\cdot\ &:\ M \to (T \to S) \to S \\
[\![x]\!]\,K &= K[x] \\
[\![\lambda x.\,M]\!]\,K &= K[\lambda xk.\,[\![M]\!][\lambda M.\,k\;M]] \\
[\![M\;N]\!]\,K &= [\![M]\!]\;[\lambda M.\,[\![N]\!]\;(\lambda N.\,M\;N\;(\lambda v.\,K[v]))] \\
[\![\mathbf{let}\;x = M\;\mathbf{in}\;N]\!]\,K &= [\![M]\!]\;[\lambda M.\,\mathbf{let}\;x = M\;\mathbf{in}\;[\![N]\!][\lambda N.\,K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!]\ &:\ M \to P \\
[\![M]\!] &= \lambda k.\,[\![M]\!][\lambda M.\,k\;M]
\end{aligned}
$$

$S ::=$

$T ::=$

$P ::=$

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\,\cdot\ &:\ M \to (T \to S) \to S \\
[\![x]\!]\,K &= K[x] \\
[\![\lambda x.M]\!]\,K &= K[\lambda xk.\,[\![M]\!][\lambda M.\,k\,M]] \\
[\![M\,N]\!]\,K &= [\![M]\!]\,[\lambda M.\,[\![N]\!]\,(\lambda N.\,M\,N\,(\lambda v.\,K[v]))] \\
[\![\mathbf{let}\,x = M\ \mathbf{in}\ N]\!]\,K &= [\![M]\!]\,[\lambda M.\,\mathbf{let}\,x = M\ \mathbf{in}\ [\![N]\!][\lambda N.\,K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!]\ &:\ M \to P \\
[\![M]\!] &= \lambda k.\,[\![M]\!][\lambda M.\,k\,M]
\end{aligned}
$$

$S ::=$

$T ::=$

$P ::=$

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\,\cdot\ &:\ M \to (T \to S) \to S \\
[\![x]\!]\,K &= K[x] \\
[\![\lambda x.M]\!]\,K &= K[\lambda xk.\,[\![M]\!][\lambda M.k\,M]] \\
[\![M\,N]\!]\,K &= [\![M]\!]\,[\lambda M.\,[\![N]\!]\,(\lambda N.M\,N\,(\lambda v.K[v]))] \\
[\![\mathbf{let}\,x = M\ \mathbf{in}\ N]\!]\,K &= [\![M]\!]\,[\lambda M.\,\mathbf{let}\,x = M\ \mathbf{in}\ [\![N]\!][\lambda N.K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!]\ &:\ M \to P \\
[\![M]\!] &= \lambda k.\,[\![M]\!][\lambda M.k\,M]
\end{aligned}
$$

$$
\begin{aligned}
S &::= \\
T &::= \lambda xk.S \mid x \mid v \\
P &::=
\end{aligned}
$$

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\,\cdot \; &: \; M \to (T \to S) \to S \\
[\![x]\!]\, K &= K[x] \\
[\![\lambda x.\, M]\!]\, K &= K[\lambda xk.\, [\![M]\!][\lambda M.\, k\; M]] \\
[\![M\; N]\!]\, K &= [\![M]\!]\; [\lambda M.\, [\![N]\!]\; (\lambda N.\, M\; N\; (\lambda v.\, K[v]))] \\
[\![\mathbf{let}\; x = M\; \mathbf{in}\; N]\!]\, K &= [\![M]\!]\; [\lambda M.\, \mathbf{let}\; x = M\; \mathbf{in}\; [\![N]\!][\lambda N.\, K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!] \; &: \; M \to P \\
[\![M]\!] &= \lambda k.\, [\![M]\!][\lambda M.\, k\; M]
\end{aligned}
$$

$$
\begin{aligned}
S &::= \\
T &::= \lambda xk.\, S \mid x \mid v \\
P &::=
\end{aligned}
$$

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\cdot \; &: \; M \to (T \to S) \to S \\
[\![x]\!]\,K &= K[x] \\
[\![\lambda x.M]\!]\,K &= K[\lambda xk.\,[\![M]\!][\lambda M.\,k\;M]] \\
[\![M\;N]\!]\,K &= [\![M]\!]\;[\lambda M.\,[\![N]\!]\;(\lambda N.\,M\;N\;(\lambda v.\,K[v]))] \\
[\![\mathbf{let}\;x = M\;\mathbf{in}\;N]\!]\,K &= [\![M]\!]\;[\lambda M.\,\mathbf{let}\;x = M\;\mathbf{in}\;[\![N]\!][\lambda N.\,K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!] \; &: \; M \to P \\
[\![M]\!] &= \lambda k.\,[\![M]\!][\lambda M.\,k\;M]
\end{aligned}
$$

$$
\begin{aligned}
S &::= k\;T \mid T\;T\;(\lambda v.\,S) \mid \mathbf{let}\;x = T\;\mathbf{in}\;S \\
T &::= \lambda xk.\,S \mid x \mid v \\
P &::=
\end{aligned}
$$

Analysis Output syntax of the one-pass CPS

$$\llbracket \cdot \rrbracket \cdot \; : \; M \to (T \to S) \to S$$

$$\llbracket x \rrbracket \, K = K[x]$$

$$\llbracket \lambda x. M \rrbracket \, K = K[\lambda xk. \llbracket M \rrbracket [\lambda M. k \, M]]$$

$$\llbracket M \, N \rrbracket \, K = \llbracket M \rrbracket \, [\lambda M. \llbracket N \rrbracket \, (\lambda N. M \, N \, (\lambda v. K[v]))]$$

$$\llbracket \mathbf{let} \, x = M \, \mathbf{in} \, N \rrbracket \, K = \llbracket M \rrbracket \, [\lambda M. \mathbf{let} \, x = M \, \mathbf{in} \, \llbracket N \rrbracket [\lambda N. K[N]]]$$

$$\llbracket \cdot \rrbracket \; : \; M \to P$$

$$\llbracket M \rrbracket = \lambda k. \llbracket M \rrbracket [\lambda M. k \, M]$$

$$S ::= k \; T \mid T \; T \; (\lambda v. S) \mid \mathbf{let} \, x = T \, \mathbf{in} \, S$$

$$T ::= \lambda xk. S \mid x \mid v$$

$$P ::=$$

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\ \cdot\ &:\ M \to (T \to S) \to S \\
[\![x]\!]\ K &= K[x] \\
[\![\lambda x.M]\!]\ K &= K[\lambda xk.\,[\![M]\!][\lambda M.\,k\ M]] \\
[\![M\ N]\!]\ K &= [\![M]\!]\ [\lambda M.\,[\![N]\!]\ (\lambda N.\,M\ N\ (\lambda v.\,K[v]))] \\
[\![\mathbf{let}\ x = M\ \mathbf{in}\ N]\!]\ K &= [\![M]\!]\ [\lambda M.\,\mathbf{let}\ x = M\ \mathbf{in}\ [\![N]\!][\lambda N.\,K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!]\ &:\ M \to P \\
[\![M]\!] &= \lambda k.\,[\![M]\!][\lambda M.\,k\ M]
\end{aligned}
$$

$$
\begin{aligned}
S &::= k\ T \mid T\ T\ (\lambda v.\,S) \mid \mathbf{let}\ x = T\ \mathbf{in}\ S \\
T &::= \lambda xk.\,S \mid x \mid v \\
P &::= \lambda k.\,S
\end{aligned}
$$

Analysis Output syntax of the one-pass CPS

$$
\begin{aligned}
[\![\cdot]\!]\ \cdot\ &:\ M \to (T \to S) \to S \\
[\![x]\!]\ K &= K[x] \\
[\![\lambda x.\,M]\!]\ K &= K[\lambda x k.\,[\![M]\!][\lambda M.\,k\ M]] \\
[\![M\ N]\!]\ K &= [\![M]\!]\ [\lambda M.\,[\![N]\!]\ (\lambda N.\,M\ N\ (\lambda v.\,K[v]))] \\
[\![\textbf{let}\ x = M\ \textbf{in}\ N]\!]\ K &= [\![M]\!]\ [\lambda M.\,\textbf{let}\ x = M\ \textbf{in}\ [\![N]\!][\lambda N.\,K[N]]]
\end{aligned}
$$

$$
\begin{aligned}
[\![\cdot]\!]\ &:\ M \to P \\
[\![M]\!] &= \lambda k.\,[\![M]\!][\lambda M.\,k\ M]
\end{aligned}
$$

$S ::= k\ T \mid T\ T\ (\lambda v.\,S) \mid \textbf{let}\ x = T\ \textbf{in}\ S$	Serious terms
$T ::= \lambda x k.\,S \mid x \mid v$	Trival terms
$P ::= \lambda k.\,S$	Programs

Result The syntax of CPS terms

$S ::= k\ T \mid T\ T\ (\lambda v.\, S) \mid \textbf{let}\ x = T\ \textbf{in}\ S$	Serious terms
$T ::= \lambda xk.\, S \mid x \mid v$	Trival terms
$P ::= \lambda k.\, S$	Programs

Result The syntax of CPS terms

$$
\begin{aligned}
S &::= k\ T \mid T\ T\ (\lambda v.\ S) \mid \textbf{let}\ x = T\ \textbf{in}\ S && \text{Serious terms} \\
T &::= \lambda xk.\ S \mid x \mid v && \text{Trival terms} \\
P &::= \lambda k.\ S && \text{Programs}
\end{aligned}
$$

Notes

- distinguished x (source), v (value), k (continuation) var.
- $(\lambda v.\ S)$ is a *continuation*
- programs await the *initial* continuation

Result The syntax of CPS terms

$$S ::= \mathbf{ret}_k\ T \mid \mathbf{bind}\ v = T\ T\ \mathbf{in}\ S \mid \mathbf{let}\ x = T\ \mathbf{in}\ S \qquad \text{Serious terms}$$

$$T ::= \lambda xk.\, S \mid x \mid v \qquad \text{Trival terms}$$

$$P ::= \lambda k.\, S \qquad \text{Programs}$$

Notes

- distinguished x (source), v (value), k (continuation) var.
- $(\lambda v.\, S)$ is a *continuation*
- programs await the *initial* continuation
- monadic operations

Result The typing of CPS terms

$\boxed{\Gamma \vdash S \mid \Delta}$

$$\textsc{Decide}\quad \frac{\Gamma \vdash T : A \mid \Delta}{\Gamma \vdash k\ T \mid \Delta, k : A} \qquad \cdots \qquad \textsc{Cut}\quad \frac{\Gamma \vdash T : A \mid \Delta \qquad \Gamma, x : A \vdash S \mid \Delta}{\Gamma \vdash \mathbf{let}\ x = T\ \mathbf{in}\ S \mid \Delta}$$

Result The typing of CPS terms

$\boxed{\Gamma \vdash S \mid \Delta}$

$$\textsc{Decide}\ \frac{\Gamma \vdash T : A \mid \Delta}{\Gamma \vdash k\ T \mid \Delta, k : A} \qquad \cdots \qquad \textsc{Cut}\ \frac{\Gamma \vdash T : A \mid \Delta \qquad \Gamma, x : A \vdash S \mid \Delta}{\Gamma \vdash \textbf{let}\ x = T\ \textbf{in}\ S \mid \Delta}$$

$\boxed{\Gamma \vdash T : A \mid \Delta}$

$$\textsc{ImplR}\ \frac{\Gamma, x : A \vdash S \mid \Delta, k : B}{\Gamma \vdash \lambda xk.S : A \to B \mid \Delta} \qquad \textsc{Init}\ \frac{}{\Gamma, x : A \vdash x : A \mid \Delta}$$

Result The typing of CPS terms

$\boxed{\Gamma \vdash S \mid \Delta}$

$$\textsc{Decide}\ \frac{\Gamma \vdash T : A \mid \Delta}{\Gamma \vdash k\ T \mid \Delta, k : A} \qquad \cdots \qquad \textsc{Cut}\ \frac{\Gamma \vdash T : A \mid \Delta \qquad \Gamma, x : A \vdash S \mid \Delta}{\Gamma \vdash \mathbf{let}\ x = T\ \mathbf{in}\ S \mid \Delta}$$

$\boxed{\Gamma \vdash T : A \mid \Delta}$

$$\textsc{ImplR}\ \frac{\Gamma, x : A \vdash S \mid \Delta, k : B}{\Gamma \vdash \lambda xk.S : A \to B \mid \Delta} \qquad \textsc{Init}\ \frac{}{\Gamma, x : A \vdash x : A \mid \Delta}$$

- Γ contains values, Δ contains continuations
- Focused and unfocused judgments
- Classical reasoning

Problem β-redexes or **let**s?

$$
\llbracket (\lambda xy.\, x)\ a\ b \rrbracket = \\
\lambda k.\, (\lambda xk.\, k\ (\lambda yk.\, k\ x))\ a\ (\lambda v.\, v\ b\ (\lambda w.\, k\ w))
$$

Problem β-redexes or **let**s?

$$[\![(\textbf{let}\, x = a \textbf{ in } \lambda y.x)\; b]\!] = \lambda k.\, \textbf{let}\, x = a \textbf{ in } (\lambda y.x)\; b\; (\lambda v.kv)$$

Problem β-redexes or **let**s?

$$[\![\textbf{let}\ x = a\ \textbf{in}\ (\lambda y.x)\ b]\!] =$$
$$\lambda k.\,\textbf{let}\ x = a\ \textbf{in}\ (\lambda y.x)\ b\ (\lambda v.\,kv)$$

Problem β-redexes or **let**s?

$$
\begin{aligned}
&[\![\textbf{let}\ x = a\ \textbf{in}\ \textbf{let}\ y = b\ \textbf{in}\ x]\!] = \\
&\qquad\qquad \lambda k.\,\textbf{let}\ x = a\ \textbf{in}\ \textbf{let}\ y = b\ \textbf{in}\ k\ x
\end{aligned}
$$

Problem β-redexes or **lets**?

$$[\![\mathbf{let}\ x = a\ \mathbf{in}\ \mathbf{let}\ y = b\ \mathbf{in}\ x]\!] = \lambda k.\,\mathbf{let}\ x = a\ \mathbf{in}\ \mathbf{let}\ y = b\ \mathbf{in}\ k\ x$$

Remarks

- two representations for redexes in CPS terms (β redexes and **let**)
- **let** gives more compact CPS terms
- let's turn *nested* β-redexes into **let**s!

Problem β-redexes or **let**s?

$$[\![\mathbf{let}\ x = a\ \mathbf{in}\ \mathbf{let}\ y = b\ \mathbf{in}\ x]\!] = \lambda k.\,\mathbf{let}\ x = a\ \mathbf{in}\ \mathbf{let}\ y = b\ \mathbf{in}\ k\ x$$

Remarks

- two representations for redexes in CPS terms (β redexes and **let**)
- **let** gives more compact CPS terms
- let's turn *nested* β-redexes into **let**s!

Motivation

More compact CPS terms [Sabry & Felleisen, 1993; Danvy 2004]

Problem β-redexes or **let**s?

$$[\![\mathbf{let}\, x = a \,\mathbf{in}\, \mathbf{let}\, y = b \,\mathbf{in}\, x]\!] = \lambda k.\, \mathbf{let}\, x = a \,\mathbf{in}\, \mathbf{let}\, y = b \,\mathbf{in}\, k\, x$$

Remarks

- two representations for redexes in CPS terms (β redexes and **let**)
- **let** gives more compact CPS terms
- let's turn *nested* β-redexes into **let**s!

Motivation

More compact CPS terms [Sabry & Felleisen, 1993; Danvy 2004]

Proposition

Nested redexes → **let**s, then CPS-transformation (2-pass)?

Problem β-redexes or **let**s?

$$[\![\textbf{let}\ x = a\ \textbf{in}\ \textbf{let}\ y = b\ \textbf{in}\ x]\!] = \lambda k.\,\textbf{let}\ x = a\ \textbf{in}\ \textbf{let}\ y = b\ \textbf{in}\ k\ x$$

Remarks

- two representations for redexes in CPS terms (β redexes and **let**)
- **let** gives more compact CPS terms
- let's turn *nested* β-redexes into **let**s!

Motivation

More compact CPS terms [Sabry & Felleisen, 1993; Danvy 2004]

Proposition

Nested redexes → **let**s, then CPS-transformation (2-pass)?
How to distinguish original and transformed **let**s?

Analysis The syntax of β-normal CPS terms

$S ::= k\ T \mid T\ T\ (\lambda v.\, S) \mid \textbf{let}\ x = T\ \textbf{in}\ S$	Serious terms
$T ::= \lambda x k.\, S \mid x \mid v$	Trival terms
$P ::= \lambda k.\, S$	Programs

Analysis The syntax of β-normal CPS terms

$$S ::= k\ T \mid T\ T\ (\lambda v.\, S) \mid \textbf{let}\ x = T\ \textbf{in}\ S \qquad \text{Serious terms}$$

$$T ::= \lambda x k.\, S \mid x \mid v \qquad \text{Trival terms}$$

$$P ::= \lambda k.\, S \qquad \text{Programs}$$

Analysis The syntax of β-normal CPS terms

$S ::= k\ T \mid I\ T\ (\lambda v.\ S) \mid \textbf{let}\ x = T\ \textbf{in}\ S$	Serious terms
$T ::= \lambda xk.\ S \mid I$	Trival terms
$I ::= x \mid v$	Identifiers
$P ::= \lambda k.\ S$	Programs

Analysis The syntax of β-normal CPS terms

$$S ::= k\ T \mid I\ T\ (\lambda v.\, S) \mid \textbf{let}\ x = T\ \textbf{in}\ S$$ Serious terms

$$T ::= \lambda xk.\, S \mid I$$ Trival terms

$$I ::= x \mid v$$ Identifiers

$$P ::= \lambda k.\, S$$ Programs

Remarks

- identifiers = "atomic terms"

Analysis The syntax of β-normal CPS terms

$S ::= k\ T \mid I\ T\ (\lambda v.\,S) \mid \mathbf{let}\ x = T\ \mathbf{in}\ S$	Serious terms
$T ::= \lambda xk.\,S \mid I$	Trival terms
$I ::= x \mid v$	Identifiers
$P ::= \lambda k.\,S$	Programs

Remarks

- identifiers = "atomic terms"
- CPS is now context-sensitive

Analysis The syntax of β-normal CPS terms

$$
\begin{aligned}
S &::= \mathbf{ret}_k\ T \mid \mathbf{bind}\ v = I\ T\ \mathbf{in}\ S \mid \mathbf{let}\ x = T\ \mathbf{in}\ S && \text{Serious terms}\\
T &::= \lambda x k.\, S \mid I && \text{Trival terms}\\
I &::= x \mid v && \text{Identifiers}\\
P &::= \lambda k.\, S && \text{Programs}
\end{aligned}
$$

Remarks

- identifiers = “atomic terms”
- CPS is now context-sensitive
- Monadic/Administrative Normal Forms [Flanagan et al., 1993]

Analysis The syntax of β-normal CPS terms

$$
\begin{array}{ll}
S ::= \mathbf{ret}_k\ T \mid \mathbf{bind}\ v = I\ T\ \mathbf{in}\ S \mid \mathbf{let}\ x = T\ \mathbf{in}\ S & \text{Serious terms} \\
T ::= \lambda xk.\, S \mid I & \text{Trival terms} \\
I ::= x \mid v & \text{Identifiers} \\
P ::= \lambda k.\, S & \text{Programs}
\end{array}
$$

Remarks

- identifiers = "atomic terms"
- CPS is now context-sensitive
- Monadic/Administrative Normal Forms [Flanagan et al., 1993]

Example

$$
\begin{aligned}
[\![g\ (f\ x)]\!] = \lambda k.\, & \mathbf{bind}\ v_1 = f\ x\ \mathbf{in} \\
& \mathbf{bind}\ v_2 = g\ v_1\ \mathbf{in} \\
& \mathbf{ret}_k\ v_2
\end{aligned}
$$

Result CPS transformation of β-redexes (Danvy, 2004)

$$
\begin{aligned}
[\![x]\!]\, K &= K[x] \\
[\![\lambda x.\, M]\!]\, K &= K[\lambda xk.\, [\![M]\!][\lambda M.\, k\, M]] \\
[\![M\, N]\!]\, K &= [\![M]\!]\, [\lambda M.\, [\![N]\!]\, (\lambda N.\, M\, N\, (\lambda v.\, K[v]))] \\
[\![\mathbf{let}\; x = M\; \mathbf{in}\; N]\!]\, K &= [\![M]\!]\, [\lambda M.\, \mathbf{let}\; x = M\; \mathbf{in}\; [\![N]\!][\lambda N.\, K[N]]]
\end{aligned}
$$

Result CPS transformation of β-redexes (Danvy, 2004)

$$\llbracket x \rrbracket_l\, K = K[\psi_l(x)]$$
$$\llbracket \lambda x.M \rrbracket_0\, K = K[\lambda xk.\llbracket M \rrbracket_0[\lambda T.k\, T]]$$

$$\llbracket M\, N \rrbracket_l\, K = \llbracket M \rrbracket_{S(l)}[\lambda T.\llbracket N \rrbracket_l[\lambda U.T[U][\lambda V.K[V]]]]$$
$$\llbracket \mathbf{let}\; x = M \;\mathbf{in}\; N \rrbracket_l\, K = K[\lambda TK.\mathbf{let}\; x = T \;\mathbf{in}\; \llbracket M \rrbracket_l[\lambda M.K[M]]]$$

$$\psi_0(I) = i$$

Result CPS transformation of β-redexes (Danvy, 2004)

$$
\begin{aligned}
[\![x]\!]_l\, K &= K[\psi_l(x)] \\
[\![\lambda x.M]\!]_0\, K &= K[\lambda xk.\,[\![M]\!]_0[\lambda T.\,k\; T]] \\
[\![\lambda x.M]\!]_{S(l)}\, K &= K[\lambda TK.\,\mathbf{let}\; x = T\;\mathbf{in}\;[\![M]\!]_l[\lambda M.\,K[M]]] \\
[\![M\; N]\!]_l\, K &= [\![M]\!]_{S(l)}[\lambda T.\,[\![N]\!]_l[\lambda U.\,T[U][\lambda V.\,K[V]]]] \\
[\![\mathbf{let}\; x = M\;\mathbf{in}\; N]\!]_l\, K &= K[\lambda TK.\,\mathbf{let}\; x = T\;\mathbf{in}\;[\![M]\!]_l[\lambda M.\,K[M]]]
\end{aligned}
$$

$$\psi_0(I) = i$$

Result CPS transformation of β-redexes (Danvy, 2004)

$$
\begin{aligned}
[\![x]\!]_l\, K &= K[\psi_l(x)] \\
[\![\lambda x.\, M]\!]_0\, K &= K[\lambda xk.\, [\![M]\!]_0[\lambda T.\, k\, T]] \\
[\![\lambda x.\, M]\!]_{S(l)}\, K &= K[\lambda TK.\, \textbf{let}\; x = T\; \textbf{in}\; [\![M]\!]_l[\lambda M.\, K[M]]] \\
[\![M\, N]\!]_l\, K &= [\![M]\!]_{S(l)}[\lambda T.\, [\![N]\!]_l[\lambda U.\, T[U][\lambda V.\, K[V]]]] \\
[\![\textbf{let}\; x = M\; \textbf{in}\; N]\!]_l\, K &= K[\lambda TK.\, \textbf{let}\; x = T\; \textbf{in}\; [\![M]\!]_l[\lambda M.\, K[M]]]
\end{aligned}
$$

$$
\begin{aligned}
\psi_0(I) &= i \\
\psi_{S(l)} &= \lambda TK.\, IT(\lambda v.\, K[\psi_l(v)])
\end{aligned}
$$

Result CPS transformation of β-redexes (Danvy, 2004)

$$[\![\cdot]\!]_{\cdot}\ \cdot : \forall l : \mathbb{N}, M \to (\tau_l \to S) \to S$$

$$\begin{aligned}
[\![x]\!]_l\, K &= K[\psi_l(x)] \\
[\![\lambda x.M]\!]_0\, K &= K[\lambda xk.[\![M]\!]_0[\lambda T.k\,T]] \\
[\![\lambda x.M]\!]_{S(l)}\, K &= K[\lambda TK.\mathbf{let}\ x = T\ \mathbf{in}\ [\![M]\!]_l[\lambda M.K[M]]] \\
[\![M\,N]\!]_l\, K &= [\![M]\!]_{S(l)}[\lambda T.[\![N]\!]_l[\lambda U.T[U][\lambda V.K[V]]]] \\
[\![\mathbf{let}\ x = M\ \mathbf{in}\ N]\!]_l\, K &= K[\lambda TK.\mathbf{let}\ x = T\ \mathbf{in}\ [\![M]\!]_l[\lambda M.K[M]]]
\end{aligned}$$

$$\psi_{\cdot}(\cdot) : \forall l : \mathbb{N}, I \to \tau_l$$

$$\begin{aligned}
\psi_0(I) &= i \\
\psi_{S(l)} &= \lambda TK.IT(\lambda v.K[\psi_l(v)])
\end{aligned}$$

Result CPS transformation of β-redexes (Danvy, 2004)

$$
\begin{aligned}
\tau_0 &= T \\
\tau_{S(l)} &= T \to (\tau_l \to S) \to S \\
[\![\cdot]\!]_. \cdot &: \forall l : \mathbb{N}, M \to (\tau_l \to S) \to S
\end{aligned}
$$

$$
\begin{aligned}
[\![x]\!]_l\, K &= K[\psi_l(x)] \\
[\![\lambda x. M]\!]_0\, K &= K[\lambda xk. [\![M]\!]_0[\lambda T. k\, T]] \\
[\![\lambda x. M]\!]_{S(l)}\, K &= K[\lambda TK. \textbf{let}\ x = T\ \textbf{in}\ [\![M]\!]_l[\lambda M. K[M]]] \\
[\![M\, N]\!]_l\, K &= [\![M]\!]_{S(l)}[\lambda T. [\![N]\!]_l[\lambda U. T[U][\lambda V. K[V]]]] \\
[\![\textbf{let}\ x = M\ \textbf{in}\ N]\!]_l\, K &= K[\lambda TK. \textbf{let}\ x = T\ \textbf{in}\ [\![M]\!]_l[\lambda M. K[M]]]
\end{aligned}
$$

$$
\begin{aligned}
\psi_.(\cdot) &: \forall l : \mathbb{N}, I \to \tau_l \\
\psi_0(I) &= i \\
\psi_{S(l)} &= \lambda TK. IT(\lambda v. K[\psi_l(v)])
\end{aligned}
$$

Result The typing of β-normal CPS terms

$$\Gamma \vdash S \mid \Delta$$

DECIDE

$$\frac{\Gamma \vdash T : A \mid \Delta}{\Gamma \vdash k\ T \mid \Delta, k : A}$$

IMPLL

$$\frac{\Gamma \vdash U : A \mid \Delta \qquad \Gamma, v : B \vdash S \mid \Delta}{\Gamma, I : A \to B \vdash I\ U\ (\lambda v. S) \mid \Delta}$$

CUT

$$\frac{\Gamma \vdash T : A \mid \Delta \qquad \Gamma, x : A \vdash S \mid \Delta}{\Gamma \vdash \textbf{let}\ x = T\ \textbf{in}\ S \mid \Delta}$$

$$\Gamma \vdash T : A \mid \Delta$$

IMPLR

$$\frac{\Gamma, x : A \vdash S \mid \Delta, k : B}{\Gamma \vdash \lambda x k. S : A \to B \mid \Delta}$$

INIT

$$\frac{}{\Gamma, I : A \vdash I : A \mid \Delta}$$

End result The LKQ focused sequent calculus [DJS, 1993]

$\Gamma \vdash S \mid \Delta$

$$\text{DECIDE}\quad \frac{\Gamma \vdash A \mid \Delta}{\Gamma \vdash \Delta, A} \qquad \text{IMPLL}\quad \frac{\Gamma \vdash A \mid \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}$$

$$\text{CUT}\quad \frac{\Gamma \vdash A \mid \Delta \qquad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}$$

$\Gamma \vdash T : A \mid \Delta$

$$\text{IMPLR}\quad \frac{\Gamma, A \vdash \Delta, B}{\Gamma \vdash A \to B \mid \Delta} \qquad \text{INIT}\quad \frac{}{\Gamma, A \vdash A \mid \Delta}$$

And finally From classical to intuitionistic: LJQ

Remark [Danvy & Pfenning, 1999]

Without control operators (`call/cc`), only one continuation variable k is ever needed.

And finally From classical to intuitionistic: LJQ

Remark [Danvy & Pfenning, 1999]

Without control operators (`call/cc`), only one continuation variable k is ever needed.

Corollary

$\Delta ::= k : A$

DECIDE is trivial.

And finally From classical to intuitionistic: LJQ

Remark [Danvy & Pfenning, 1999]

Without control operators (`call/cc`), only one continuation variable k is ever needed.

Corollary

$\Delta ::= k : A$

DECIDE is trivial.

$\rightsquigarrow$ *We recover LJQ*

Conclusion

To sum up

We reconstructed LJQ out of a fine analysis of CPS:

- control flow $\rightsquigarrow$ *one-pass* CPS
- syntax aggregation $\rightsquigarrow$ typed, dedicated syntax
- reduction restriction $\rightsquigarrow$ β-normal syntax

Conclusion

To sum up

We reconstructed LJQ out of a fine analysis of CPS:

- control flow $\rightsquigarrow$ *one-pass* CPS
- syntax aggregation $\rightsquigarrow$ typed, dedicated syntax
- reduction restriction $\rightsquigarrow$ β-normal syntax

The lessons learned

- The syntax of β-normal CPS terms is the ANF
- Its typing corresponds to LKQ

Conclusion

To sum up

We reconstructed LJQ out of a fine analysis of CPS:

- control flow $\rightsquigarrow$ *one-pass* CPS
- syntax aggregation $\rightsquigarrow$ typed, dedicated syntax
- reduction restriction $\rightsquigarrow$ β-normal syntax

The lessons learned

- The syntax of β-normal CPS terms is the ANF
- Its typing corresponds to LKQ

The future article

- *one-pass*, *β-normal*, *tail-recursive* CPS in a dedicated syntax.
- methodology to infer typing rules using OCaml/GADTs

Overall conclusion

1. From NJ to LJT by program transformations
2. From CPS to LJQ by program analyses

Overall conclusion

1. From NJ to LJT by program transformations
2. From CPS to LJQ by program analyses

Are these coincidences?
Are these manifestations of the same thing?

Overall conclusion

1. From NJ to LJT by program transformations
2. From CPS to LJQ by program analyses

Are these coincidences?
Are these manifestations of the same thing?

Further work

3. From ? to LKF by ?

Overall conclusion

1. From NJ to LJT by program transformations
2. From CPS to LJQ by program analyses

Are these coincidences?
Are these manifestations of the same thing?

Further work

3. From ? to LKF by ?

Goal

Understand *focusing* through *program transformation*

Overall conclusion

1. From NJ to LJT by program transformations
2. From CPS to LJQ by program analyses

Are these coincidences?
Are these manifestations of the same thing?

Further work

3. From ? to LKF by ?

Lifetime goal

Understand *proof theory* through *compilation*