Multiple Prompts

In little steps

Paul Downen

Shonan 203: Effect Handlers & General Purpose Languages

Tuesday, September 26, 2023 — Implementor-Facing Aspects

Why Multiple Prompts?

Avoid crosstalk

try

$lookup(dict, total) / lookup(dict, count)$

catch $KeyNotFound(err) \Rightarrow 0$

Why Multiple Prompts?

Avoid crosstalk

$$\begin{array}{l}\textbf{try}\\ \quad \mathit{lookup}(\mathit{dict}, \mathit{total}) / \mathit{lookup}(\mathit{dict}, \mathit{count})\\ \textbf{catch}\ \mathit{KeyNotFound}(\mathit{err}) \Rightarrow 0\end{array}$$

$$\begin{array}{l}\textbf{handle}\\ \quad \left(\begin{array}{l}\textbf{handle}\\ \quad \mathit{Print}(\texttt{"Hello"})\\ \textbf{with}\ \mathit{Get}(), k \Rightarrow \ldots\\ \qquad\quad \mathit{Put}(x), k \Rightarrow \ldots\end{array}\right)\\ \textbf{with}\ \mathit{Print}(\mathit{msg}), k \Rightarrow \ldots\end{array}$$

A Systematic Approach

To build up to multiple prompts

Goal: Reason flexibly about complex control flow

- For optimizing code ahead of time
- For comparing equality of programs
- For understanding the tradeoffs in design decisions

Idea: Break down big operations into little pieces

- Modular operators combined into more familiar operators
- Small, localized, fine-grained reduction steps

Approach: model in CPS, reflect back to source

- Can point out missing concepts
- Can justify that theory is sound
- Gives an effective implementation method

The Basic CPS

Of the call-by-value λ-calculus

$$\mathit{Value} \ni V ::= \lambda x.\, M \mid x \qquad \mathit{Term} \ni M, N ::= V \mid M\, N$$

$$(\lambda x.\, M)\; V \mapsto M\{V/x\}$$

$$\begin{aligned}
\mathcal{C}[\![x]\!] &= \lambda\alpha.\, \alpha\; x \\
\mathcal{C}[\![\lambda x.\, M]\!] &= \lambda\alpha.\, \alpha\, (\lambda x.\, \mathcal{C}[\![M]\!]) \\
\mathcal{C}[\![M\, N]\!] &= \lambda\alpha.\, \mathcal{C}[\![M]\!]\; \lambda f.\, \mathcal{C}[\![N]\!]\; \lambda x.\, f\; x\; \alpha
\end{aligned}$$

Writing the Initial Continuation

In the source

The target CPS language:

$$
\begin{aligned}
\mathit{CPSValue} \ni \quad V_{val} &::= \lambda x.\, V_{trm} \mid x \\
\mathit{CPSTerm} \ni \quad V_{trm} &::= \lambda \alpha.\, M_{com} \\
\mathit{CPSContinuation} \ni \quad V_{knt} &::= \lambda x.\, M_{com} \mid \alpha \\
\mathit{CPSCommand} \ni M_{com} &::= V_{trm}\ V_{knt} \mid V_{knt}\ V_{val} \mid V_{val}\ V_{val}\ V_{knt}
\end{aligned}
$$

$$\mathcal{C}[\![M]\!] \in \mathit{CPSTerm}$$

CPS Terms are inert. Only CPS Commands can run

Writing the Initial Continuation

In the source

The target CPS language:

$$
\begin{aligned}
\mathit{CPSValue} \ni \quad V_{val} &::= \lambda x.\, V_{trm} \mid x \\
\mathit{CPSTerm} \ni \quad V_{trm} &::= \lambda \alpha.\, M_{com} \\
\mathit{CPSContinuation} \ni \quad V_{knt} &::= \lambda x.\, M_{com} \mid \alpha \mid \lambda x.\, x \\
\mathit{CPSCommand} \ni M_{com} &::= V_{trm}\ V_{knt} \mid V_{knt}\ V_{val} \mid V_{val}\ V_{val}\ V_{knt}
\end{aligned}
$$

$$\mathcal{C}[\![M]\!] \in \mathit{CPSTerm}$$

CPS Terms are inert. Only CPS Commands can run

$$\mathcal{C}[\![\langle M \| \mathbf{tp} \rangle]\!] = \mathcal{C}[\![M]\!]\ (\lambda x.\, x) \in \mathit{CPSCommand}$$

Adding Classical Control

The μ of Parigot's $\lambda\mu$

$$\begin{aligned} \mathit{Value} \ni \quad V &::= \lambda x.\, M \mid x & \mathit{Continuation} \ni q &::= \alpha \mid \mathbf{tp} \\ \mathit{Term} \ni M, N &::= V \mid M\,N \mid \mu\alpha.c & \mathit{Command} \ni c &::= \langle M \| q \rangle \end{aligned}$$

$$E ::= \Box \mid E\,M \mid V\,E$$

$$\langle E[\mu\alpha.c] \| q \rangle \mapsto c\{\langle E[M] \| q \rangle / \langle M \| \alpha \rangle\}$$

$$\begin{aligned} \mathcal{C}[\![\alpha]\!] &= \alpha \\ \mathcal{C}[\![\mathbf{tp}]\!] &= \lambda x.\, x \\ \mathcal{C}[\![\mu\alpha.c]\!] &= \lambda\alpha.\, \mathcal{C}[\![c]\!] \\ \mathcal{C}[\![\langle M \| q \rangle]\!] &= \mathcal{C}[\![M]\!]\ \mathcal{C}[\![q]\!] \end{aligned}$$

Delimited Control à la shift

$\widehat{\mathrm{tp}}$ is like the "top-level" **tp**, but it can be rebound

$$\langle E[\mu\widehat{\mathrm{tp}}.\langle\widehat{\mathrm{tp}}\|V\rangle]\|q\rangle \mapsto \langle E[V]\|q\rangle$$

$$\mathcal{C}\left[\!\left[\widehat{\mathrm{tp}}\right]\!\right] = \lambda x.\, x$$
$$\mathcal{C}\left[\!\left[\mu\widehat{\mathrm{tp}}.c\right]\!\right] = \lambda\alpha.\, \alpha\,(\mathcal{C}[\![c]\!])$$

Delimited Control à la shift

Via a Dynamically-Rebindable "top-level" $\widehat{\mathrm{tp}}$

$\widehat{\mathrm{tp}}$ is like the "top-level" **tp**, but it can be rebound

$$\langle E[\mu\widehat{\mathrm{tp}}.\langle\widehat{\mathrm{tp}}\|V\rangle]\|q\rangle \mapsto \langle E[V]\|q\rangle$$

$$\mathcal{C}[\![\widehat{\mathrm{tp}}]\!] = \lambda x.\, x$$
$$\mathcal{C}[\![\mu\widehat{\mathrm{tp}}.c]\!] = \lambda\alpha.\, \alpha\,(\mathcal{C}[\![c]\!])$$

Oops… $\alpha\,(\mathcal{C}[\![c]\!])$ is not in CPS anymore!

Delimited Control à la shift

Via a Dynamically-Rebindable "top-level" $\widehat{\mathrm{tp}}$

$\widehat{\mathrm{tp}}$ is like the "top-level" **tp**, but it can be rebound

$$\langle E[\mu\widehat{\mathrm{tp}}.\langle\widehat{\mathrm{tp}}\|V\rangle]\|q\rangle \mapsto \langle E[V]\|q\rangle$$

$$\mathcal{C}\left[\!\left[\widehat{\mathrm{tp}}\right]\!\right] = \lambda x.\, x$$

$$\mathcal{C}\left[\!\left[\mu\widehat{\mathrm{tp}}.c\right]\!\right] = \lambda\alpha.\, \alpha\,(\mathcal{C}[\![c]\!])$$

Oops... $\alpha\,(\mathcal{C}[\![c]\!])$ is not in CPS anymore!

No problem, just CPS again!

Another "Meta" Continuation

From "double-barrel" CPS

$$\mathcal{C}^2[\![\_]\!] = \mathcal{C}[\![\mathcal{C}[\![\_]\!]]\!]$$

$$\begin{aligned}
\mathcal{C}^2\left[\!\left[\widehat{\mathsf{tp}}\right]\!\right] &= \mathcal{C}[\![\lambda x.\, x]\!] &&= \lambda x.\, \lambda\gamma.\, \gamma x \\
\mathcal{C}^2\left[\!\left[\mu\widehat{\mathsf{tp}}.c\right]\!\right] &= \mathcal{C}[\![\lambda\alpha.\, \alpha\, (\mathcal{C}[\![c]\!])]\!] &&= \lambda\alpha.\, \lambda\gamma.\, \mathcal{C}^2[\![c]\!]\ \lambda x.\, \alpha\ x\ \gamma \\
\mathcal{C}^2[\![\langle M \| q\rangle]\!] &= \mathcal{C}[\![\mathcal{C}[\![M]\!]\ \mathcal{C}[\![q]\!]]\!] &&= \lambda\gamma.\, \mathcal{C}^2[\![M]\!]\ \mathcal{C}^2[\![q]\!]\ \gamma
\end{aligned}$$

Another "Meta" Continuation

From "double-barrel" CPS

$$\mathcal{C}^2[\![\_]\!] = \mathcal{C}[\![\mathcal{C}[\![\_]\!]]\!]$$

$$\begin{aligned}
\mathcal{C}^2\left[\!\left[\widehat{\mathrm{tp}}\right]\!\right] &= \mathcal{C}[\![\lambda x.\, x]\!] &&= \lambda x.\, \lambda\gamma.\, \gamma x \\
\mathcal{C}^2\left[\!\left[\mu\widehat{\mathrm{tp}}.c\right]\!\right] &= \mathcal{C}[\![\lambda\alpha.\, \alpha\ (\mathcal{C}[\![c]\!])]\!] &&= \lambda\alpha.\, \lambda\gamma.\, \mathcal{C}^2[\![c]\!]\ \lambda x.\, \alpha\ x\ \gamma \\
\mathcal{C}^2[\![\langle M \| q \rangle]\!] &= \mathcal{C}[\![\mathcal{C}[\![M]\!]\ \mathcal{C}[\![q]\!]]\!] &&= \lambda\gamma.\, \mathcal{C}^2[\![M]\!]\ \mathcal{C}^2[\![q]\!]\ \gamma
\end{aligned}$$

Now (first-level) commands are inert.

Need a second level of initial continuation $\langle c \| q^2 \rangle$ to run

$$\mathcal{C}^2\left[\!\left[\langle c \| q^2 \rangle\right]\!\right] = \mathcal{C}^2[\![c]\!]\ \mathcal{C}^2\left[\!\left[q^2\right]\!\right]$$

Multiple Prompts: First Attempt

Idea: (second-level) meta-continuation γ is an environment, mapping many rebindable "top-levels" to (first-level) continuations

$$\langle E[\mu\hat{\alpha}.\langle V \| \hat{\alpha}\rangle] \| q\rangle \mapsto \langle E[V] \| q\rangle$$
$$\langle E[\mu\hat{\beta}.\langle V \| \hat{\alpha}\rangle] \| q\rangle \mapsto \langle V \| \hat{\alpha}\rangle$$

$$\widehat{\mathcal{C}}[\![\hat{\alpha}]\!] = \lambda x.\, \lambda\gamma.\, \gamma(\hat{\alpha})\; x$$
$$\widehat{\mathcal{C}}[\![\mu\hat{\alpha}.c]\!] = \lambda\beta.\, \lambda\gamma.\, \widehat{\mathcal{C}}[\![c]\!]\; (\gamma\{\hat{\alpha} \mapsto \beta\})$$

Multiple Prompts: First Attempt

Destroy the trail through the meta-continuation ☹

Idea: (second-level) meta-continuation γ is an environment, mapping many rebindable "top-levels" to (first-level) continuations

$$\langle E[\mu\hat{\alpha}.\langle V \| \hat{\alpha}\rangle] \| q\rangle \mapsto \langle E[V] \| q\rangle$$

$$\langle E[\mu\hat{\beta}.\langle V \| \hat{\alpha}\rangle] \| q\rangle \mapsto \langle V \| \hat{\alpha}\rangle \qquad \langle E \| q\rangle \text{ gone forever!}$$

$$\widehat{\mathcal{C}}[\![\hat{\alpha}]\!] = \lambda x.\, \lambda\gamma.\, \gamma(\hat{\alpha})\; x$$

$$\widehat{\mathcal{C}}[\![\mu\hat{\alpha}.c]\!] = \lambda\beta.\, \lambda\gamma.\, \widehat{\mathcal{C}}[\![c]\!]\; (\gamma\{\hat{\alpha} \mapsto \beta\})$$

Delimited Control à la shift_0

Passing through the "top-level" $\widehat{\mathrm{tp}}$

shift's prompt only lets values through; shift_0 passes through

$$\mu\widehat{\mathrm{tp}}.\langle V \| \widehat{\mathrm{tp}}\rangle \to V \qquad\qquad \mu\widehat{\mathrm{tp}}.\langle \widehat{\mathrm{tp}} \uparrow M\rangle \to M$$

$$\begin{aligned}
\mathcal{C}\!\uparrow\![\![\mu\widehat{\mathrm{tp}}.c]\!] &= \mathcal{C}\!\uparrow\![\![c]\!] \\
\mathcal{C}\!\uparrow\![\![\widehat{\mathrm{tp}}]\!] &= \lambda x.\,\lambda\gamma.\,\gamma\; x \\
\mathcal{C}\!\uparrow\![\![\langle\widehat{\mathrm{tp}} \uparrow M\rangle]\!] &= \mathcal{C}\!\uparrow\![\![M]\!]
\end{aligned}$$

$$\begin{aligned}
\mathcal{C}\!\uparrow\![\![\mu\widehat{\mathrm{tp}}.\langle M \| \widehat{\mathrm{tp}}\rangle]\!]\;\alpha &= \mathcal{C}\!\uparrow\![\![M]\!]\;(\lambda x.\,\lambda\gamma.\,\gamma\; x)\;\alpha \\
\mathcal{C}\!\uparrow\![\![\mu\widehat{\mathrm{tp}}.\langle x \| \widehat{\mathrm{tp}}\rangle]\!]\;\alpha &\twoheadrightarrow \alpha\; x \\
\mathcal{C}\!\uparrow\![\![\mu\widehat{\mathrm{tp}}.\langle\widehat{\mathrm{tp}} \uparrow M\rangle]\!]\;\alpha &= \mathcal{C}\!\uparrow\![\![M]\!]\;\alpha
\end{aligned}$$

Multiple Prompts: Second Attempt

Remembering the trail through the meta-continuation ⌣̈

$$D ::= \Box \mid \langle E[\mu\hat{\alpha}.D] \| q \rangle$$

$$\mu\hat{\alpha}.D[\langle \hat{\alpha} \uparrow \Delta.\, M \rangle] \mapsto M\{D[c]/\Delta c\}$$

$$(\hat{\alpha} \text{ not bound by } D)$$

$$\begin{aligned}
\widehat{\mathcal{C}}{\uparrow}[\![\mu\hat{\alpha}.c]\!] &= \lambda\beta.\, \lambda\gamma.\, \widehat{\mathcal{C}}{\uparrow}[\![c]\!]\ (\gamma\{\hat{\alpha} \mapsto \beta\}) \\
\widehat{\mathcal{C}}{\uparrow}[\![\langle \hat{\alpha} \uparrow \Delta.\, M \rangle]\!] &= \lambda\gamma.\, \mathbf{let}\, (\Delta, \beta, \gamma') = \gamma(\hat{\alpha}) \\
&\qquad\quad \mathbf{in}\, \widehat{\mathcal{C}}{\uparrow}[\![M]\!]\ \beta\ \gamma' \\
\widehat{\mathcal{C}}{\uparrow}[\![\Delta c]\!] &= \lambda\gamma.\, \Delta\ (\widehat{\mathcal{C}}{\uparrow}[\![c]\!])\ \gamma
\end{aligned}$$

Fine-grained Reduction Theory à la $\lambda\mu$

For local optimizations of multi-prompt control

$$
\begin{aligned}
(\lambda x.\, M)\; V &\to M\{V/x\} \\
(\mu\alpha.c)\; M &\to \mu\beta.c\{\langle N\; M \| \beta\rangle / \langle N \| \alpha\rangle\} \\
V\; (\mu\alpha.c) &\to \mu\beta.c\{\langle V\; N \| \beta\rangle / \langle N \| \alpha\rangle\} \\
\langle \mu\alpha.c \| q\rangle &\to c\{q/\alpha\}
\end{aligned}
$$

Fine-grained Reduction Theory à la $\lambda\mu$

For local optimizations of multi-prompt control

$$
\begin{aligned}
(\lambda x.\, M)\ V &\to M\{V/x\} \\
(\mu\alpha.c)\ M &\to \mu\beta.c\{\langle N\ M \| \beta\rangle/\langle N \| \alpha\rangle\} \\
V\ (\mu\alpha.c) &\to \mu\beta.c\{\langle V\ N \| \beta\rangle/\langle N \| \alpha\rangle\} \\
\langle \mu\alpha.c \| q\rangle &\to c\{q/\alpha\} \\
\mu\hat{\alpha}.\langle V \| \hat{\alpha}\rangle &\to V \\
\mu\hat{\beta}.\langle V \| \hat{\alpha}\rangle &\to \mu\_.\langle V \| \hat{\alpha}\rangle
\end{aligned}
$$

Fine-grained Reduction Theory à la $\lambda\mu$

For local optimizations of multi-prompt control

$$
\begin{aligned}
(\lambda x.\, M)\ V &\to M\{V/x\} \\
(\mu\alpha.c)\ M &\to \mu\beta.c\{\langle N\ M \| \beta\rangle / \langle N \| \alpha\rangle\} \\
V\ (\mu\alpha.c) &\to \mu\beta.c\{\langle V\ N \| \beta\rangle / \langle N \| \alpha\rangle\} \\
\langle \mu\alpha.c \| q\rangle &\to c\{q/\alpha\} \\
\mu\hat{\alpha}.\langle V \| \hat{\alpha}\rangle &\to V \\
\mu\hat{\beta}.\langle V \| \hat{\alpha}\rangle &\to \mu\_.\langle V \| \hat{\alpha}\rangle \\
\mu\hat{\alpha}.\langle \hat{\alpha} \uparrow \Delta.\, M\rangle &\to M\{c/\Delta c\} \\
\mu\hat{\beta}.\langle \hat{\alpha} \uparrow \Delta.\, M\rangle &\to \mu\beta.\langle \hat{\alpha} \uparrow \Delta'.\, M\{\Delta'\langle \mu\hat{\beta}.c \| \beta\rangle / \Delta c\}\rangle
\end{aligned}
$$

Which Style of Control Is Most "Primitive"?

Expressive "power" versus reasonability

$$(+\mathcal{F}+) \qquad \#E[\text{shift } V] \mapsto \#V\ (\lambda x.\ \#E[x])$$

$$(-\mathcal{F}+) \qquad \#E[\text{shift}_0\ V] \mapsto V\ (\lambda x.\ \#E[x])$$

$$(+\mathcal{F}-) \qquad \#E[\text{control } V] \mapsto \#V\ (\lambda x.\ E[x])$$

$$(-\mathcal{F}-) \qquad \#E[\text{control}_0\ V] \mapsto V\ (\lambda x.\ E[x])$$

Which Style of Control Is Most "Primitive"?

Expressive "power" versus reasonability

$$(+\mathcal{F}+) \qquad \#E[\mathsf{shift}\ V] \mapsto \#V\ (\lambda x.\ \#E[x])$$

$$(-\mathcal{F}+) \qquad \#E[\mathsf{shift}_0\ V] \mapsto V\ (\lambda x.\ \#E[x])$$

$$(+\mathcal{F}-) \qquad \#E[\mathsf{control}\ V] \mapsto \#V\ (\lambda x.\ E[x])$$

$$(-\mathcal{F}-) \qquad \#E[\mathsf{control}_0\ V] \mapsto V\ (\lambda x.\ E[x])$$

shift $(+\mathcal{F}+)$ and shift_0 $(-\mathcal{F}+)$ have a "nice" semantics

…but $\mathsf{control}_0$ $(-\mathcal{F}-)$ leaves the fewest prompts around

- Can always just put the ones you want back, right?
- What's the harm?

Have Your Cake and Eat It Too

shift$_0$ $\implies$ shift, **and with 2 prompts**, shift$_0$ $\implies$ control$_0$

Positive expressiveness vs negative expressiveness

shift$_0$ subsumes shift (Materzok and Biernacki)

- Actually, subsumes all levels of hierarchically-nested shift
- And while respecting all equations & semantics of shift
- Unlike control or control$_0$

shift$_0$ with (at least) 2 prompts subsumes control$_0$

- Just reserve 1 prompt for the "unwanted" one used to compose continuations "without" the prompt
- If you never seek it out, it's as if it's never there

$$\#_0^{\hat{\alpha}} M = \mu\widehat{\mathrm{tp}}.\langle \mu\hat{\alpha}.\langle M \| \widehat{\mathrm{tp}}\rangle \| \widehat{\mathrm{tp}}\rangle$$

$$\mathrm{control}_0^{\hat{\alpha}}\ f = \mu\beta.\langle \hat{\alpha} \uparrow \Delta.\ f\ (\lambda x.\ \mu\widehat{\mathrm{tp}}.\Delta\langle x \| \beta\rangle)\rangle$$