CPS Translation For The Domain-Free λ-Cube

Flip van Spaendonck

Institute for Computing and Information Sciences
Radboud University Nijmegen

26th June 2018

- Being able to transform a λ-theory into a CPS, is a welcome addition, as it provides more flexibility, and a step closer to turning it into a programming language.
- λC provides rich data structures which can be very useful.
- Thus being able to transform λC into a CPS is the next logical and desired step.

We need to define CPS translation functions for the domain-free *lambda*-cube.

Objects, Constructors, Kinds and Contexts

$C\langle \cdot \rangle$ is defined.

Top-level

$C[\![\cdot]\!]$ is defined.
Continuation only really occurs at Object level.

Objects

$$
\begin{aligned}
\mathcal{C}\langle x\rangle &= \lambda k.x\,k\\
\mathcal{C}\langle \lambda x.O\rangle &= \lambda k.k\,(\lambda x.\mathcal{C}\langle O\rangle)\\
\mathcal{C}\langle O\,O'\rangle &= \lambda k.\mathcal{C}\langle O\rangle\,(\lambda y.y\,\mathcal{C}\langle O'\rangle\,k)\\
\mathcal{C}\langle \lambda\alpha.O\rangle &= \lambda k.k\,(\lambda\alpha.\mathcal{C}\langle O\rangle)\\
\mathcal{C}\langle O\,C\rangle &= \lambda k.\mathcal{C}\langle O\rangle\,(\lambda y.y\,\mathcal{C}\langle C\rangle\,k)
\end{aligned}
$$

Constructors

$$
\begin{aligned}
\mathcal{C}\langle\alpha\rangle &= \alpha \\
\mathcal{C}\langle\lambda x.C'\rangle &= \lambda x.\mathcal{C}\langle C'\rangle \\
\mathcal{C}\langle C\ O\rangle &= \mathcal{C}\langle C\rangle\,\mathcal{C}\langle O\rangle \\
\mathcal{C}\langle\lambda\alpha.C\rangle &= \lambda\alpha.\mathcal{C}\langle C\rangle \\
\mathcal{C}\langle C\ C'\rangle &= \mathcal{C}\langle C\rangle\,\mathcal{C}\langle C'\rangle \\
\mathcal{C}\langle\Pi x{:}C.\ C'\rangle &= \Pi x{:}\neg\neg\mathcal{C}\langle C\rangle.\ \neg\neg\mathcal{C}\langle C'\rangle \\
\mathcal{C}\langle\Pi\alpha{:}K.\ C\rangle &= \Pi\alpha{:}\mathcal{C}\langle K\rangle.\ \neg\neg\mathcal{C}\langle C\rangle
\end{aligned}
$$

Kinds

$$
\begin{aligned}
\mathcal{C}\langle * \rangle &= * \\
\mathcal{C}\langle \Pi x{:}C.\, K \rangle &= \Pi x{:}\neg\neg\mathcal{C}\langle C \rangle.\, \mathcal{C}\langle K \rangle \\
\mathcal{C}\langle \Pi \alpha{:}K.\, K' \rangle &= \Pi \alpha{:}\mathcal{C}\langle K \rangle.\, \mathcal{C}\langle K' \rangle
\end{aligned}
$$

Contexts

$$
\begin{aligned}
\mathcal{C}\langle\cdot\rangle &= \cdot \\
\mathcal{C}\langle\Gamma, x{:}C\rangle &= \mathcal{C}\langle\Gamma\rangle, x{:}\neg\neg\mathcal{C}\langle C\rangle \\
\mathcal{C}\langle\Gamma, \alpha{:}K\rangle &= \mathcal{C}\langle\Gamma\rangle, \alpha{:}\mathcal{C}\langle K\rangle
\end{aligned}
$$

Top-level translation

$$
\begin{aligned}
\mathcal{C}⦃O⦄ &= \mathcal{C}\langle O\rangle \\
\mathcal{C}⦃C⦄ &= \neg\neg\mathcal{C}\langle C\rangle \\
\mathcal{C}⦃K⦄ &= \mathcal{C}\langle K\rangle \\
\mathcal{C}⦃\Box⦄ &= \Box \\
\mathcal{C}⦃\Gamma⦄ &= \bot : *, \mathcal{C}\langle\Gamma\rangle
\end{aligned}
$$

Proof of Correctness:

$\Gamma \Vdash A : B \Rightarrow C\llbracket\Gamma\rrbracket \Vdash C\langle A\rangle : C\llbracket B\rrbracket$

Proof Sketch:

1. Show that for all $O \in Obj[DFCUBE]$, $C\langle O\rangle \equiv \lambda k.O'$ for some O' and therefore $\lambda k.C\langle O\rangle k \to_\beta C\langle O\rangle$
2. Prove by induction on the structure $A, B \in Terms[DFCUBE]$ that
   a) $C\langle A\rangle\{x := C\langle O\rangle\} \twoheadrightarrow_\beta C\langle A\{x := O\}\rangle$
   b) $C\langle A\rangle\{x := C\langle B\rangle\} \equiv C\langle A\{\alpha := B\}\rangle$
   c) $(A \twoheadrightarrow_\beta B) \Rightarrow (C\langle A\rangle \twoheadrightarrow_\beta C\langle B\rangle)$
3. Prove by induction on the structure of derivations that: $\Gamma \Vdash A : B \Rightarrow C\llbracket\Gamma\rrbracket \Vdash C\langle A\rangle : C\llbracket B\rrbracket$

Transforming to CPS creates a lot of abstractions, we want to reduce this to make it easier and faster to work with.
We will do this by reducing the amount of administrative abstractions.
These only form through translation on objects, thus the rest is left untouched.

$$\mathcal{C}^+\langle O\rangle = \lambda k.(O : k)$$

$$\begin{aligned}
x : \mathcal{K} &= x\,\mathcal{K} \\
(\lambda x.O) : k &= k\,(\lambda x.\mathcal{C}^+\langle O\rangle) \\
(\lambda \alpha.O) : k &= k\,(\lambda \alpha.\mathcal{C}^+\langle O\rangle) \\
(\lambda x.O) : (\lambda y.y\,O'\,\mathcal{K}) &= (\lambda x.\mathcal{C}^+\langle O\rangle)\,O'\,\mathcal{K} \\
(\lambda \alpha.O) : (\lambda y.y\,C\,\mathcal{K}) &= (\lambda \alpha.\mathcal{C}^+\langle O\rangle)\,C\,\mathcal{K} \\
O\,O' : \mathcal{K} &= O : (\lambda y.y\,\mathcal{C}^+\langle O'\rangle\,\mathcal{K}) \\
O\,C : \mathcal{K} &= O : (\lambda y.y\,\mathcal{C}^+\langle C\rangle\,\mathcal{K})
\end{aligned}$$

Figure 8. Optimizing CPS translation for the domain-free λ-cube (excerpts).

Questions?