Value Conversion in IL1 after Lambda Hoisting

Lee Gao (lg342)

April 15, 2013

Given the closure-conversion and then hoisted restricted language IL1 detailed below

$$\begin{aligned}
v &::= n \mid x \mid \mathsf{halt} \\
e &::= v \mid v_0 + v_1 \mid (v_0, \cdots, v_n) \mid \pi_n v \\
c &::= \mathsf{let}\, x = e \,\mathsf{in}\, c \mid v_0\ v_1\ v_2 \mid v_0\ v_1
\end{aligned}$$

we want to "lift" numbers and halt into expressions (as Val used in bindings only) and leave only variables as values, so in effect our language will now look like

$$\begin{aligned}
v &::= x \\
e &::= v \mid \mathsf{val}(n) \mid \mathsf{val}(\mathsf{halt}) v_0 + v_1 \mid (v_0, \cdots, v_n) \mid \pi_n v \\
c &::= \mathsf{let}\, x = e \,\mathsf{in}\, c \mid v_0\ v_1\ v_2 \mid v_0\ v_1
\end{aligned}$$

We want to define a set of "lowering" translation $\mathcal{LV}[\![v]\!], \mathcal{LE}[\![e]\!], \mathcal{LC}[\![c]\!]$ that binds all non-variable values (integers and halts) into their own variables. Therefore, we need to have both the value and the expressions translation be able to be abstracted as bindings.

$$\begin{aligned}
\mathcal{LV}[\![v]\!] &: (\mathsf{var} \times e)\mathsf{list} \times \mathsf{var} \\
\mathcal{LE}[\![e]\!] &: (\mathsf{var} \times e)\mathsf{list} \times \mathsf{e} \\
\mathcal{LC}[\![c]\!] &: c
\end{aligned}$$