Assignment 6
The **K**-Machine

15-814: Types and Programming Languages

Jan Hoffmann & C.B. Aberlé

Due Tuesday, October 31, 2023
75 points

You should hand in one file:

- `hw06.pdf` with your written solutions to the questions.

1 Products in the K-Machine

Recall the formulation of positive products from Assignment four:

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (e_1, e_2) : \tau_1 \otimes \tau_2} \qquad \frac{\Gamma \vdash e_0 : \tau_1 \otimes \tau_2 \qquad \Gamma, x : \tau_1, y : \tau_2 \vdash e : \tau}{\Gamma \vdash \texttt{split}(e_0, x.y.e) : \tau}$$

Dynamics:

$$V_{prod^+}\ \frac{e_1\ \mathsf{val} \qquad e_2\ \mathsf{val}}{(e_1, e_2)\ \mathsf{val}} \qquad E_{prod1^+}\ \frac{e_1 \mapsto e_1'}{(e_1, e_2) \mapsto (e_1', e_2)} \qquad E_{prod2^+}\ \frac{e_1\ \mathsf{val} \qquad e_2 \mapsto e_2'}{(e_1, e_2) \mapsto (e_1, e_2')}$$

$$E_{split1}\ \frac{e_0 \mapsto e_0'}{\texttt{split}(e_0, x.y.e) \mapsto \texttt{split}(e_0', x.y.e)} \qquad E_{split2}\ \frac{(e_1, e_2)\ \mathsf{val}}{\texttt{split}((e_1, e_2), x.y.e) \mapsto [e_1/x, e_2/y]e}$$

We will extend the **K**-Machine for call-by-value **PCF** with positive products. First, we define the following new frames:

$$\frac{}{(-, e_2)\ \mathsf{frame}} \qquad \frac{e_1\ \mathsf{val}}{(e_1, -)\ \mathsf{frame}} \qquad \frac{}{\texttt{split}(-, x.y.e)\ \mathsf{frame}}$$

The statics of the frames are as follows:

$$\frac{e_2 : \tau_2}{(-, e_2) : \tau_1 \rightsquigarrow \tau_1 \otimes \tau_2} \qquad \frac{e_1 : \tau_1}{(e_1, -) : \tau_2 \rightsquigarrow \tau_1 \otimes \tau_2} \qquad \frac{x : \tau_1, y : \tau_2 \vdash e : \tau}{\texttt{split}(-, x.y.e) : \tau_1 \otimes \tau_2 \rightsquigarrow \tau}$$

Task 1 (10 points) *Extend the **K**-Machine for call-by-value **PCF** from Lecture 12 with positive products. Progress and preservation should continue to hold, but you do not need to prove this.*

Hint. You should write down five transition rules.

2 Soundness for the K-Machine

In lecture we showed that the **K**-Machine for call-by-value **PCF** is *complete* with respect to the small-step operational semantics in the sense that if $e \mapsto^* v$ and v val then $k \triangleright e \mapsto^* k \triangleleft v$ for all stacks k. The proof relied on the fact that evaluation defined in terms of the small-step semantics is equivalent to the big-step semantics $e \Downarrow v$.

In this assignment we will show that the converse property also holds for the **K**-Machine:

*Soundness of the **K**-Machine w.r.t. big-step semantics*. If $\epsilon \triangleright e \mapsto^* \epsilon \triangleleft v$, then $e \Downarrow v$.

The soundness with respect to small-step semantics then follows directly via the aforementioned equivalence. The complete definition for big-step operational semantics of call-by-value **PCF** (the relation $e \Downarrow v$) can be found in Lecture 13. The definition of the **K**-Machine for call-by-value **PCF** can be found in Lecture 12.

Properties of big-step operational semantics. We need the following lemma:

Lemma 1 *If v val then $v \Downarrow v$.*

and the following notation:

Notation: if expressions e, d are such that, if there exists some v val for which then $e \Downarrow v$, then $d \Downarrow v$ (in other words, if either evaluation of e does not terminate, or e and d both evaluate to a value which is moreover the *same* value), then we write $e \ll d$.

Stacks as functions on expressions. A stack k can be thought of as a concrete representation of a unary function Expr $\to$ Expr, where Expr is the set of PCF expressions. To make this idea precise, we define an operation $- \bullet -$ (called the fusing) which "applies" a stack k to an expression e, yielding an expression $k \bullet e$. We define $- \bullet -$ by induction on the length of the stack k:

$$\begin{aligned}
\epsilon \bullet e &= e \\
k; \mathtt{suc}(-) \bullet e &= k \bullet \mathtt{suc}(e) \\
k; \mathtt{ap}(-, e_2) \bullet e &= k \bullet \mathtt{ap}(e, e_2) \\
k; \mathtt{ap}(e_1, -) \bullet e &= k \bullet \mathtt{ap}(e_1, e) \\
k; \mathtt{ifz}(-, e_0, x.e_1) \bullet e &= k \bullet \mathtt{ifz}(e, e_0, x.e_1)
\end{aligned}$$

To prove the result we will need the following lemma, which shows that the application of stacks is compatible with evaluation:

Lemma 2 *For all stacks k, if $e \ll d$, then $k \bullet e \ll k \bullet d$.*

Proof: By induction on the length of the stack k. □

Task 2 (20 points) *Complete the proof of Lemma 2. You only need to show the case for $k = k'; \mathtt{ap}(-, e_2)$.*

We write $k \bowtie e$ to denote either $k \triangleright e$ or $k \triangleleft e$. Similar to the proof of the equivalence of big-step and small-step operational semantics, we need head expansion for evaluation:

Lemma 3 *Suppose that $s \mapsto k' \bowtie e'$. Then we have the following:*

1. *If $s = k \triangleright e$, then $k' \bullet e' \ll k \bullet e$.*
2. *If $s = k \triangleleft e$ and e val, then $k' \bullet e' \ll k \bullet e$.*

Proof: By induction on the derivation of $s \mapsto k' \bowtie e'$.

Case:

$$\frac{}{k; \mathtt{ifz}(-, e_0, x.e_1) \triangleleft \mathtt{suc}(e) \mapsto k \triangleright [e/x]e_1}$$

□

Task 3 (20 points) *Complete the proof of Lemma 3 for the case indicated.*

Observe that (internal) states of the form $k \triangleleft e$ satisfy the following property:

Lemma 4 *Suppose that $s \mapsto^* k' \triangleleft e'$. Then the following holds:*

1. *If $s = k \triangleright e$ then e' val.*
2. *If $s = k \triangleleft e$ and e val then e' val.*

Using Lemmata 1, 3, and 4, we may prove the following theorem:

Theorem 5 *Suppose that $s \mapsto^* \epsilon \triangleleft v$. Then the following holds:*

1. *If $s = k \triangleright e$, then $k \bullet e \Downarrow v$.*
2. *If $s = k \triangleleft e$ and e val, then $k \bullet e \Downarrow v$.*

Proof: By induction on the length of the transition sequence. □

Task 4 (25 points) *Complete the proof of Theorem 5.*

Finally we obtain the soundness of the **K**-Machine as a corollary to Theorem 5:

Corollary 6 *If $\epsilon \triangleright e \mapsto^* \epsilon \triangleleft v$ then $e \Downarrow v$.*