Elements of Programming Languages
Tutorial 7: Small-step semantics and type soundness
Week 9 (November 14–18, 2016)

Exercises marked * are more advanced. Please try all unstarred exercises before the tutorial meeting.

1. **Comparing large-step and small-step derivations**
Write both large-step and small-step derivations for the following expressions. For the small-step derivations, construct the derivations of each $e \mapsto e'$ step explicitly.

(a) $(\lambda x. x + 1) \ 42$
(b) $(\lambda x. \text{if } x == 1 \text{ then } 2 \text{ else } x + 1) \ 42$

2. **Small-step derivations that go wrong**
For each of the following expressions, show the small-step evaluation leading to the point where evaluation becomes stuck due to a dynamic type error. (There is no need to show the derivations of each step.)

(a) $((\lambda x. \lambda y. \text{let } z = x + y \text{ in } z + 1) \ 42) \ \text{true}$
(b) $(\lambda x. \text{if } x \text{ then } x + 1 \text{ else } x + 2) \ \text{true}$

3. **Small-step rules for $L_{\text{Data}}$**
Recall that we defined the semantics for $L_{\text{Data}}$ using big-step rules, as follows:

\[
e \Downarrow v
\]

\[
\begin{align*}
& e_1 \Downarrow v_1 \quad e_2 \Downarrow v_2 \quad e \Downarrow (v_1, v_2) \quad e \Downarrow (v_1, v_2) \\
& (e_1, e_2) \Downarrow (v_1, v_2) \quad \text{fst } e \Downarrow v_1 \quad \text{snd } e \Downarrow v_2
\end{align*}
\]

(a) For each construct, write out equivalent small-step rules. Are there any design choices in translating the big-step rules to small-step rules?
(b) (⋆) Construct small-step derivations reducing the following expressions to values:

i. $(\lambda p. (\text{snd } p, \text{fst } p + 2)) \ (17, 42)$
ii. $(\lambda x. \text{case } x \text{ of } \{ \text{left}(y). y + 1 ; \text{right}(z). z \}) \ (\text{left}(42))$
4. (⋆) **Type soundness for nondeterminism**

This question builds on the nondeterministic choice construct mentioned in an earlier tutorial, with the following typing rules:

\[
\Gamma \vdash e : \tau
\]

\[
\begin{array}{c}
\Gamma \vdash e_1 : \tau \quad \Gamma \vdash e_2 : \tau \\
\hline
\Gamma \vdash e_1 \Box e_2 : \tau
\end{array}
\]

and small-step evaluation rules:

\[
e \rightarrow e'
\]

\[
\begin{array}{c}
e_1 \Box e_2 \rightarrow e_1 \\
e_1 \Box e_2 \rightarrow e_2
\end{array}
\]

(a) State the *preservation* property. Outline how we could prove the cases of preservation for nondeterministic expressions.

(b) State the *progress* property. Outline how we could prove the cases of progress for nondeterministic expressions.