Sample Mid-Term Exam 2

CS 5510, Fall 2016

November 3

Instructions: You have eighty minutes to complete this open-book, open-note, closed-interpreter exam. Please write all answers in the provided space, plus the back of the exam if necessary.

Note on actual exam: The exam will refer to the lambda-k.rkt interpreter. If you need the interpreter for reference to answer the questions, please bring a copy (paper or electronic) with you.

1) [15 pts] Which of the following produce different results in a eager language and a lazy language? Both produce the same result if they both produce the same number or they both produce a procedure (even if the procedure doesn’t behave exactly the same when applied), but they can differ in errors reported.

   a) {{lambda {y} 12} {1 2}}
   b) {lambda {x} {{lambda {y} 12} {1 2}}}
   c) {+ 1 {lambda {y} 12}}
   d) {+ 1 {{lambda {x} {+ 1 13}} {+ 1 {lambda {z} 12}}}}
   e) {+ 1 {{lambda {x} {+ x 13}} {+ 1 {lambda {z} 12}}}}
2) [25 pts] Given the type rules

\[
\begin{align*}
\text{\ldots x} & \leftarrow \tau \ldots, \Gamma \vdash x : \tau \quad \Gamma \vdash 1 : \text{num} \\
\Gamma \vdash e_1 : \text{num} \quad \Gamma \vdash e_2 : \text{num} \\
\Gamma \vdash \{e_1 \ e_2\} : \text{num}
\end{align*}
\]

in one of the following expressions, the \____ can be filled in with a type so that the resulting expression has a type in the empty environment, while there is no type for the \____ that causes the other to have a type. Pick the right expression and show a derivation tree (which is a trace of typecheck that's written in the style as the type rules above) demonstrating that the chosen expression has a type.

\[
\begin{align*}
\{\lambda \{x : \ldots\}\ \{\ + \ x \ 1\}\ x\}
\end{align*}
\]

\[
\begin{align*}
\{\lambda \{x : \ldots\}\ \{\ + \ \{x \ 1\} \ 1\}\}
\end{align*}
\]

Note that your answer should not include symbols like $\Gamma$, $\tau$, or $e$, except when used as designated abbreviations, since those are meta-variables that are replaced by concrete environments, types, and expressions in the derivation tree.
3) [60 pts] Given the following expression:

\[
\begin{align*}
&\{\text{lambda } \{x\} \{x \ x\}\} \\
&\{\text{lambda } \{y\} \ 12\}\}
\end{align*}
\]

Describe a trace of the evaluation in terms of arguments to interp and continue functions for every call of each in the lambda-k.rkt interpreter. (There will be 7 calls to interp and 5 calls to continue.) The interp function takes three arguments — an expression, an environment, and a continuation — so show all three for each interp call. The continue function takes two arguments — a continuation and a value — so show both for each continue call. Represent continuations using records.
Answers

1) \( a \) and \( d \).

2) 
\[
\frac{\Gamma_1 \vdash x : (\text{num} \rightarrow \text{num}) \quad \Gamma_1 \vdash 1 : \text{num}}{
\Gamma_1 \vdash \{x\} : \text{num} \quad \Gamma_1 \vdash 1 : \text{num}}
\]

\[
\emptyset \vdash \{\lambda x : (\text{num} \rightarrow \text{num})\} \{+ \{x\} 1\} : ((\text{num} \rightarrow \text{num}) \rightarrow \text{num})
\]

3) 
\[
\text{interp expr} = \quad \{\lambda x : (\text{num} \rightarrow \text{num})\} \{\lambda y : 12\}
\]
env = mt-env
k = (doneK)

\[
\text{interp expr} = \quad \{\lambda x : (\text{num} \rightarrow \text{num})\}
\]
env = mt-env
k = (appArgK \{\lambda y : 12\} mt-env (doneK)) = \( k_1 \)
cont k = (appArgK \{\lambda y : 12\} mt-env (doneK)) or \( k_1 \)
val = (closV 'x \{x\} mt-env) = v_1

\[
\text{interp expr} = \quad \{\lambda y : 12\}
\]
env = mt-env
k = (doAppK \{\lambda y : 12\} (doneK)) = \( k_2 \)
cont k = (doAppK \{\lambda y : 12\} (doneK)) or \( k_2 \)
val = (closV 'y \{12\} mt-env) = v_2

\[
\text{interp expr} = \quad \{x\}
\]
env = (extend-env (bind 'x \{x\}) mt-env) = \( e_1 \)
k = (doneK)

\[
\text{interp expr} = \quad x
\]
env = \( e_1 \)
k = (appArgK x \( e_1 \) (doneK)) = \( k_3 \)
cont k = (appArgK x \( e_1 \) (doneK)) or \( k_3 \)
val = v_2

\[
\text{interp expr} = \quad x
\]
env = \( e_1 \)
k = (doAppK \{x\} \( v_2 \) (doneK)) = \( k_4 \)
cont k = (doAppK \{x\} \( v_2 \) (doneK)) or \( k_4 \)
val = v_2

\[
\text{interp expr} = \quad 12
\]
\[
\text{env} = (\text{extend-env} (\text{bind } 'y \, v_2) \, \text{mt-env}) \\
\text{k} = (\text{doneK})
\]

\[
\text{cont } \text{k} = (\text{doneK}) \\
\text{val} = (\text{numV } 12)
\]