CSE 526: Principles of Programming Languages

March 25, 2010 — **Mid-Term Exam** — Max: 100 points

Spring 2010 — Duration: 1h 20m

---

1. For this question, consider untyped lambda calculus with call by value (CBV) reduction strategy. The syntax and evaluation rules are recalled below.

Terms and Values:

$$\begin{array}{rcll} t & ::= & x \mid (t\ t) \mid \lambda x.\ t & \boxed{\text{Terms}} \\ v & ::= & \lambda x.\ t & \boxed{\text{Values}} \end{array}$$

Evaluation Rules:

$$\frac{t_1 \to t_1'}{t_1\ t_2 \to t_1'\ t_2} \quad \text{E-App1}$$

$$\frac{t_2 \to t_2'}{v_1\ t_2 \to v_1\ t_2'} \quad \text{E-Abs2}$$

$$(\lambda x.\ t_1)\ v_2 \to [x \mapsto v_2]t_1 \quad \text{E-AppAbs}$$

   (a) [15 points] Show that CBV evaluation preserves "closedness" of lambda terms. That is, if t is a closed lambda term and $t \to t'$ then t' is a closed lambda term.

   (b) [7 points] Show that CBV evaluation does not necessarily preserve "openness" of lambda terms. That is, give two lambda terms t and t' such that t is not closed, t' is closed, and $t \to t'$.

   (c) [15 points] Show the progress property for closed lambda terms under CBV. That is, if t is a closed lambda term then either t is a value, or there is a t' such that $t \to t'$.

2. [3 points each] Determine the type of each of the following terms in typed lambda calculus with simple extensions. For each term, write its type, or state that it is not well-typed.

   (a) $\lambda x : A.\ \lambda y : B.\ x$

   (b) $\lambda x : A.\ \lambda y : A \to B.\ \lambda z : B \to C.\ z\ (y\ x)$

   (c) $\lambda x : A.\ \texttt{let}\ y = \lambda z : A.\ z\ \texttt{in}\ (y\ x)$

   (d) $\lambda x : \{p : A, q : A \to B\}.\ \lambda y : B \to A.\ y\ (x.q\ x.p)$

   (e) $\texttt{fix}\ \lambda x : A \to A.\ \lambda y : A.\ y$

[**Question 3 is on the next page**]

3. Consider the language **BN** whose syntax and single-step operational semantics is given below.

Terms and Values:

$$
\begin{array}{rcll}
t & ::= & & \boxed{\textbf{\textit{Terms:}}} \\
 & & 0 & \boxed{\textit{constant zero}} \\
 & | & 1 & \boxed{\textit{constant one}} \\
 & | & \texttt{seq}(t,t) & \boxed{\textit{sequence}} \\
 & | & \texttt{nil} & \boxed{\textit{empty sequence}} \\
 & | & \texttt{next}\ t & \boxed{\textit{successor}} \\
 & | & \texttt{iszero}\ t & \boxed{\textit{zero test}}
\end{array}
\qquad
\begin{array}{rcll}
v & ::= & & \boxed{\textbf{\textit{Values:}}} \\
 & & b & \boxed{\textit{bit}} \\
 & | & nv & \boxed{\textit{numeric value}} \\
 & & & \\
b & ::= & 0 & \\
 & | & 1 & \\
 & & & \\
nv & ::= & \texttt{nil} & \\
 & | & \texttt{seq}(b, nv) &
\end{array}
$$

Evaluation Rules:

$$\frac{t_1 \rightarrow t_1'}{\texttt{seq}(t_1,t_2) \rightarrow \texttt{seq}(t_1',t_2)}\ \text{E-S}{\small\text{EQ}}1 \qquad \frac{t_2 \rightarrow t_2'}{\texttt{seq}(t_1,t_2) \rightarrow \texttt{seq}(t_1,t_2')}\ \text{E-S}{\small\text{EQ}}2$$

$$\frac{t_1 \rightarrow t_1'}{\texttt{next}\ t_1 \rightarrow \texttt{next}\ t_1'}\ \text{E-N}{\small\text{EXT}} \qquad \frac{t_1 \rightarrow t_1'}{\texttt{iszero}\ t_1 \rightarrow \texttt{iszero}\ t_1'}\ \text{E-I}{\small\text{S}}\text{Z}{\small\text{ERO}}$$

$$\frac{\texttt{next}\ t_1 \rightarrow t_1'}{\texttt{next}\ \texttt{seq}(1,t_1) \rightarrow \texttt{seq}(0,t_1')}\ \text{E-N}{\small\text{EXT}}\text{S}{\small\text{EQ}}\text{O}{\small\text{NE}} \qquad \frac{\texttt{iszero}\ t_1 \rightarrow t_1'}{\texttt{iszero}\ \texttt{seq}(0,t_1) \rightarrow t_1'}\ \text{E-I}{\small\text{S}}\text{Z}{\small\text{ERO}}\text{S}{\small\text{EQ}}\text{Z}{\small\text{ERO}}$$

$$\texttt{next}\ \texttt{seq}(0,t_1) \rightarrow \texttt{seq}(1,t_1)\ \ \text{E-N}{\small\text{EXT}}\text{S}{\small\text{EQ}}\text{Z}{\small\text{ERO}} \qquad \texttt{iszero}\ \texttt{seq}(1,t_1) \rightarrow 0\ \ \text{E-I}{\small\text{S}}\text{Z}{\small\text{ERO}}\text{S}{\small\text{EQ}}\text{O}{\small\text{NE}}$$

$$\texttt{next}\ \texttt{nil} \rightarrow \texttt{seq}(1,\texttt{nil})\ \ \text{E-N}{\small\text{EXT}}\text{N}{\small\text{IL}} \qquad \texttt{iszero}\ \texttt{nil} \rightarrow 1\ \ \text{E-I}{\small\text{S}}\text{Z}{\small\text{ERO}}\text{N}{\small\text{IL}}$$

(a) [6 points] Define an OCAML type `bnterm` to represent terms in **BN** as OCAML data structures. For full credit every instance of type `bnterm` must represent a term in **BN**, and every term in **BN** must be represented by an instance of type `bnterm`.

(b) [6 points] Give the derivation for the evaluation statement

$$\texttt{next}\ \texttt{seq}(1,\texttt{seq}(1,\texttt{nil})) \rightarrow \texttt{seq}(0,\texttt{seq}(0,\texttt{seq}(1,\texttt{nil})))$$

(c) [6 points] Recall the theorem on the determinacy of one-step evaluation: If $t \rightarrow t'$ and $t \rightarrow t''$ then $t' = t''$. Show that this theorem *does not hold* for the single step semantics given above.

(d) [6 points] Give a sequence of single step evaluations that take the term

$$\texttt{iszero}\ \texttt{next}\ \texttt{seq}(\ \texttt{iszero}\ \texttt{seq}(0,\texttt{nil}),\quad \texttt{nil}\ )$$

to its normal form 0.

(e) [8 points] Show that not all normal forms are values in this language. That is, give an example of a term t in **BN** that is in normal form such that t is not a value. For full credit you should justify clearly that t is a term in normal form, and that it is not a value.

(f) [10 points] Using two types `Bit` and `Num`, define a typing relation for **BN** such that for every well-typed term t, the normal form of t is a value.

(g) [6 points] Give a term t in **BN** that is *not well typed* according to your typing relation defined in part (3f) above but is such that the normal form of t is a value.

END OF EXAM