CIS 705 — Programming Languages — Spring 2009

Final Examination

Model Answers

Question 1

```
(* val long : 'a list -> ('a * int)option * int

   long xs returns (opt, n), where

     if xs = [], then opt = None;

     if xs <> [], then opt = Some(List.hd xs, m), where
     m is the length of the longest strictly ascending prefix
     of xs;

     n is largest such that there is a strictly ascending sublist of
     xs with length n *)

let rec long xs =
      match xs with
            []      -> (None, 0)
          | x :: xs ->
              match long xs with
                    (None,       n) -> (Some(x, 1), 1)  (* n = 0 *)
                  | (Some(y, m), n) ->  (* n >= 1 *)
                      if x < y
                      then (Some(x, m + 1),
                            if n > m + 1 then n else m + 1)
                      else (Some(x, 1), n)

let longest xs = snd(long xs)
```

Question 2

Determinacy of Evaluation We disprove the statement. Let x be a variable and $t = (\lambda x.\, x)\mathsf{unit}$. By (E-AppAbs), we have that $t \rightarrow [x \mapsto \mathsf{unit}]x = \mathsf{unit}$. Thus, by (E-App2), we have that $(\lambda x.\, \mathsf{unit})t \rightarrow (\lambda x.\, \mathsf{unit})\mathsf{unit}$. But, by (E-AppAbs), we have that $(\lambda x.\, \mathsf{unit})t \rightarrow [x \mapsto t]\mathsf{unit} = \mathsf{unit}$, and $(\lambda x.\, \mathsf{unit})\mathsf{unit} \neq \mathsf{unit}$.

Uniqueness of Typing We disprove the statement. Suppose x is a variable.

First, we prove that, for all types T, $\emptyset \vdash \lambda x.\, x : T \rightarrow T$. Suppose T is a type. Because $(x, T) \in \{(x, T)\} = \emptyset[x \mapsto T]$, (T-Var) tells us that $\emptyset[x \mapsto T] \vdash x : T$. Thus $\emptyset \vdash \lambda x.\, x : T \rightarrow T$, by (T-Abs).

From this result, we have that $\emptyset \vdash \lambda x.\, x : \mathsf{Unit} \to \mathsf{Unit}$ and $\emptyset \vdash \lambda x.\, x : (\mathsf{Unit} \to \mathsf{Unit}) \to (\mathsf{Unit} \to \mathsf{Unit})$. But $\mathsf{Unit} \to \mathsf{Unit} \neq (\mathsf{Unit} \to \mathsf{Unit}) \to (\mathsf{Unit} \to \mathsf{Unit})$.

Progress We prove the statement. Let $P = \{\, (\Gamma, t, T) \mid \Gamma$ is a context, t is a term, T is a type and, if $\Gamma = \emptyset$, then t is a closed term that is not stuck$\,\}$. We use induction on the typing relation to prove that, for all contexts Γ, terms t and types T, if $\Gamma \vdash t : T$, then $P(\Gamma, t, T)$. (Then, suppose t is a closed term, and t is well-typed. Thus $\emptyset \vdash t : T$ for some type T. By the result of the induction, we have that $P(\emptyset, t, T)$. Thus, if $\emptyset = \emptyset$, then t is a closed term that is not stuck. But $\emptyset = \emptyset$, and thus t is not stuck.)

(T-Unit) Suppose Γ is a context. We must show that $P(\Gamma, \mathsf{unit}, \mathsf{Unit})$, and this follows because unit is a closed value, and so is a closed term that is not stuck.

(T-Var) Suppose Γ is a context, x is a variable, T is a type and $(x, T) \in \Gamma$. We must show that $P(\Gamma, x, T)$. Suppose $\Gamma = \emptyset$. We must show that x is a closed term that is not stuck. But $(x, T) \in \Gamma = \emptyset$—contradiction. Thus x is a closed term that is not stuck.

(T-Abs) Suppose Γ is a context, x is a variable, t is a term, T_1 and T_2 are types, and $\Gamma[x \mapsto T_1] \vdash t : T_2$, and assume the inductive hypothesis, $P(\Gamma[x \mapsto T_1], t, T_2)$. We must show that $P(\Gamma, \lambda x.\, t, T_1 \to T_2)$. Suppose $\Gamma = \emptyset$. We must show that $\lambda x.\, t$ is a closed term that is not stuck. Because $\emptyset[x \mapsto T_1] \vdash t : T_2$, we have that $\mathsf{FV}(t) \subseteq \mathsf{dom}(\emptyset[x \mapsto T_1]) = \{x\}$. Thus $\mathsf{FV}(\lambda x.\, t) = \mathsf{FV}(t) \setminus \{x\} \subseteq \{x\} \setminus \{x\} = \emptyset$, showing that $\lambda x.\, t$ is closed. And $\lambda x.\, t$ is a value, showing that $\lambda x.\, t$ is not stuck. Thus $\lambda x.\, t$ is a closed term that is not stuck.

(T-App) Suppose Γ is a context, t_1 and t_2 are terms, T_1 and T_2 are types, $\Gamma \vdash t_1 : T_1 \to T_2$ and $\Gamma \vdash t_2 : T_1$, and assume the inductive hypothesis, $P(\Gamma, t_1, T_1 \to T_2)$ and $P(\Gamma, t_2, T_1)$. We must show that $P(\Gamma, t_1\, t_2, T_2)$. Suppose $\Gamma = \emptyset$. We must show that $t_1\, t_2$ is a closed term that is not stuck. By $P(\Gamma, t_1, T_1 \to T_2)$, we have that t_1 is a closed term that is not stuck. And, by $P(\Gamma, t_2, T_1)$, t_2 is a closed term. Thus $t_1\, t_2$ is closed. Because t_1 is a closed term that is not stuck, there are two cases to consider.

- Suppose t_1 is not a normal form. Thus there is a closed term t_1' such that $t_1 \to t_1'$. Hence $t_1\, t_2 \to t_1'\, t_2$, by (E-App1), showing that $t_1\, t_2$ is not a normal form, and so is not stuck. Thus $t_1\, t_2$ is a closed term that is not stuck.
- Suppose t_1 is a closed value. Because $\emptyset \vdash t_1 : T_1 \to T_2$, inversion of the typing lemma shows that $t_1 \neq \mathsf{unit}$. Thus $t_1 = \lambda x.\, u$ for a variable x and a term u. Hence $t_1\, t_2 = (\lambda x.\, u) t_2 \to [x \mapsto t_2]u$, by (E-AppAbs), showing that $t_1\, t_2$ is not a normal form, and so is not stuck. Thus $t_1\, t_2$ is a closed term that is not stuck.

Preservation We prove the statement. Let $P = \{\, (t, t') \mid t$ and t' are closed terms, and, for all types T, if $\emptyset \vdash t : T$, then $\emptyset \vdash t' : T\,\}$. We use induction on the evaluation relation to show that, for all closed terms t and t', if $t \to t'$, then $P(t, t')$. (Then, suppose t and t' are closed terms, T is a type, $\emptyset \vdash t : T$ and $t \to t'$. By the result of the induction, we have that $P(t, t')$. But $\emptyset \vdash t : T$, and thus $\emptyset \vdash t' : T$.)

(E-App1) Suppose t_1, t_1' and t_2 are closed terms, and $t_1 \to t_1'$, and assume the inductive hypothesis, $P(t_1, t_1')$. We must show that $P(t_1\, t_2, t_1'\, t_2)$. Suppose T is a type and $\emptyset \vdash t_1\, t_2 : T$. We must show that $\emptyset \vdash t_1'\, t_2 : T$. By inversion of the typing relation, we have that there is a type T' such that $\emptyset \vdash t_1 : T' \to T$ and $\emptyset \vdash t_2 : T'$. By the inductive hypothesis, it follows that $\emptyset \vdash t_1' : T' \to T$. Thus by (T-App), we have that $\emptyset \vdash t_1' \, t_2 : T$.

(E-App2) Suppose t_1, t_2 and t_2' are closed terms, and $t_2 \to t_2'$, and assume the inductive hypothesis, $P(t_2, t_2')$. We must show that $P(t_1 \, t_2, t_1 \, t_2')$. Suppose T is a type and $\emptyset \vdash t_1 \, t_2 : T$. We must show that $\emptyset \vdash t_1 \, t_2' : T$. By inversion of the typing relation, we have that there is a type T' such that $\emptyset \vdash t_1 : T' \to T$ and $\emptyset \vdash t_2 : T'$. By the inductive hypothesis, it follows that $\emptyset \vdash t_2' : T'$. Thus by (T-App), we have that $\emptyset \vdash t_1 \, t_2' : T$.

(E-AppAbs) Suppose x is a variable, t is a term such that $\mathsf{FV}(t) \subseteq \{x\}$, and t' is a closed term. We must show that $P((\lambda x.\, t)t', [x \mapsto t']t)$. Suppose T is a type and $\emptyset \vdash (\lambda x.\, t)t' : T$. We must show that $\emptyset \vdash [x \mapsto t']t : T$. By the inversion of the typing relation, we have that there is a type T' such that $\emptyset \vdash \lambda x.\, t : T' \to T$ and $\emptyset \vdash t' : T'$. Because $\emptyset \vdash \lambda x.\, t : T' \to T$, by another inversion of the typing relation, we have that $\emptyset[x \mapsto T'] \vdash t : T$. Since $\emptyset[x \mapsto T'] \vdash t : T$ and $\emptyset \vdash t' : T'$, the typing of substitutions lemma tells us that $\emptyset \vdash [x \mapsto t']t : T$.

Reverse Preservation We disprove the statement. Let x be a variable, $t = \lambda x.\, \mathsf{unit}$ and $u = \lambda x.\, \mathsf{unit}\, \mathsf{unit}$. Then $t\, u \to [x \mapsto u]\mathsf{unit} = \mathsf{unit}$ and $\emptyset \vdash \mathsf{unit} : \mathsf{Unit}$. Suppose, toward a contradiction, that $\emptyset \vdash t\, u : \mathsf{Unit}$. By inversion of the typing relation, we have that there is a type T such that $\emptyset \vdash t : T \to \mathsf{Unit}$ and $\emptyset \vdash u : T$. Since $\emptyset \vdash \lambda x.\, \mathsf{unit}\, \mathsf{unit} : T$, by inversion of the typing relation we have that there are types T_1 and T_2 such that $T = T_1 \to T_2$ and $\emptyset[x \mapsto T_1] \vdash \mathsf{unit}\, \mathsf{unit} : T_2$. Thus, by inversion of the typing relation, we have that there is a type T' such that $\emptyset[x \mapsto T_1] \vdash \mathsf{unit} : T' \to T_2$ and $\emptyset[x \mapsto T_1] \vdash \mathsf{unit} : T'$. Finally, since $\emptyset[x \mapsto T_1] \vdash \mathsf{unit} : T' \to T_2$, inversion of the typing relation tells us that $T' \to T_2 = \mathsf{Unit}$—contradiction. Thus it is not the case that $\emptyset \vdash t\, u : \mathsf{Unit}$.