UNIVERSITY OF EDINBURGH
COLLEGE OF SCIENCE AND ENGINEERING
SCHOOL OF INFORMATICS

TYPES AND SEMANTICS FOR PROGRAMMING LANGUAGES

Sunday 1 st April 2012

00:00 to 00:00

INSTRUCTIONS TO CANDIDATES

MOCK EXAM MOCK EXAM

Answer any TWO questions

All questions carry equal weight

MOCK EXAM MOCK EXAM

Year 4 Courses

Convener: ITO-Will-Determine
External Examiners: ITO-Will-Determine

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. This question uses the library definition of `list` and `In` in Coq. (The predicate `In` is also sometimes called `member`.)

   Here are informal definitions of the predicates `In` and `break`.

$$\texttt{in\_eq}\frac{}{\texttt{In}\ x\ (x :: xs)} \qquad \texttt{in\_cons}\frac{\texttt{In}\ y\ xs}{\texttt{In}\ y\ (x :: xs)}$$

$$\texttt{break\_eq}\frac{}{\texttt{break}\ (x :: xs)\ x\ xs} \qquad \texttt{break\_cons}\frac{\texttt{break}\ xs\ y\ ys}{\texttt{break}\ (x :: xs)\ y\ (x :: ys)}$$

   - Formalise the definition of `break`. (The definition of `In` is part of the library.)

     [*10 marks*]

   - Prove each of the following:

     (a) $\texttt{break}\ [1;2;3]\ 1\ [2;3]$.

     (b) $\texttt{break}\ [1;2;3]\ 2\ [1;3]$.

     (c) $\texttt{break}\ [1;2;3]\ 3\ [1;2]$.

     [*5 marks*]

   - Prove the following.

$$\texttt{Theorem break\_in} : \texttt{forall}\ (X : \texttt{Type})\ (x\ y : X)\ (xs\ ys : \texttt{list}\ X),$$
$$\texttt{break}\ xs\ y\ ys \to \texttt{In}\ x\ xs \to x = y \lor \texttt{In}\ x\ ys.$$

     [*10 marks*]

2. You will be provided with a definition of a simple imperative language in Coq.

Consider constructs satisfying the following rules.

Evaluation:

$$\texttt{E\_For}\frac{\begin{array}{c}\texttt{aeval}\ st\ a_1 = n \\ \texttt{t\_update}\ st\ x\ n = st' \\ \texttt{LOOP}\ x\ \texttt{TO}\ a_2\ \texttt{DO}\ c\ \texttt{END}/st' \Downarrow st''\end{array}}{\texttt{FOR}\ x\ \ ==\ \ a_1\ \texttt{TO}\ a_2\ \texttt{DO}\ c\ \texttt{END}/st \Downarrow st''}$$

$$\texttt{E\_LoopEnd}\frac{st\ x > \texttt{aeval}\ st\ a_2}{\texttt{LOOP}\ x\ \texttt{TO}\ a_2\ \texttt{DO}\ c\ \texttt{END}/st \Downarrow st}$$

$$\texttt{E\_LoopLoop}\frac{\begin{array}{c}st\ x \leq \texttt{aeval}\ st\ a_2 \\ c/st \Downarrow st' \\ \texttt{update}\ st'\ x\ (st'\ x + 1) = st'' \\ \texttt{LOOP}\ x\ \texttt{TO}\ a_2\ \texttt{DO}\ c\ \texttt{END}/st'' \Downarrow st'''\end{array}}{\texttt{LOOP}\ x\ \texttt{TO}\ a_2\ \texttt{DO}\ c\ \texttt{END}/st \Downarrow st'''}$$

Hoare logic:

$$\texttt{hoare\_for}\frac{\{\{P\}\}\ \texttt{LOOP}\ X\ \texttt{TO}\ a_2\ \texttt{DO}\ c\ \texttt{END}\ \{\{Q\}\}}{\{\{P[X \mapsto a_1]\}\}\ \texttt{FOR}\ X\ \ ==\ \ a_1\ \texttt{TO}\ a_2\ \texttt{DO}\ c\ \texttt{END}\ \{\{Q\}\}}$$

$$\texttt{hoare\_loop}\frac{\{\{P \wedge X \leq a_2\}\}\ c\ \{\{P[X \mapsto X+1]\}\}}{\{\{P\}\}\ \texttt{LOOP}\ X\ \texttt{TO}\ a_2\ \texttt{DO}\ c\ \texttt{END}\ \{\{P \wedge X > a_2\}\}}$$

- Extend the given definition to formalise the evaluation rules. [*12 marks*]
- Prove the Hoare rules. You will be provided with proofs of Hoare rules for the simple imperative language that you may modify. [*13 marks*]

3. You will be provided with a definition of simply-typed lambda calculus in Coq.

   Consider constructs satisfying the following rules.

   Evaluation:

$$\texttt{ST\_Snoc1}\ \frac{t_1 \Longrightarrow t_1'}{(\texttt{snoc}\ t_1\ t_2) \Longrightarrow (\texttt{snoc}\ t_1'\ t_2)}$$

$$\texttt{ST\_Snoc2}\ \frac{\texttt{value}\ v_1 \qquad t_2 \Longrightarrow t_2'}{(\texttt{snoc}\ v_1\ t_2) \Longrightarrow (\texttt{snoc}\ v_1\ t_2')}$$

$$\texttt{ST\_TCase}\ \frac{t_1 \Longrightarrow t_1'}{\begin{array}{c}(\texttt{tcase}\ t_1\ \texttt{of}\ \texttt{lin} \Rightarrow t_2 \mid \texttt{snoc}\ xs\ x \Rightarrow t_3) \\ \Longrightarrow (\texttt{tcase}\ t_1'\ \texttt{of}\ \texttt{lin} \Rightarrow t_2 \mid \texttt{snoc}\ xs\ x \Rightarrow t_3)\end{array}}$$

$$\texttt{ST\_TCaseLin}\ \frac{}{(\texttt{tcase}\ \texttt{lin}\ \texttt{of}\ \texttt{lin} \Rightarrow t_2 \mid \texttt{snoc}\ xs\ x \Rightarrow t_3) \Longrightarrow t_2}$$

$$\texttt{ST\_TCaseSnoc}\ \frac{\texttt{value}\ v_1 \qquad \texttt{value}\ v_2}{\begin{array}{c}(\texttt{tcase}\ (\texttt{snoc}\ v_1\ v_2)\ \texttt{of}\ \texttt{lin} \Rightarrow t_2 \mid \texttt{snoc}\ xs\ x \Rightarrow t_3) \\ \Longrightarrow [xs := v_1][x := v_2]t_3\end{array}}$$

   Typing

$$\texttt{T\_Lin}\ \frac{}{\Gamma \vdash \texttt{lin} \in \texttt{Tsil}\ T}$$

$$\texttt{T\_Snoc}\ \frac{\Gamma \vdash t_1 \in \texttt{Tsil}\ T \qquad \Gamma \vdash t_2 \in T}{\Gamma \vdash (\texttt{snoc}\ t_1\ t_2) \in \texttt{Tsil}\ T}$$

$$\texttt{T\_TCase}\ \frac{\begin{array}{c}\Gamma \vdash t_1 \in \texttt{Tsil}\ T \\ \Gamma \vdash t_2 \in T' \\ \Gamma, xs \in \texttt{Tsil}\ T, x \in T \vdash t_3 \in T'\end{array}}{\Gamma \vdash (\texttt{tcase}\ t_1\ \texttt{of}\ \texttt{lin} \Rightarrow t_2 \mid \texttt{snoc}\ xs\ x \Rightarrow t_3) \in T'}$$

   - Extend the given definition to formalise the evaluation and typing rules. [*12 marks*]
   - Prove progress. You will be provided with a proof of progress for simply-typed lambda calculus that you may extend. [*13 marks*]