INSTRUCTIONS TO CANDIDATES

Answer any TWO questions

All questions carry equal weight

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.

\[
\begin{align*}
\text{in}_{\text{eq}} & \quad \text{in} \quad x \quad (x :: xs) \\
\text{in}_{\text{cons}} & \quad \text{in} \quad y \quad (x :: xs) \\
\text{break}_{\text{eq}} & \quad \text{break} \quad (x :: xs) \quad x \quad xs \\
\text{break}_{\text{cons}} & \quad \text{break} \quad (x :: xs) \quad y \quad (x :: ys)
\end{align*}
\]

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

- Prove each of the following:
  
  (a) \text{break} \ [1; 2; 3] \ 1 \ [2; 3].
  
  (b) \text{break} \ [1; 2; 3] \ 2 \ [1; 3].
  
  (c) \text{break} \ [1; 2; 3] \ 3 \ [1; 2].

- Prove the following.

Theorem \text{break}_{\text{in}} : \forall (X : \text{Type}) \ (x \ y : X) \ (xs \ ys : \text{list} \ X),

\text{break} \ xs \ y \ ys \ \rightarrow \ \text{In} \ x \ xs \ \rightarrow \ x = y \ \vee \ \text{In} \ x \ ys.
2. You will be provided with a definition of a simple imperative language in Coq. Consider constructs satisfying the following rules.

Evaluation:

- \( \text{aeval } st \ a_1 = n \)
- \( \text{t_update } st \ x \ n = st' \)
- \( \frac{\text{E_For} \quad \text{LOOP } x \ TO \ a_2 \ DO \ c \ END / st' \Downarrow \ st''}{\text{FOR } x \ == \ a_1 \ TO \ a_2 \ DO \ c \ END / st \Downarrow \ st''} \)
- \( \frac{st \ x > \text{aeval } st \ a_2}{\text{E_LoopEnd} \quad \text{LOOP } x \ TO \ a_2 \ DO \ c \ END / st \Downarrow \ st} \)
- \( \frac{st \ x \leq \text{aeval } st \ a_2}{\text{E_LoopLoop} \quad \text{update } st' \ x \ (st' \ x + 1) = st'' \quad \text{LOOP } x \ TO \ a_2 \ DO \ c \ END / st'' / st'''} \)

Hoare logic:

- \( \frac{\{P\} \ \text{LOOP } X \ TO \ a_2 \ \text{DO } c \ \text{END } \{Q\}}{\{P[X ↦ a_1]\} \ \text{FOR } X \ == \ a_1 \ TO \ a_2 \ \text{DO } c \ \text{END } \{Q\}} \) \ (\text{hoare_for})
- \( \frac{\{P \land X \leq a_2\} \ \text{c } \{P[X ↦ X + 1]\}}{\{P\} \ \text{LOOP } X \ TO \ a_2 \ \text{DO } c \ \text{END } \{P \land X > a_2\}} \) \ (\text{hoare_loop})

- 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:

\[
\text{ST}\_\text{SnoC1} \quad t_1 \Rightarrow t'_1 \quad (\text{snoC} t_1 \ t_2) \Rightarrow (\text{snoC} t'_1 \ t_2)
\]

\[
\text{ST}\_\text{SnoC2} \quad \text{value} \ v_1 \ t_2 \Rightarrow t'_2 \quad (\text{snoC} v_1 \ t_2) \Rightarrow (\text{snoC} v_1 \ t'_2)
\]

\[
\text{ST}\_\text{TCase} \quad t_1 \Rightarrow t'_1 \quad (\text{tcase} t_1 \ \text{of lin} \Rightarrow t_2 \ | \ \text{snoC} \ xs \ x \Rightarrow t_3) \Rightarrow (\text{tcase} t'_1 \ \text{of lin} \Rightarrow t_2 \ | \ \text{snoC} \ xs \ x \Rightarrow t_3)
\]

\[
\text{ST}\_\text{TCaseLin} \quad (\text{tcase} \ \text{lin} \ \text{of lin} \Rightarrow t_2 \ | \ \text{snoC} \ xs \ x \Rightarrow t_3) \Rightarrow t_2
\]

\[
\text{ST}\_\text{TCaseSnoC} \quad \text{value} \ v_1 \ \text{value} \ v_2 \quad (\text{tcase} \ \text{snoC} \ v_1 \ v_2 \ \text{of lin} \Rightarrow t_2 \ | \ \text{snoC} \ xs \ x \Rightarrow t_3) \Rightarrow [xs := v_1][x := v_2]t_3
\]

Typing

\[
\text{T}\_\text{Lin} \quad \Gamma \vdash \text{lin} \in \text{Tsil} \ T
\]

\[
\text{T}\_\text{SnoC} \quad \Gamma \vdash t_1 \in \text{Tsil} \ T \quad \Gamma \vdash t_2 \in T \quad \Gamma \vdash (\text{snoC} \ t_1 \ t_2) \in \text{Tsil} \ T
\]

\[
\quad \Gamma \vdash t_1 \in \text{Tsil} \ T
\quad \Gamma \vdash t_2 \in T'
\quad \Gamma, xs \in \text{Tsil} \ T, \ x \in T \vdash t_3 \in T'
\quad \Gamma \vdash (\text{tcase} \ t_1 \ \text{of lin} \Rightarrow t_2 \ | \ \text{snoC} \ xs \ x \Rightarrow t_3) \in T'
\]

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