Final

Note

This is an open-book exam. You may consult any books, papers, or notes, but discussion with other students or seeking outside help is strictly forbidden.

Problems

1. (20 %) Prove, using *Natural Deduction* (in the sequent form), the validity of the following sequents.

   (a) $\neg B \rightarrow \neg A \vdash A \rightarrow B$

   (b) $\forall x A(x) \vdash \neg \exists x (\neg A(x))$

2. (20 %) Prove, using Coq, the validity of the following sequents. Write down the proof scripts that you used to complete the proofs.

   (a) $\vdash (A \rightarrow B \rightarrow C) \rightarrow (A \wedge B \rightarrow C)$

   (b) $\vdash (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$

3. (20 %) The following program segment finds the maximum element of an array A with n elements.

   S1: $i := 0$;
   S2: $max := A[i]$;
   S3: **while** $i < n - 1$ **do**
   S4: $i := i + 1$;
   S5: **if** $max < A[i]$ **then**
   S6: $max := A[i]$;
   od

   (a) Give a pair of pre and post-conditions to describe as precisely as possible what the program segment achieves. You should assume only a simple assertion language with constants $(0, 1)$, basic arithmetic operations $(+, -)$ and equality and inequality relations $(=, <, \cdots)$. So, that means you will have to define the relations that would be convenient for writing the needed assertions.

   (b) Annotate the program segment into a proof outline that clearly shows the total correctness of the program (according to the pre and post-conditions).

4. (20 %) Refinement is one of the most fundamental concepts in formal software development.

   (a) Explain the concept of refinement in words (without logical notations).

   (b) How is refinement formulated in Z or B (choose one of them)?

   Please try to be brief, but to the point.

5. (20 %) Prove the partial correctness of the following program using the Owicki-Gries method. Variables T, s_0, and s_1 are of the same type.

$$
\begin{array}{l}
\{true\} \\
acc := 0; \\
\left[\begin{array}{lcl}
T := 0; & & T := 1; \\
\textbf{await } T \neq 0; & & \textbf{await } T \neq 1; \\
s_0 := acc; & \| & s_1 := acc; \\
acc := s_0 + 1; & & acc := s_1 + 1; \\
T := 0; & & T := 1;
\end{array}\right] \\
\{acc = 2\}
\end{array}
$$