Final

Important Notes

This is an open-book exam. You may consult any book, paper, note, or on-line resource, but discussion with others (in person or via a network) is strictly forbidden.

Problems 2 and 5 require electronic submission. Please pack all files for the two problems in one single .zip file and email it to the instructor (`tsay@ntu.edu.tw`).

Problems

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

   (a) $\neg p \vee \neg q \vdash \neg(p \wedge q)$

   (b) $\forall x(\exists y(A \wedge B)) \vdash \forall x(\exists y A) \wedge \forall x(\exists y B)$

2. (20 %) Consider the following definition of `is_even` in Coq.

```
From Coq Require Import Arith.

Open Scope nat_scope.

Fixpoint is_even (n : nat) :=
  match n with
  | O => True
  | S O => False
  | S (S m) => is_even m
  end.

Lemma double_is_even (n : nat) : is_even (2 * n).
Proof.
  induction n.
  (* to be completed *)
Qed.
```

   Complete the proof of the lemma `double_is_even` in Coq. (Hint: `Nat.double` and lemmas in the `Nat` module are useful.)

   Please write down the proof script on the exam paper and include the corresponding self-contained .v file in the single .zip file for the instructor.

3. (10 %) Why the law of Distributivity of Disjunction, namely $wp(S, Q_1) \vee wp(S, Q_2) \equiv wp(S, Q_1 \vee Q_2)$, works only for deterministic S but not nondeterministic S? Please explain with an example.

4. (10 %) Prove that $\models wlp(S_1; S_2, q) \leftrightarrow wlp(S_1, wlp(S_2, q))$ which we claimed when proving the completeness of System *PD* (for the validity of a Hoare triple with partial correctness semantics).

   Here, assuming a sufficiently expressive assertion language, $wlp(S, q)$ denotes the assertion p such that $[\![p]\!] = wlp(S, [\![q]\!])$, where $[\![p]\!]$ is defined as $\{\sigma \in \Sigma \mid \sigma \models p\}$ (i.e., the set of states where p holds) and $wlp(S, \Phi)$ as $\{\sigma \in \Sigma \mid \mathcal{M}[\![S]\!](\sigma) \subseteq \Phi\}$. Recall that, for $\sigma \in \Sigma$, $\mathcal{M}[\![S]\!](\sigma) = \{\tau \in \Sigma \mid \langle S, \sigma\rangle \to^* \langle E, \tau\rangle\}$, $\mathcal{M}[\![S]\!](\bot) = \emptyset$, and, for $X \subseteq \Sigma \cup \{\bot\}$, $\mathcal{M}[\![S]\!](X) = \bigcup_{\sigma \in X} \mathcal{M}[\![S]\!](\sigma)$.

5. (20 %) The following C code implements a variant of the partition function for Quicksort. Annotate the code to show its behavior (including particularly an adequate function contract) and prove correctness of your annotation using Frama-C. Please write down the annotations on the exam paper and include the corresponding self-contained .c file in the single .zip file for the instructor.

```
int partition(int* a, int n)
{ int l,r,mid,tmp;

  if (n <= 0) return -1;
  // pivot = a[0];
  l = 0;
  r = n-1;

  while (l < r) {
    while ((l < n) && (a[l] <= a[0]))
      l = l + 1;
    while ((0 <= r) && (a[0] < a[r]))
      r = r - 1;
    if (l < r) {
      tmp = a[l];
      a[l] = a[r];
      a[r] = tmp;
    }
  }


  mid = r;
```

```
    tmp = a[0];
    a[0] = a[mid];
    a[mid] = tmp;

    return mid;
}
```

6. (20 %) Prove the partial correctness of the following program using the Owicki-Gries method.

$$\{acc = 0\}$$

$$\left[\begin{array}{lcl} Q_0 := \mathit{true}; & & Q_1 := \mathit{true}; \\ T := 0; & & T := 1; \\ \textbf{if } Q_1 \textbf{ then} & & \textbf{if } Q_0 \textbf{ then} \\ \quad \textbf{await } T \neq 0 & & \quad \textbf{await } T \neq 1 \\ \textbf{fi}; & \| & \textbf{fi}; \\ s_0 := acc; & & s_1 := acc; \\ acc := s_0 + 1; & & acc := s_1 + 1; \\ Q_0 := \mathit{false}; & & Q_1 := \mathit{false}; \\ T := 0 & & T := 1 \end{array}\right]$$

$$\{acc = 2\}$$