1.) Consider the following problem:

**LOOPS-OR-HALTS (LOH)**

**INSTANCE:** A tuple \((\Pi_1, \Pi_2, I_1, I_2)\), where \(I_1, I_2\) are strings, and \(\Pi_1, \Pi_2\) are programs that take a string as input.

**QUESTION:** Does at least one of the following conditions hold? The conditions are:

(a) \(\Pi_1\) does not halt on \(I_1\)

(b) \(\Pi_2\) halts on \(I_2\)

Provide a reduction from **co-HALTING** to **LOOPS-OR-HALTS**. Argue formally that your reduction is correct.

HINT: Recall that **co-HALTING** is defined as follows:

**co-HALTING**

**INSTANCE:** A pair \((\Pi, I)\), where \(I\) is a string and \(\Pi\) is a program that takes a string as input.

**QUESTION:** Is it true that \(\Pi\) does not halt on \(I\)?

(15 points)

2.) (a) Let \(\varphi^E\) be any \(E\)-formula with Boolean variables \(b_1, \ldots, b_n\). Construct an \(E\)-formula \(\psi^E\) without any Boolean variable by replacing each \(b_i\) \((i = 1, \ldots, n)\) by an equality \(e_i\) of the form \(v_i = w_i\), where \(v_1, w_1, \ldots, v_n, w_n\) are new distinct term variables (identifiers).

Prove: \(\psi^E\) is \(E\)-satisfiable if \(\varphi^E\) is \(E\)-satisfiable.

(11 points)

(b) Answer the following questions to the CDCL algorithm and justify your answers in detail.

i. Suppose that CDCL learns a unit clause \(C\). Given \(C\), to which decision level does CDCL backtrack?

ii. Consider a run of CDCL on a given CNF \(F\), and suppose that the run has terminated. What is the current decision level in CDCL at the time when CDCL terminates?

(4 points)
3.) (a) Is the following rule admissible regarding partial correctness? If it is, prove it using either the Hoare calculus or the definition of partial correctness. If it is not, give a counter-example. In both cases, substantiate any claims that some correctness assertion is true/false or some formula is valid/not valid.

\[ \frac{\{F\} p \{G\} \quad \{F\} q \{G\}}{\{F\} p ; q \{G\}} \]

Remember that a rule is admissible, if for every instance of the rule the following is true: Whenever all formulas among the premises are valid and all correctness assertions among the premises are true, then the correctness assertion in the conclusion is also true.

Some axioms and rules of the Hoare calculus for partial correctness that may be useful:

\[
\begin{align*}
\{F\} \text{skip} \{F\} & \quad \{F\} p \{G\} \quad \{G\} q \{H\} \\
\{F\} \text{abort} \{G\} & \quad \{F\} p ; q \{H\} \\
\{F[v/e]\} v \leftarrow e \{F\} & \quad \{F\} \text{if } e \text{ then } p \text{ else } q \{H\}
\end{align*}
\]

(7 points)

(b) Characterize all programs \(p\) that satisfy \(wp(p, true) = false\), i.e., specify a condition such that the equation holds exactly when \(p\) satisfies the condition. (4 points)

(c) Characterize all programs \(p\) that satisfy \(wlp(p, true) = false\), i.e., specify a condition such that the equation holds exactly when \(p\) satisfies the condition. (4 points)
(a) Provide a non-empty simulation relation $H$ that witnesses $M_1 \leq M_2$, where $M_1$ and $M_2$ are shown below. The initial state of $M_1$ is $s_0$, the initial state of $M_2$ is $t_0$:

**Kripke structure $M_1$:**

(b) Consider the following Kripke structure $M$:

For each of the following formulae $\varphi$,

i. check the respective box if the formula is in CTL, LTL, and/or CTL*, and

ii. list the states $s_i$ on which the formula $\varphi$ holds; i.e. for which states $s_i$ do we have $M, s_i \models \varphi$?
(c) Given a graph, write a C program such that CBMC can determine whether the given graph is 3-colorable. Augment the given code corresponding to the following subtasks. The 2-dimensional array `graph` encodes the adjacency matrix.

```c
#define TRUE 1
#define FALSE 0
#define RED 0
#define GREEN 1
#define BLUE 2
#define N 4 // Number of nodes in the graph

int graph[N][N] = { { 0, 1, 0, 1 }, { 1, 0, 0, 0 }, ... };
int coloring[N];

int nondet_int();
```

i. Write a loop that nondeterministically guesses a coloring for the graph. A coloring assigns to every node of the given graph either the color red, green, or blue.

ii. Write a loop that checks whether the coloring assigns to every node in the graph a color that is different to the colors of its neighbors. Furthermore, ensure that CBMC reports a 3-coloring of the graph in case there exists one.

(6 points)