1.) Consider the following problem:

**PROB**

**INSTANCE:** An undirected graph \( G = (V, E) \) such that \( G \) has a vertex \( v_0 \) that is connected to precisely all other vertices of \( G \), i.e. there exists \( v_0 \in V \) such that (a) \([v_0, v] \in E \) for all \( v \in V \setminus \{v_0\} \), and (b) \([v_0, v_0] \not\in E \).

**QUESTION:** Is it the case that \( G \) is 4-colorable? That is, does there exist a function \( \mu \) from vertices in \( V \) to values in \( \{1, 2, 3, 4\} \) such that \( \mu(v_1) \neq \mu(v_2) \) for any edge \([v_1, v_2] \in E \).

Provide a reduction from **PROB** to **3-COLORABILITY**, and explain the intuition behind your reduction. (15 points)

2.) (a) Show the following:

\( \varphi^{EUF} \) is satisfiable iff \( FC^E \land flat^E \) is satisfiable.

\( FC^E \) and \( flat^E \) are obtained from \( \varphi^{EUF} \) by Ackermann’s reduction.

(Hint: \( FC^E \) is the same for \( \varphi^{EUF} \) and \( \neg \varphi^{EUF} \).) (10 points)

(b) Clarify the logical status of each of the following formulas:

i. \( \varphi_1^{EUF} : f(x) \equiv f(y) \land x \neq y \)

ii. \( \varphi_2^{EUF} : x \equiv y \land f(x) \neq f(y) \)

If the formula is E-valid or E-unsatisfiable, then give a proof based on E-interpretations and semantics. If the formula is E-satisfiable but not E-valid, then present two E-interpretations, one satisfying the formula and one falsifying it. Argue formally why the formula is true respectively false under the considered E-interpretation. (5 points)

3.) (a) Show that \( \{ F \} v := e \{ F[v/e] \} \) is not a sound axiom. (5 points)

(b) Prove that the following correctness assertion is true regarding total correctness. Use the invariant \( 2x + y^2 + y = 4z(z + 1) \land y \geq 0 \).

You may need one of the following annotation rules:

\[ \{ \text{Inv} \} \text{while } e \text{ do } \{ \text{Inv} \land e \land t = t_0 \} \cdots \{ \text{Inv} \land 0 \leq t < t_0 \} \text{ od } \{ \text{Inv} \land \neg e \} \]

\[ \{ \text{Inv} \} \text{while } e \text{ do } \{ \text{Inv} \land e \land t = t_0 \} \cdots \{ \text{Inv} \land (e \Rightarrow 0 \leq t < t_0) \} \text{ od } \{ \text{Inv} \land \neg e \} \]

\[ \{ x = z \land z \geq 0 \} \]

\[ y := 2x; \]

\[ \text{while } y > 0 \text{ do } \]

\[ x := x + y; \]

\[ y := y - 1; \]

\[ \text{od} \]

\[ \{ x \geq 2z \} \]

(10 points)
DEFINITIONS

Let $M_1 = (S_1, I_1, R_1, L_1)$ and $M_2 = (S_2, I_2, R_2, L_2)$ be two Kripke structures.

Simulation

A relation $H \subseteq S_1 \times S_2$ is a simulation relation if for each $(s, s') \in H$ holds:

- $L_1(s) = L_2(s')$, and
- for each $(s, t) \in R_1$ there is a $(s', t') \in R_2$ such that $(t, t') \in H$.

$M_2$ simulates $M_1$ (denoted as $M_1 \leq M_2$), if there is a simulation relation $H \subseteq S_1 \times S_2$ such that

- for each initial state $s \in I_1$ there is an initial state $s' \in I_2$ with $(s, s') \in H$.

Trace Inclusion

Let $M = (S, I, R, L)$ be a Kripke structure. We define the language $\mathcal{L}(M)$ of $M$ to be the set of (infinite) words

$$\mathcal{L}(M) = \{ L(s_1)L(s_2)\cdots | s_1s_2\ldots \text{ is a path of } M \text{ and } s_1 \in I \}$$

We speak of trace inclusion if the language of $M_1$ is included in the language of $M_2$, i.e. $\mathcal{L}(M_1) \subseteq \mathcal{L}(M_2)$.

Show that trace inclusion does not entail simulation, i.e.

$$\mathcal{L}(M_1) \subseteq \mathcal{L}(M_2) \not\Rightarrow M_1 \leq M_2$$

(b) Consider the following Kripke structure:

For each of the following formulae

- determine if the formula is in CTL, LTL, and/or CTL*, and
- state on which states $s_i$ the formula holds

G(a)
X(b \land c)
AG(c)
EF(c)

(c) Use CBMC to solve the Knapsack problem:

Given a set of $N$ items $I = \{i_1, i_2, \ldots, i_N\}$, with respective weights $w(i_j)$ and values $v(i_j)$, we select a subset $I' \subseteq I$ to pack into a backpack.

The backpack can carry only up to a maximal weight $W$ that the selected items must not exceed, i.e.

$$\sum_{i \in I'} w(i) \leq W$$
Decide if there is a subset $I’ \subseteq I$ such that the value of the selected objects is at least $V$, i.e.

$$\sum_{i \in I'} v(i) \geq V$$

Write a C program that implements a *guess and check* routine for the Knapsack problem. Ensure that CBMC reports a valid selection of items $I'$ if one exists.

You may assume the following template:

```c
int nondet_bool(); // nondeterministically returns 0 or 1

int N = 10; // number of items
int values[] = { 15, 22, 13, 44, 82, 12, 32, 41, 33, 14 }; // values of items
int weights[] = { 23, 12, 18, 11, 99, 12, 11, 30, 11, 83 }; // weights of items

int V = 200;
int W = 120;

int main() {
    // add your code here:
    // 1. guess a subset
    // 2. put an assertion such that CBMC reports a solution
    //     (if one exists)
}
```

(5 points)