1.) Consider the following problem:

**ECO-VERTEX-COVER (EVC)**

INSTANCE: Undirected graph \( G = (V, E) \).

QUESTION: Does there exist a set \( N \subseteq V \), where \( N \) is such that: for all edges \( [a, b] \in E \), we have \(|\{a, b\} \cap N| = 1| \)?

We provide next a reduction from \textbf{EVC} to \textbf{2-SAT}. Let \( G = (V, E) \) be an arbitrary undirected graph (i.e., an arbitrary instance of \textbf{EVC}), where \( V = \{v_1, \ldots, v_n\} \). For the reduction we use propositional variables \( x_1, \ldots, x_n \). Then the instance \( \varphi_G \) of \textbf{2-SAT} resulting from \( G \) is defined as follows:

\[
\varphi_G = \bigwedge_{[v_i, v_j] \in E} (x_i \lor x_j) \land (\neg x_i \lor \neg x_j).
\]

**Task:** Prove the \( \Rightarrow \) direction in the proof of correctness of the reduction, i.e., prove the following statement: if \( G \) is a positive instance of \textbf{EVC}, then \( \varphi_G \) is a positive instance of \textbf{2-SAT}.

**Note:** For any property that you use in your proof, make it perfectly clear why this property holds (e.g., “by the problem reduction”, “by the assumption \( X \)”, “by the definition \( X \)”, etc.)

(15 points)

2.) (a) We discussed in class the big picture of the SAT block. Describe in detail how a SAT solver can be employed to decide whether a given equality formula containing uninterpreted functions is valid. Explain the logical relation between the different problems in your description.

(4 points)

(b) In the lecture, we discussed reasoning under different theories. Here we are concerned with LISP-like lists and the theory \( T_{cons}^E \cap T_E \). In a verification attempt of some program, we have to prove the following:

For non-atomic lists \( \ell_1, \ell_2 \), if the “car” of both lists are equal and the “cdr” of both lists are equal, then \( \ell_1 \) is equal to \( \ell_2 \).

We formalize the above statement as follows:

\[
\phi: \quad \left[ \neg \text{atom}(\ell_1) \land \neg \text{atom}(\ell_2) \land \text{car}(\ell_1) \equiv \text{car}(\ell_2) \land \text{cdr}(\ell_1) \equiv \text{cdr}(\ell_2) \right] \quad \Rightarrow \quad \ell_1 \equiv \ell_2
\]

Prove the statement \( T_{cons}^E \) - valid, i.e., show that \( T_{cons}^E \models \phi \).

**Hint:** Besides the equality axioms reflexivity, symmetry and transitivity, the following axioms from \( T_{cons}^E \) are sufficient for a proof:

1. Substitution axioms (functional congruence) for \( cons \):

\[
\forall x_1 \forall x_2 \forall y_1 \forall y_2 \left[ (x_1 \equiv x_2 \land y_1 \equiv y_2) \rightarrow cons(x_1, y_1) \equiv cons(x_2, y_2) \right]
\]

2. Construction:

\[
\forall x \left[ \neg \text{atom}(x) \rightarrow cons(\text{car}(x), \text{cdr}(x)) \equiv x \right]
\]

(11 points)
3. (a) Consider a statement consisting only of the keyword “loopforever”. When executed within a program, the program enters an infinite loop. Define the structural operational and the natural semantics of loopforever-statements. Specify the weakest precondition \( wp(\text{loopforever}, F) \), the weakest liberal precondition \( wlp(\text{loopforever}, F) \), and the strongest postcondition \( sp(F, \text{loopforever}) \) with respect to an arbitrary formula \( F \).

\[ \text{(5 points)} \]

(b) Compute the weakest precondition of the following program with respect to the postcondition \( x = y \).
\[
\begin{align*}
    &y \leftarrow 0; \\
    &z \leftarrow x; \\
    &\text{while } z \neq 0 \text{ do} \\
    &\quad y \leftarrow y + 1; \\
    &\quad z \leftarrow z - 2; \\
&\text{od}
\end{align*}
\]
Remember the weakest precondition of loops: \( wp(\text{while } e \text{ do } p \text{ od}, G) = \exists i (i \geq 0 \land F_i) \), where \( F_0 = \neg e \land G \) and \( F_{i+1} = e \land wp(p, F_i) \).

\[ \text{(10 points)} \]

4.) Bisimulation.

(a) Consider two \( \text{LTL} \) formulas \( \varphi = G(p \rightarrow X (\neg p \land q)) \) and \( \psi = GF(p \land XX (\neg p \land \neg q)) \).

Give two Kripke structures \( K_1 \) and \( K_2 \) satisfying the following:
- \( K_1 \models \varphi \) and \( K_1 \models \psi \);
- \( K_2 \models \varphi \) and \( K_2 \not\models \psi \).

\[ \text{(3 points)} \]

\[ \begin{array}{c}
\text{s1} \\
\text{a} \\
\text{s2} \\
\text{a} \\
\text{s3} \\
\text{s4}
\end{array} \]

\( M_1 \)

(b) For the Kripke structure \( M_1 = (S_1, I_1, R_1, L_1) \) given above, find a Kripke structure \( M_2 = (S_2, I_2, R_2, L_2) \) with the following properties:

i. \( M_2 \) is bisimilar to \( M_1 \).

ii. \( M_2 \) is minimal in the number of states, that is, there is no other Kripke structure \( M = (S, I, R, L) \) that is bisimilar to \( M_1 (M \approx M_1) \) and \( |S| < |S_2| \).

Give a bisimulation relation \( H \) between \( M_1 \) and \( M_2 \).

\[ \text{Hint: Recall the definition of bisimulation from the lectures: } M_1 \text{ and } M_2 \text{ are bisimilar (in signs } M_1 \approx M_2 \text{) iff there is a bisimulation relation } H \subseteq S_1 \times S_2 \text{ with the following properties satisfied for every pair } (s_1, s_2) \in H:\]

i. Labels coincide: \( L_1(s_1) = L_2(s_2) \).

ii. For every transition \( (s_1, t_1) \in R_1 \) there is a matching transition \( (s_2, t_2) \in R_2 \) with \( (t_1, t_2) \in H \). In the other direction, for every transition \( (s_2, t_2) \in R_2 \) there is a matching transition \( (s_1, t_1) \in R_1 \) such that \( (t_1, t_2) \in H \).

iii. For every initial state \( s_1 \in I_1 \) there is a corresponding initial state \( s_2 \in I_2 \) such that \( (s_1, s_2) \in H \). In the other direction, for every initial state \( s_2 \in I_2 \) there is a corresponding initial state \( s_1 \in I_1 \) with \( (s_1, s_2) \in H \).\]
(c) Show that the following theorem holds.

**Theorem.**
Consider two Kripke structures $M_1 = (S_1, I_1, R_1, L_1)$ and $M_2 = (S_2, I_2, R_2, L_2)$ that are bisimilar, i.e., $M_1 \approx M_2$.
Prove that for every path $s_0s_1 \ldots s_k$ of $M_1$ starting with $s_0 \in I_1$ there exists a corresponding path $t_0t_1 \ldots t_k$ of $M_2$ with the following properties:

i. It holds that $t_0 \in I_2$.
ii. For every $i \geq 0$ it holds that $L_1(s_i) = L_2(t_i)$.

**Hint:**
Recall the definition of bisimulation (see Exercise b) and use induction on the length of a path.

(6 points)