1.) Consider the following problem:

**RUN-FOREVER-NO-INPUT (RFNO)**

**INSTANCE:** A program \( \Pi \) such that \( \Pi \) takes no input.

**QUESTION:** Does \( \Pi \) not terminate, i.e. does \( \Pi \) run forever?

By providing a reduction from an undecidable problem to \( \text{RFNO} \), prove that \( \text{RFNO} \) is undecidable. Argue formally that your reduction is correct.

Hint: If a problem \( P \) is undecidable, then its complement \( \overline{P} \) is also undecidable.

(15 points)

2.) (a) Assume \textit{fmiSAT} is a SAT solver that uses unit-propagation (BCP) to build an implication graph (IG). Decisions are done using the DLIS heuristics, and backtracking is done using dependency-directed backtracking.

i. Prove that the implication graph \( IG \) built by \textit{fmiSAT} is acyclic at any time. (6 points)

ii. Show that in a conflict graph the first UIP is uniquely defined, i.e., there is exactly one node in the implication graph which is a first UIP. (5 points)

(b) Use Ackermann’s reduction and translate

\[
\varphi : \quad F(x_1) = F(a) \rightarrow G(F(a), a) = G(F(x_1), b) \land F(F(a)) \neq F(x_1)
\]

to a validity-equivalent E-formula \( \varphi^E \). (4 points)

3.) (a) Determine the strongest postcondition of the weakest liberal precondition of an assignment statement, i.e., compute \( \text{sp(wlp}(v := e, G) \mid v \leftarrow e) \). Why is it different from \( G \)?

Remember the following properties of wlp and sp:

\[
\text{wlp}(v := e, G) = G[v/e]
\]

\[
\text{sp}(F; v := e) = \exists v' (F[v/v'] \land v = e[v/v'])
\]

(5 points)

(b) Show that the following correctness assertion is totally correct. Depending on how you choose the variant, use one of the following annotation rules:

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

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

\[
\{ 1 : x \geq 0 \}
\]

\[
z := x;
\]

\[
y := 0;
\]

\[
\{ \text{Inv} : x = y + z \land z \geq 0 \}
\]

\[
\text{while } z \neq 0 \text{ do }
\]

\[
y := y + 1;
\]

\[
z := z - 1
\]

\[
\text{od}
\]

\[
\{ 2 : x = y \}
\]

(10 points)
4.) Simulation.

(a) Show that $M_2$ simulates $M_1$, i.e., that $M_1 \leq M_2$. (6 points)

(b) For Kripke structures $M_1$ and $M_2$ given above, provide an LTL formula $\varphi$ such that $M_1 \models \varphi$ but $M_2 \not\models \varphi$. Justify why $M_1 \models \varphi$ and $M_2 \not\models \varphi$ hold, respectively. (6 points)

(c) Show that the following theorem for $\text{ACTL}^*$ does not hold for all $\text{CTL}^*$ formulas.

Theorem.

Let $\varphi$ be an $\text{ACTL}^*$ formula and let $M_1, M_2$ be Kripke structures satisfying $M_1 \leq M_2$. Then, $M_2 \models \varphi$ implies $M_1 \models \varphi$.

Hint: Give a $\text{CTL}^*$ formula which contradicts the theorem for $M_1$ and $M_2$ and explain why this formula is a counterexample to the theorem. (3 points)