1.) Recall the **HALTING** problem which takes a program and a string as input, and consider the following variant thereof:

**HALTING-C**

**INSTANCE:** A program \( \Pi \) that takes a string as input, a string \( I \) of even length \( 2 \times n \).

**QUESTION:** Does \( \Pi \) terminate on both strings resulting from \( I \) being cut into two halves, i.e. does \( \Pi \) halt on \( I[1..n] \) and \( I[n+1..2 \times n] \).

(a) The following function \( f \) provides a polynomial-time many-one reduction from **HALTING** to **HALTING-C**; for a program \( \Pi \) and a string \( I \), let \( f(\Pi, I) = (\Pi', I') \) with \( \Pi' = \Pi \) and \( I' = I + I \) (i.e. the concatenation of two copies of string \( I \)).

Show that \( (\Pi, I) \) is a yes-instance of **HALTING** \( \iff \) \( (\Pi', I') \) is a yes-instance of **HALTING-C**.

(b) Please answer the following questions and explain your answers

- Is **HALTING-C** decidable?
- Is **HALTING-C** semi-decidable?
- Is the complement of **HALTING-C** semi-decidable?

(9 points)
2.) (a) Consider the following theory $\mathcal{T}_{\text{tree}}$ of trees with the signature

$$\Sigma_{\text{tree}} = \{\{\text{tree}, \text{le}, \text{ri}\}, \{\text{atom}, \#\}\}.$$  

The axioms of $\mathcal{T}_{\text{tree}}$ include symmetry, reflexivity and transitivity of equality, functional congruence for tree, le, ri, and predicate congruence for atom. In addition we have:

\[
\begin{align*}
\forall x \forall y \le(\text{tree}(x), y) &\equiv x & \text{(left subtree)} \\
\forall x \forall y \text{ri}(\text{tree}(x), y) &\equiv y & \text{(right subtree)} \\
\forall x (\neg \text{atom}(x) \rightarrow \text{tree}(\text{le}(x), \text{ri}(x)) \equiv x) & \text{(construction)} \\
\forall x \forall y \neg \text{atom}(\text{tree}(x, y)) & \equiv y & \text{(atom)}
\end{align*}
\]

We augment theory $\mathcal{T}_{\text{tree}}$ by $\mathcal{T}_E$ (with uninterpreted function symbol $h$) resulting in $\mathcal{T}_{\text{tree}}^E$. Clarify the logical status of each of the following formulas. If one is $\mathcal{T}_{\text{tree}}^E$-valid or $\mathcal{T}_{\text{tree}}^E$-unsatisfiable, then prove it using the semantic argument method. If one is $\mathcal{T}_{\text{tree}}^E$-satisfiable but not $\mathcal{T}_{\text{tree}}^E$-valid, then present a satisfying and a falsifying interpretation. Argue formally that the formula evaluates to true resp. false under the constructed interpretations.

$$\varphi_0: \le(a) \equiv \le(b) \wedge \text{ri}(a) \equiv \text{ri}(b) \wedge \neg \text{atom}(a) \wedge \neg \text{atom}(b) \rightarrow h(a) = h(b)$$
$$\varphi_1: \neg \text{atom}(x) \wedge \le(x) \equiv y \wedge \text{ri}(x) \equiv z \wedge x \not\equiv \text{tree}(y, z)$$

(8 points)

(b) Consider the following clause set $\hat{\delta}(\varphi)$ which has been derived from an (unknown) formula $\varphi$ by an improved version of Tseitin’s translation (atoms have not been labeled and $\equiv$ means $\neg \equiv$).

\[
C_1: \ell_1 \lor x_1 \lor x_2 \\
C_2: \ell_1 \lor x_1 \lor x_2 \\
C_3: \ell_1 \lor x_1 \lor x_2 \\
C_4: \ell_1 \lor x_1 \lor x_2 \\
C_5: \ell_2 \lor x_2 \lor x_3 \\
C_6: \ell_2 \lor x_2 \lor x_3 \\
C_7: \ell_2 \lor x_2 \lor x_3 \\
C_8: \ell_2 \lor x_2 \lor x_3 \\
C_9: \ell_3 \lor \ell_1 \lor \ell_2 \\
C_{10}: \ell_3 \lor \ell_1 \lor \ell_2 \\
C_{11}: \ell_3 \lor \ell_1 \lor \ell_2 \\
C_{12}: \ell_3 \lor \ell_1 \lor \ell_2 \\
C_{13}: \ell_4 \lor x_2 \\
C_{14}: \ell_4 \lor x_2 \\
C_{15}: \ell_4 \lor x_2 \\
C_{16}: \ell_5 \lor x_1 \lor x_3 \\
C_{17}: \ell_5 \lor x_1 \lor x_3 \\
C_{18}: \ell_5 \lor x_1 \lor x_3 \\
C_{19}: \ell_6 \lor \ell_4 \lor \ell_5 \\
C_{20}: \ell_6 \lor \ell_4 \lor \ell_5 \\
C_{21}: \ell_6 \lor \ell_4 \lor \ell_5 \\
\]

(i) Reconstruct $\varphi$ from $\hat{\delta}(\varphi)$.

(ii) Prove the validity of $\varphi$ by resolution (no additional translation to normal form is allowed!). You are allowed to add a single unit clause (i.e., a clause containing exactly one literal). Please explain your approach!

(7 points)
3.) (a) Let \( p \) be the following IMP program loop, containing the integer-valued program variables \( x, y \):

\[
\text{while } x \neq y \text{ do } \\
x := x - 2 \times y; \\
y := 2 \times y - x \\
\text{od}
\]

Which of the following program assertions are inductive loop invariants of \( p \)?

- \( I_1 : x = 0 \land y = 0 \)
- \( I_2 : x + y = 0 \)
- \( I_3 : x = 0 \land y = 1 \)

Give formal details justifying your answer. That is, if an assertion is an inductive loop invariant, provide a formal proof of it based on Hoare logic or using weakest liberal preconditions. If an assertion is not an inductive loop invariant, give a counterexample.

\textbf{Note:} You need to use the definition of an assertion being an inductive invariant.

(9 points)

(b) Let \( A \) be an arbitrary post-condition. Which of the following Hoare triples are valid partial correctness assertions?

- \( \{ \text{true} \} \text{ skip } \{ A \} \)
- \( \{ \text{false} \} \text{ skip } \{ A \} \)

Give formal details justifying your answer. That is, if a triple is valid, provide a formal proof of it based on Hoare logic. If an assertion is not valid, give a counterexample (that is, an instance of \( A \) for which the triple does not hold).

(4 points)

(c) Consider the Hoare triple \( \{ A \} p \{ B \} \), where \( p \) is an arbitrary IMP program and \( A, B \) are arbitrary program assertions. Assume there is a state \( \sigma \) that satisfies \( A \) and there is a state \( \sigma' \) such that \( < p, \sigma > \rightarrow \sigma' \) and \( \sigma' \) satisfies \( B \).

Given this information, is \( \{ A \} p \{ B \} \) partially correct?

Answer the question with either a Yes or a No answer, and provide a short justification for your answer.

(2 points)
4.) (a) Consider the Kripke structures $M_1$ and $M_2$. The initial state of $M_1$ is $s_0$, the initial
state of $M_2$ is $t_0$.

Kripke structure $M_1$:

Kripke structure $M_2$:

\begin{itemize}
  \item[i.] Check whether $M_2$ simulates $M_1$, i.e., provide a simulation relation that witnesses $M_1 \preceq M_2$, or briefly explain why $M_2$ does not simulate $M_1$.
  \item[ii.] Check whether $M_1$ simulates $M_2$, i.e., provide a simulation relation that witnesses $M_2 \preceq M_1$, or briefly explain why $M_1$ does not simulate $M_2$.
\end{itemize}

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

For each of the following formulae $\varphi$,

i. indicate whether 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 $M, s_i \models \varphi$?

(If $\varphi$ is a path formula, list the states $s_i$ such that $M, s_i \models A\varphi$.)

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>CTL</th>
<th>LTL</th>
<th>CTL*</th>
<th>States $s_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A[(b) U (a)]$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$G(a)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$FG(c)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$EGF(c)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$EX(c)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(5 points)
(c) **LTL tautologies**

An LTL formula is a *tautology* if it holds for every Kripke structure $M$ and every path $\pi$ in $M$. For each of the following formulas, prove that it is a tautology, or find a Kripke structure $M$ and path $\pi$ in $M$ for which the formula does not hold and justify your answer.

i. $(b \mathbf{U} G(a \land \neg b)) \Rightarrow G(b \Rightarrow F a)$

ii. $G(b \Rightarrow F a) \Rightarrow (b \mathbf{U} G(a \land \neg b))$

(6 points)

---

Grading scheme: 0–29 nicht genügend, 30–35 genügend, 36–41 befriedigend, 42–47 gut, 48–60 sehr gut