1.) Consider the following problem:

**FIND-INPUT**

**INSTANCE:** A pair \((\Pi_1, \Pi_2)\) of programs such that:
- \(\Pi_1\) takes a *string* as input and outputs a string, and
- \(\Pi_2\) takes an *integer* as input and outputs a string.

It is guaranteed that \(\Pi_1\) and \(\Pi_2\) terminate on any input.

**QUESTION:** Does there exist a string \(S\) and an integer \(n\) such that \(\Pi_1(S) = \Pi_2(n)\)?

Here \(\Pi_1(S)\) is the string returned by \(\Pi_1\) on the string \(S\), and \(\Pi_2(n)\) is the string returned by \(\Pi_2\) on the integer \(n\).

Prove that the problem **FIND-INPUT** is semi-decidable. For this, describe a procedure that shows the semi-decidability of the problem (i.e. a semi-decision procedure for **FIND-INPUT**) and argue that it is correct.

(15 points)

2.)

(a) The removal of Boolean variables from an E-formula is defined as follows:

**Definition.** 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 \(b_i\) by \(v_{b_i, 1} = v_{b_i, 2}\) where \(v_{b_i, 1}, v_{b_i, 2}\) are two new term variables (identifiers).

Prove that \(\varphi^E\) is E-satisfiable if \(\psi^E\) is E-satisfiable. (10 points)

(b) Transform the EUF-formula \(\varphi^{EUF}\) below to an E-formula \(\varphi^E\) using Ackermann’s reduction. Note that \(\varphi^{EUF}\) contains an uninterpreted predicate, which requires special treatment first.

\[
\varphi^{EUF} : F(F(x_1)) = G(x_2, G(x_3, x_4), F(x_2)) \rightarrow p(x_1, y).
\]

(5 points)

3.)

(a) We use \([x]\) to denote the function associated with the syntactic entity \(x\), where \(x\) may be a program, an expression, or one of the pre-defined operators. Investigate for each of the three cases, whether \([x] = [y]\) implies \(x = y\) for arbitrary programs/expressions/operators \(x\) and \(y\). If yes, give an argument for it, if not, give a counterexample. Note that these are three separate questions. What about the converse: Does \(x = y\) necessarily imply \([x] = [y]\)? (5 points)

(b) Show that the following correctness assertion is totally correct. Describe the function computed by the program; assume \(x\) and \(y\) to be the inputs and \(z\) the output of the program.

Depending on how you choose the variant, use one of the following annotation rules:

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

(5 points)
{1: x \geq 1 \land y \geq 2}
\begin{align*}
u & := y; \\
z & := 0; \\
\{ \text{Inv: } u = y^{x+1} \land u \leq x \ast y \land y > 0 \}
\end{align*}
while u \leq x do
\begin{align*}
u & := u \ast y; \\
z & := z + 1
\end{align*}
\text{od}
\{2: y^2 \leq x < y^{x+1}\}

(10 points)

4.) Simulation.

\begin{minipage}{0.4\textwidth}
\begin{center}
\begin{tikzpicture}
\node[draw,shape=circle] (s1) at (0,0) {$s_{1}$};
\node[draw,shape=circle] (s2) at (1,0) {$s_{2}$};
\node[draw,shape=circle] (s3) at (2,0) {$s_{3}$};
\node[draw,shape=circle] (s4) at (0,-1) {$s_{4}$};
\node[draw,shape=circle] (s5) at (1,-1) {$s_{5}$};
\node[draw,shape=circle] (s6) at (2,-1) {$s_{6}$};
\draw[->] (s1) -- (s2);
\draw[->] (s2) -- (s3);
\draw[->] (s3) -- (s4);
\draw[->] (s4) -- (s1);
\draw[->,loop above] (s1) -- (s1);
\draw[->,loop below] (s6) -- (s6);
\end{tikzpicture}
\end{center}
\end{minipage}
\begin{minipage}{0.4\textwidth}
\begin{center}
\begin{tikzpicture}
\node[draw,shape=circle] (s1) at (0,0) {$s'_{1}$};
\node[draw,shape=circle] (s2) at (1,0) {$s'_{2}$};
\node[draw,shape=circle] (s3) at (2,0) {$s'_{3}$};
\node[draw,shape=circle] (s4) at (0,-1) {$s'_{4}$};
\node[draw,shape=circle] (s5) at (1,-1) {$s'_{5}$};
\node[draw,shape=circle] (s6) at (2,-1) {$s'_{6}$};
\draw[->] (s1) -- (s2);
\draw[->] (s2) -- (s3);
\draw[->] (s3) -- (s4);
\draw[->] (s4) -- (s1);
\draw[->,loop above] (s1) -- (s1);
\draw[->,loop below] (s6) -- (s6);
\end{tikzpicture}
\end{center}
\end{minipage}

\begin{enumerate}
\item[(a)] Show that $M_2$ simulates $M_1$, i.e., that $M_1 \leq M_2$.
\end{enumerate}

(6 points)

\begin{enumerate}
\item[(b)] For Kripke structures $M_1$ and $M_2$ given above, provide an \textit{ECTL}⋆ formula $\varphi$ such that $M_1 \not\models \varphi$ but $M_2 \models \varphi$. Justify why $M_1 \not\models \varphi$ and $M_2 \models \varphi$ holds, respectively.
\end{enumerate}

(6 points)

\begin{enumerate}
\item[(c)] Show that the following theorem holds.
\end{enumerate}

\begin{block}{Theorem.}
If $M_1 \leq M_2$ holds, then for all \textit{ECTL}⋆ formulas $\varphi$ holds that if $M_1 \models \varphi$, then $M_2 \models \varphi$.
\end{block}

\begin{block}{Hint:}
You can use the following theorem from the lecture:
\end{block}

\begin{block}{Theorem.}
If $M_1 \leq M_2$ holds, then for all \textit{ACTL}⋆ formulas $\varphi$ holds that if $M_2 \models \varphi$, then $M_1 \models \varphi$.
\end{block}

(3 points)