1.) Consider the following decision problem:

**DIFFERENT RUNTIME**

*INSTANCE:* A tuple \((\Pi, I_1, I_2)\), where \(\Pi\) is a program that takes a string as input, and \(I_1, I_2\) are strings.

*QUESTION:* Does \(\Pi\) hold on \(I_1\) within a different number of computation steps than on \(I_2\)?

Remark: If a program \(\Pi\) neither terminates on \(I_1\) nor on \(I_2\), we say that \(\Pi\) requires the same number of computation steps (i.e., infinitely many) for \(I_1\) and \(I_2\).

(1) By providing a suitable reduction from the HALTING problem, prove undecidability of **DIFFERENT RUNTIME**.

(2) Is **DIFFERENT RUNTIME** semi-decidable? Explain your answer. (15 points)

2.) (a) Recall that arrays are represented functionally. For instance, \(\text{write}(a, i, e)\) is denoted by \(a\langle i\triangleleft e\rangle\). Similarly, \(\text{read}(a, k)\) is denoted by \(a\lbrack k\rbrack\). Show that the following formula

\[
\begin{align*}
&\forall a, i, j \ (i \neq j \rightarrow a\langle i\triangleleft e\rangle\lbrack j\rbrack \neq a\lbrack j\rbrack) \\
&\forall a, v, i, j \ (i \neq j \rightarrow a\langle i\triangleleft v\rangle\lbrack j\rbrack \neq a\lbrack j\rbrack)
\end{align*}
\]

is \(T_{\lambda}\)-unsatisfiable. Please justify any step in your proof in detail.

Besides the equality axioms, you have the following ones for the arrays.

i. \(\forall a, i, j \ (i = j \rightarrow a\lbrack i\rbrack = a\lbrack j\rbrack)\) (array congruence)

ii. \(\forall a, v, i, j \ (i = j \rightarrow a\langle i\triangleleft v\rangle\lbrack j\rbrack = v)\) (read-over-write 1)

iii. \(\forall a, v, i, j \ (i \neq j \rightarrow a\langle i\triangleleft v\rangle\lbrack j\rbrack = a\lbrack j\rbrack)\) (read-over-write 2)

(12 points)

(b) Show the soundness of the following variant of the resolution rule.

\[
\frac{C \lor p \lor q \ D \lor \lnot p \ E \lor \lnot q }{C \lor D \lor E}
\]

(3 points)

3.) Prove that the following correctness assertion is true regarding total correctness. Use the invariant \(x \cdot m \leq n < y \cdot m \land m > 0\). Describe the function computed by the program, if we consider \(m\) and \(n\) as the inputs and \(x\) as the result.

Some annotation rules you might need:

\[
\begin{align*}
{F}v := e & \iff \{F\}v := e\{\exists v' (F[v/v'] \land v = e[v/v'])\} \\
\text{if } e \text{ then } \{F\} \cdots \text{ else } \{G\} & \iff (e \Rightarrow F) \land (\lnot e \Rightarrow G) \text{ if } e \text{ then } \{F\} \cdots \text{ else } \{G\} \\
\{F\} \text{ if } e \text{ then } \cdots \text{ else } & \iff \{F\} \text{ if } e \text{ then } \{F \land e\} \cdots \text{ else } \{F \land \lnot e\} \\
\text{while } e \text{ do } \cdots \text{ od } & \iff \{\text{Inv}\} \text{ while } e \text{ do } \{\text{Inv} \land e \land t = t_0\} \cdots \{\text{Inv} \land (e \Rightarrow 0 \leq t < t_0)\} \text{ od } \{\text{Inv} \land \lnot e\}
\end{align*}
\]
\[ \{ m > 0 \land n \geq 0 \} \]

\begin{verbatim}
x := 0;
y := n + 1;
while \( x + 1 \neq y \) do
    \( z := (x + y)/2; \)
    if \( z \times m > n \) then
        \( y := z \)
    else
        \( x := z; \)
    fi
od
\{ x \times m \leq n < (x + 1) \times m \}
\end{verbatim}

(15 points)

4.) (a) Provide a non-empty simulation relation \( H \) that witnesses \( M_1 \leq M_2 \), where \( M_1 \) and \( M_2 \) are shown below. The initial state of \( M_1 \) is \( s_0 \), the initial state of \( M_2 \) is \( t_0 \):

Kripke structure \( M_1 \): 

\begin{itemize}
  \item \( s_0: \{ b \} \)
  \item \( s_1: \{ a \} \)
  \item \( s_2: \{ c \} \)
  \item \( s_3: \{ b \} \)
  \item \( s_4: \{ a \} \)
\end{itemize}

Kripke structure \( M_2 \): 

\begin{itemize}
  \item \( t_0: \{ b \} \)
  \item \( t_1: \{ c \} \)
  \item \( t_2: \{ b \} \)
  \item \( t_3: \{ b \} \)
  \item \( t_4: \{ a \} \)
  \item \( t_5: \{ c \} \)
\end{itemize}

(4 points)

(b) Consider the following Kripke structure \( M \):
For each of the following formulae $\varphi$,

i. check the respective box if the formula is in CTL, LTL, and/or $\text{CTL}^*$, and

ii. list the states $s_i$ on which the formula $\varphi$ holds; i.e. for which states $s_i$ do we have $M, s_i \models \varphi$?

Note that by a $\text{CTL}^*$ formula, we here mean a $\text{CTL}^*$ state formula or a $\text{CTL}^*$ path formula. Also recall that for an LTL formula $\varphi$ we have $M, s \models \varphi$ if and only if $\pi \models \varphi$ for all paths in $M$ that start at $s$.

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>CTL</th>
<th>LTL</th>
<th>$\text{CTL}^*$</th>
<th>States $s_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G(c)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(a) U (a)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{AAF}(a \land c)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AX(a \land b)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E[(b \land c) U (c)]$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(5 points)

(c) **LTL tautologies**

Prove or disprove (e.g. by providing a counter-example) the following LTL formulas:

i. 

\[
\neg(p U q) \rightarrow (\neg Gp \lor \neg Fq)
\]

ii. 

\[
G(p U q) \rightarrow Gp
\]

iii. 

\[
F(p U q) \rightarrow Fq
\]

(6 points)