1.) Consider the following problem:

**SOME-INPUT**

INSTANCE: A program (i.e. a source code) $\Pi$ such that $\Pi$ takes one string as input and outputs either $true$ or $false$. Each input string for $\Pi$ uses only symbols 0 and 1.

QUESTION: Does there exist an input string $I$ for $\Pi$ such that: $\Pi$ outputs $true$ on $I$ in at most $|I|^2$ computation steps? Here $|I|$ denotes the length of $I$.

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

**Hint:** For your construction of a semi-decision procedure you may use another procedure $\Pi'$ that does the following:

(a) $\Pi'$ takes as input a program $\Pi$, a string $I$ and a natural number $n$.

(b) $\Pi'$ checks whether $\Pi$ outputs $true$ on $I$ in at most $n$ computation steps (intuitively, to check this the program $\Pi'$ simulates the first $n$ steps of the computation of $\Pi$ on $I$).

(15 points)

2.) Compute the definitional form (Tseitin form) of the propositional formula $\varphi: (\neg q \rightarrow \neg p) \rightarrow (p \rightarrow q)$.

**Hint:** Draw a formula tree, label its nodes and derive the Tseitin form.

(3 points)

3.) Given the following clauses, draw an implication graph starting with $x_3 = 1\oplus 1$.

$C_1: \neg x_1 \lor x_2$
$C_2: \neg x_1 \lor x_3 \lor x_5$
$C_3: \neg x_2 \lor x_4$
$C_4: \neg x_3 \lor \neg x_4$
$C_5: x_1 \lor x_5 \lor \neg x_2$
$C_6: x_2 \lor x_3$
$C_7: x_2 \lor \neg x_3$
$C_8: x_6 \lor \neg x_5$

Is the clause set unsatisfiable? If yes, then give a proof; if not, then provide a model.

(4 points)

4.) Show the following:

$\varphi^{uf}$ is satisfiable iff $FC^E \land flat^E$ is satisfiable.

$FC^E$ and $flat^E$ are obtained from $\varphi^{uf}$ by Ackermann’s reduction. (Hints: $FC^E$ is the same for $\varphi^{uf}$ and $\neg \varphi^{uf}$, i.e., $FC^E(\varphi^{uf}) = FC^E(\neg \varphi^{uf})$ and $flat^E(\neg \varphi^{uf}) = \neg flat^E(\varphi^{uf})$.)

(8 points)

5.) Suppose we extend the toy language by a new statement type consisting only of the keyword “loop”. When executed within a program, the program enters an infinite loop.

(a) Extend the structural operational and the natural semantics for loop-statements.

(b) Define correct axioms for the Hoare calculus for partial/total correctness. (The axioms should not refer to an unspecified invariant. This is not necessary, since the loop-statement is completely determined.)

(4 points)
(c) Specify the weakest precondition \( \text{wp}(\text{loop}, F) \), the weakest liberal precondition \( \text{wlp}(\text{loop}, F) \), and the strongest postcondition \( \text{sp}(\text{loop}, F) \) with respect to an arbitrary formula \( F \).

(3 points)

6.) Prove the total correctness of the assertion below. Describe the function computed by the program when considering \( x \) and \( y \) as the inputs and \( z \) as the output.

\[
\begin{align*}
\{ \text{Pre: } & x \geq 2 \land y \geq 1 \} \\
& z \leftarrow 0; \\
& a \leftarrow x; \\
\{ \text{Inv: } & a = x^{z+1} \land 1 \leq a \leq x \times y \land x \geq 2 \} \\
\text{while } & a \leq y \text{ do} \\
& z \leftarrow z + 1; \\
& a \leftarrow a \times x \\
\text{od;}
\{ \text{Post: } & x^{z} \leq y < x^{z+1} \}
\end{align*}
\]

(12 points)

7.) CTL and LTL:
Find a Kripke structure \( K \) with initial state \( s \) that has the property \( \text{AGEF} p \) at state \( s \), but not \( \text{AGF} p \). Justify your choice.

(4 points)

8.) CTL Model Checking Algorithm:
Let \( K = (S, T, L) \) be a Kripke structure and \( p \) be an atomic proposition. Give a graph-theoretic algorithm for computing the set of states where \( \text{AG} p \) holds.

(3 points)

9.) Bisimulation:
Given two models \( M_1 = (S_1, I_1, R_1, L_1) \) and \( M_2 = (S_2, I_2, R_2, I_2) \), give an algorithm that determines whether \( M_1 \) is bisimilar to \( M_2 \), i.e., whether \( M_1 \equiv M_2 \) holds.

(4 points)

10.) Predicate Abstraction:
Consider the following program, where the semantics of the statement \((x, y) := (a, b)\) is that the values \( a \) and \( b \) are simultaneously assigned to the variables \( x \) and \( y \).

```c
int x, y;

void foo() {
    (x, y) := (0, 0);

    while (x < 50) {
        (x, y) := (x + 1, y + 1);
    }

    assert(y >= 50);
}
```

(a) Provide a labeled transition system for the given program.

(b) Provide an abstraction for the labeled transition system that uses the predicate \( x < 50 \).

(c) Give an error trace in the abstraction.

(d) Introduce a new predicate to refine the abstraction to get rid of the error state. Give the new abstraction.

(4 points)