1.) We want to prove the *NP-hardness* of **SUBSET SUM**. Your task is to give a polynomial time reduction \( R \) from **PARTITION** (which is NP-complete) to **SUBSET SUM**. Additionally, prove the “\( \Leftarrow \)” direction in the proof of correctness of the reduction, i.e., let \( x \) denote an arbitrary instance of the **PARTITION** problem and let \( R(x) \) denote the corresponding instance of the **SUBSET SUM** problem. You have to prove the following statement: if \( R(x) \) is a positive instance of **SUBSET SUM**, then \( x \) is a positive instance of **PARTITION**.

The definition of these two problems is given below:

**PARTITION:**
Instance: A finite set of \( n \) positive integers \( P = \{p_1, p_2, \ldots, p_n\} \).
Question: Can the set \( P \) be partitioned into two subsets \( P_1, P_2 \) such that the sum of the numbers in \( P_1 \) equals the sum of the numbers in \( P_2 \)?

**SUBSET SUM:**
Instance: A finite set of integer numbers \( S = \{a_1, a_2, \ldots, a_n\} \) and an integer number \( t \).
Question: Does there exist a subset \( S' \subseteq S \), s.t. the sum of the elements in \( S' \) is equal to \( t \), i.e., \( \left( \sum_{a_i \in S'} a_i \right) = t \)?

(15 points)

2.) (a) Let \( \varphi^E \) be the following equality logic formula:
\[
(x_5 = x_6 \lor x_4 \neq x_5) \land x_4 \neq x_6 \land x_4 = x_2 \land x_2 = x_3 \land (x_3 \neq x_1 \lor x_4 = x_1)
\]
Apply the *Sparse Method* to obtain an equisatisfiable propositional formula: Apply simplification/preprocessing to obtain an equi-satisfiable \( \varphi^E_S \); draw the nonpolar equality graph \( G^E_{NP}(\varphi^E_S) \); make \( G^E_{NP}(\varphi^E_S) \) chordal; compute the propositional skeleton \( e(\varphi^E_S) \) and transitivity constraints \( B_t \); and give the resulting propositional formula. (6 points)

(b) Given a set \( C \) of clauses, a conflict graph \( G \) with respect to \( C \), and some clause \( D \). Prove the following:
If \( D \) was learned from \( G \) following the first-UIP scheme, then the following formula is valid:
\[
\left( \bigwedge_{C \in C} C \right) \rightarrow D
\]
(9 points)

3.) (a) Show that the following version of the ‘logical consequence’-rule is not sound, by means of a counter-example; argue that it is a counter-example.

\[
\frac{\{ F \} p \{ G \} \quad G' \Rightarrow G}{\{ F \} p \{ G' \}}
\]

(A rule being sound means: “Whenever all premises are true, the conclusion is also true.”) (5 points)

(b) Show that the following correctness assertion is totally correct. Describe the function computed by the program if we consider \( n \) as its input and \( a \) as its output.

*Hint:* Use the annotation rule
\[
\text{while } e \text{ do } \cdots \text{od} \rightarrow \{ \text{Inv} \} \text{while } e \text{ do } \{ \text{Inv} \land e \land t = t_0 \} \cdots \{ \text{Inv} \land (e \Rightarrow 0 \leq t < t_0) \} \text{od} \{ \text{Inv} \land \neg e \}
\]
\{ n \geq 1 \}
\begin{align*}
a & \leftarrow 0; \\
b & \leftarrow 2; \\
\{ \text{Inv: } b = 2^{a+1} \land 0 < b \leq 2n \}
\end{align*}
\begin{align*}
\text{while } b \leq n \\
& \text{do} \\
a & \leftarrow a + 1; \\
b & \leftarrow b + b \\
\od \\
\{ 2^a \leq n < 2^{a+1} \}
\end{align*}

(10 points)

4.) Computation Tree Logic.

Let \( AP \) be a set of propositional symbols, and \( AP' \subseteq AP \) be a subset of \( AP \).

We recall the definition of ACTL formulae over \( AP \):
- \( p \in AP \) and \( \neg p \in AP \) are ACTL formulae,
- if \( \varphi \) and \( \psi \) are ACTL formulae, then \( \varphi \land \psi \), \( \varphi \lor \psi \), \( \text{AX} \varphi \), \( \text{AG} \varphi \), and \( \text{A} [\varphi \text{ U } \psi] \) are ACTL formulae.

Let \( M = (S, I, R, L) \) and \( M' = (S', I', R', L') \) be two Kripke structures related as follows:
- \( S = S' \), \( I = I' \), \( R = R' \), and
- \( L'(s) = L(s) \cap AP' \), where \( s \in S \).

Let \( \hat{M} = (\hat{S}, \hat{I}, \hat{R}, \hat{L}) \) be a Kripke structure related to \( M' \) as follows:
- \( \hat{S} = 2^{AP'} \), i.e., a state \( \hat{s} \in \hat{S} \) is a subset of \( AP' \),
- \( \hat{I} = \{ \hat{s} \in \hat{S} \mid \exists s \in S'. L'(s) = \hat{s} \} \), i.e., a state \( \hat{s} \in \hat{S} \) is an initial state of \( \hat{M} \) if there is an initial state \( s \in S' \) such that \( s \) is labeled with \( \hat{s} \),
- \( \hat{R} = \{ (\hat{s}, \hat{t}) \in \hat{S} \times \hat{S} \mid \exists s, t \in S. \hat{s} = L'(s) \land \hat{t} = L'(t) \land (s, t) \in R' \} \), i.e., for each transition \( (\hat{s}, \hat{t}) \in \hat{R} \) there are states \( s, t \in S' \) such that there is a transition from \( s \) to \( t \) and \( s \) is labeled with \( \hat{s} \) and \( t \) is labeled with \( \hat{t} \),
- \( \hat{L}(\hat{s}) = \hat{s} \) for all \( \hat{s} \in \hat{S} \), i.e., each state \( \hat{s} \in \hat{S} \) is labeled with the atomic propositions it contains.

(a) Prove that for any ACTL formula \( \varphi \) over propositions from \( AP' \) the following holds:

\[ M \models \varphi \text{ if and only if } M' \models \varphi \]

\textit{Hint:} Use the semantics of ACTL. You can either use an induction on the structure of the formula (structural induction) or an induction on the formula length.

(5 points)

(b) Prove that for any ACTL formula \( \varphi \) over propositions from \( AP' \) the following holds:

\[ \text{If } \hat{M} \models \varphi, \text{ then } M' \models \varphi \]

\textit{Hint:} You can use the following theorem from the lecture:

\textit{Let } \( M_1 \) \text{ and } \( M_2 \) \text{ be Kripke structures such that } M_1 \preceq M_2. \text{ Let } \varphi \text{ be an ACTL* formula. If } M_2 \models \varphi, \text{ then } M_1 \models \varphi. \]

(10 points)