Block 1.) Assume for this exercise that all strings are over an alphabet \{0, 1\} and contain at least one symbol. For a string \( I = b_1b_2\ldots b_n \) we define the bit-flipped string \( \text{flip}(I) \) as the bitwise complement of \( I \), i.e., \( \text{flip}(I) = c_1c_2\ldots c_n \) where \( c_i = 1 - b_i \). Example: for \( I = 00110 \) we get \( \text{flip}(I) = 11001 \). Recall that \text{co-HALTING} is the complement of the \text{HALTING} problem; consider the following variant thereof.

**BITFLIP-HALTING**

**INSTANCE:** A program \( \Pi \) that takes a string as input, and a string \( I \).

**QUESTION:** Does \( \Pi \) halt on \( I \), but not halt on the bit-flipped string \( \text{flip}(I) \)?

1.a) The following function \( g \) provides a computable many-one reduction from the problem \text{co-HALTING} to \text{BITFLIP-HALTING}: \( g((\Pi, I)) = (\Pi_1, I_1) \), where \( I_1 = \text{flip}(I) \) and \( \Pi_1 \) is given as follows:

\[
\Pi_1(string~S)\
\quad\text{if}(S = \text{flip}(I))\
\quad\quad\text{return}\
\quad\}
\]

\[
\Pi_1(S)\
\quad\text{return}\
\]

Show the correctness of the reduction, i.e., show that \((\Pi, I)\) is a positive instance of \text{co-HALTING} if and only if \( g((\Pi, I)) \) is a positive instance of \text{BITFLIP-HALTING}.

(9 points)
1.b) Check which statements are true/false. 1 point for each correct answer, -1 for each incorrect answer, 0 for no answer. Negative points do not carry over to other exercises.

You may use the fact that co-HALTING is undecidable and not even semi-decidable.

true    false
○  ○ The correctness of the reduction in (a) proves undecidability of BITFLIP-HALTING.
○  ○ If the reduction in (a) is correct, we know that BITFLIP-HALTING must be semi-decidable.
○  ○ If the reduction in (a) is correct, we know that BITFLIP-HALTING cannot be semi-decidable.
○  ○ If the reduction in (a) is correct, we know that BITFLIP-HALTING cannot be decidable.
○  ○ If the reduction in (a) is correct, we know that the complement of BITFLIP-HALTING must be semi-decidable.
○  ○ If the reduction in (a) is correct, we know that BITFLIP-HALTING cannot be in NP.

(6 points)
Block 2.

2.a) Use the semantic argument method together with the stepwise induction principle to prove the $T_{cons^+}$-validity of the formula

$$
\varphi: \forall u \forall v \left( (\text{flat}(u) \land \text{flat}(v)) \rightarrow \text{flat}(\text{concat}(u,v)) \right).
$$

The tasks are as follows:

i. Base case: Use the semantic argument method to prove that the formula

$$
\psi: \forall v \left( (\text{atom}(u) \land \text{flat}(u) \land \text{flat}(v)) \rightarrow \text{flat}(\text{concat}(u,v)) \right)
$$

is $T_{cons^+}$-valid.

ii. Let the induction hypothesis be that, for some list $v$, the formula

$$
\forall w \left( (\text{flat}(v) \land \text{flat}(w)) \rightarrow \text{flat}(\text{concat}(v,w)) \right)
$$

is $T_{cons^+}$-valid. Use the semantic argument method to prove that the formula

$$
\forall w \left( (\text{flat}(\text{cons}(u,v)) \land \text{flat}(w)) \rightarrow \text{flat}(\text{concat}(\text{cons}(u,v),w)) \right)
$$

is $T_{cons^+}$-valid.

**Hint:** The available axioms are the equality axioms, the substitution axioms for the function and predicate symbols, and

- $\forall u \forall v \text{car}(\text{cons}(u,v)) \equiv u$ (left projection)
- $\forall u \forall v \text{cdr}(\text{cons}(u,v)) \equiv v$ (right projection)
- $\forall u \left( (\neg \text{atom}(u)) \rightarrow \text{cons}(\text{car}(u), \text{cdr}(u)) \equiv u \right)$ (construction)
- $\forall u \forall v \neg \text{atom}(\text{cons}(u,v))$ (atom)
- $\forall u \forall v \left( \text{atom}(u) \rightarrow \text{concat}(u,v) \equiv \text{cons}(u,v) \right)$ (concat atom)
- $\forall u \forall v \forall w \text{concat}(\text{cons}(u,v),w) \equiv \text{cons}(u,\text{concat}(v,w))$ (concat list)
- $\forall u \left( \text{atom}(u) \rightarrow \text{flat}(u) \right)$ (flat atom)
- $\forall u \forall v \left( \text{flat}(\text{cons}(u,v)) \leftrightarrow (\text{atom}(u) \land \text{flat}(v)) \right)$ (flat list)
2.b) Consider the clauses $C_0, \ldots, C_6$ in dimacs format (in this order from top to bottom, shown in the box) which are given as input to a SAT solver.

- Apply CDCL using the convention that if a variable is assigned as a decision, then it is assigned 'true'. Select variables as decisions in increasing order of their respective integer IDs in the dimacs format, starting with variable 1. Recall that unit clauses require a special treatment.

- When the first conflict occurs, draw the complete implication graph, mark the first UIP, give the resolution derivation of the learned asserting clause that corresponds to the first UIP, and stop CDCL. You do not have to solve the formula!

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-4</td>
<td>5</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>-4</td>
<td>6</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>-6</td>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-7</td>
<td>9</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-9</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

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

$$x := n; y := 0; z := 0;$$

$$\textbf{while } z < n \textbf{ do}$$

$$x := x + 5 \times z;$$

$$y := y - 5 \times z;$$

$$z := z + 1;$$

$$\textbf{od}$$

Give an inductive invariant and a variant for \texttt{while} loop in $p$ and prove the validity of the total correctness triple $[n \geq 1] p [x + y = n]$.

(9 points)
3.b) Let $B$ be a non-trivial post-condition; that is, $B$ is not equivalent to true or false.
Let $A_1$ denote $\text{wlp}(\text{abort}, B)$ and $A_2$ denote $\text{wlp}(\text{skip}, B)$.

Is $A_1$ weaker than $A_2$ or is $A_2$ weaker than $A_1$? Justify your answer! (3 points)
3.c) Let $p$ be the following IMP program:

\[
\text{if } x \neq y \text{ then } x := 3 \ast y; y := 3 \ast x \text{ else } y := 2 \ast x; x := 2 \ast y
\]

where $x, y$ are integer-valued program variables.

(a) Give a state $\sigma_1$ such that $\sigma_1 \models \{x < y\} p \{x > y\}$.

(b) Give a state $\sigma_2$ such that $\sigma_2 \nvdash \{x < y\} p \{x > y\}$.

Justify your answers.  

(3 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 $M_1$:**

```
s0: {a}
t1: {d}
t2: {c}
t3: {c}
t4: {b}
t5: {b}
t6: {d}
```

**Kripke structure $M_2$:**

```
s0: {a}
s1: {b}
s2: {c}
s3: {d}
```

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 $M_1$.

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 $M_2$.

(4 points)
4.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>$X(\neg a \land c)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E[(b \land c) U (a)]$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(b \land \neg c) U (a)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$EX(\neg b \land c)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$EGc$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(5 points)
4.c) Prove that the following LTL-formulas are tautologies, i.e., they hold for every Kripke structure $M$ and every path $\pi$ in $M$, or find a Kripke structure $M$ and path $\pi$ in $M$, for which the formula does not hold and justify your answer.

i. $G((Fb) \land \neg a) \rightarrow G\neg aUb$.

ii. $G\neg aUb \rightarrow G((Fb) \land \neg a)$.

(6 points)