1	2	3	4	Σ	Grade

6.0/4.0 VU Formale Methoden der Informatik

185.291 **June 20, 2023**

Kennz. (study id)	Matrikelnummer (student id)	Nachname (surname)	Vorname (first name)

Block 1.) Assume for this exercise that all strings are over an alphabet $\{0, 1\}$ and **contain at least one symbol**. For a string $I = b_1 b_2 \dots b_n$ we define the *bit-flipped string* $\mathit{flip}(I)$ as the bitwise complement of I, i.e., $\mathit{flip}(I) = c_1 c_2 \dots c_n$ where $c_i = 1 - b_1$. Example: for $I = 00110$ we get $\mathit{flip}(I) = 11001$. Recall that **co-HALTING** is the complement of the **HALTING** problem; consider the following variant thereof.

> **BITFLIP-HALTING**
>
> INSTANCE: A program Π that takes a string as input, and a string I.
>
> QUESTION: Does Π halt on I, but not halt on the bit-flipped string $\mathit{flip}(I)$?

1.a) The following function g provides a computable many-one reduction from the problem **co-HALTING** to **BITFLIP-HALTING**: $g((\Pi, I)) = (\Pi_1, I_1)$, where $I_1 = \mathit{flip}(I)$ and Π_1 is given as follows:

```
Π₁(string S){
    if(S = flip(I)){
        return;
    }
    Π(S);
    return;
}
```

Show the correctness of the reduction, i.e., show that (Π, I) is a positive instance of **co-HALTING** if and only if $g((\Pi, I))$ is a positive instance of **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 $\mathcal{T}_{cons}^{+}$-validity of the formula

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

The tasks are as follows:

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

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

is $\mathcal{T}_{cons}^{+}$-valid.

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

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

is $\mathcal{T}_{cons}^{+}$-valid. Use the semantic argument method to prove that the formula

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

is $\mathcal{T}_{cons}^{+}$-valid.

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

- $\forall u \forall v\ \mathit{car}(\mathit{cons}(u, v)) \doteq u$ (left projection)
- $\forall u \forall v\ \mathit{cdr}(\mathit{cons}(u, v)) \doteq v$ (right projection)
- $\forall u \left((\neg \mathit{atom}(u)) \rightarrow \mathit{cons}(\mathit{car}(u), \mathit{cdr}(u)) \doteq u \right)$ (construction)
- $\forall u \forall v\ \neg \mathit{atom}(\mathit{cons}(u, v))$ (atom)
- $\forall u \forall v \left(\mathit{atom}(u) \rightarrow \mathit{concat}(u, v) \doteq \mathit{cons}(u, v) \right)$ (concat atom)
- $\forall u \forall v \forall w\ \mathit{concat}(\mathit{cons}(u, v), w) \doteq \mathit{cons}(u, \mathit{concat}(v, w))$ (concat list)
- $\forall u \left(\mathit{atom}(u) \rightarrow \mathit{flat}(u) \right)$ (flat atom)
- $\forall u \forall v \left(\mathit{flat}(\mathit{cons}(u, v)) \leftrightarrow (\mathit{atom}(u) \land \mathit{flat}(v)) \right)$ (flat list)

(12 points)

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!

```
2 0
-1 -2 4 0
-4 5 0
-2 -4 6 0
-3 -6 7 0
-7 9 0
-5 -6 -7 -9 0
```

(3 points)

(15 points)

Block 3.)

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

$$
\begin{array}{l}
x := n; y := 0; z := 0; \\
\textbf{while } z < n \textbf{ do} \\
\quad x := x + 5 * z; \\
\quad y := y - 5 * z; \\
\quad z := z + 1; \\
\textbf{od}
\end{array}
$$

Give an inductive invariant and a variant for **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 $wlp(\textbf{abort}, B)$ and A_2 denote $wlp(\textbf{skip}, B)$.

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:

$$\textbf{if } x \neq y \quad \textbf{then} \quad x := 3 * y; y := 3 * x \quad \textbf{else} \quad y := 2 * x; x := 2 * y$$

where x, y are integer-valued program variables.

(a) Give a state σ_1 such that $\sigma_1 \models \{x < y\}p\{x > y\}$.

(b) Give a state σ_2 such that $\sigma_2 \not\models \{x < y\}p\{x > y\}$.

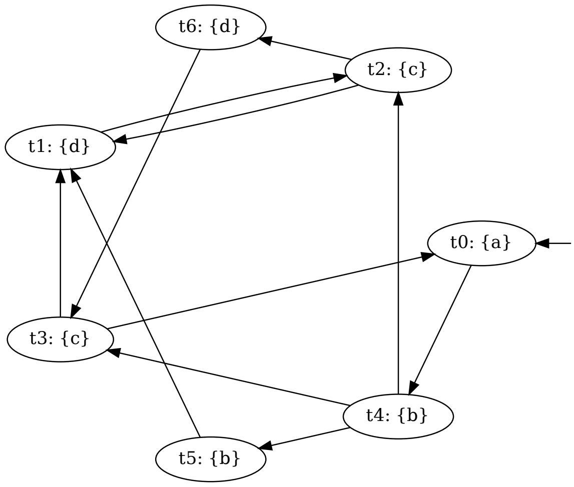
Justify your answers.

(3 points)

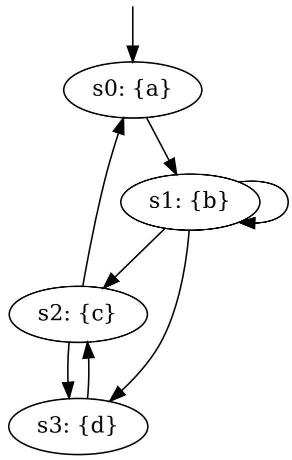
Block 4.)

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:



Kripke structure M_2:

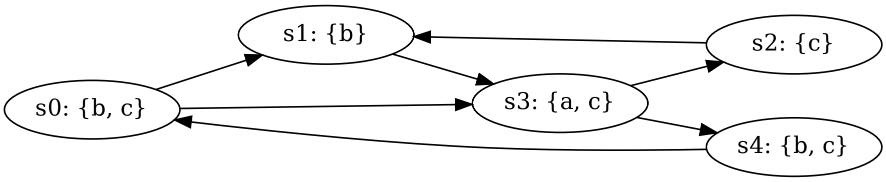


i. Check whether M_2 simulates M_1, i.e., provide a simulation relation that witnesses $M_1 \preceq M_2$, or briefly explain why M_2 does not simulate M_1.

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 φ,

i. indicate whether the formula is in CTL, LTL, and/or CTL*, and
ii. list the states s_i on which the formula φ holds; i.e. for which states s_i do we have $M, s_i \models \varphi$?
(If φ is a path formula, list the states s_i such that $M, s_i \models \mathbf{A}\varphi$.)

φ	CTL	LTL	CTL*	States s_i
$\mathbf{X}(\neg a \wedge c)$	☐	☐	☐	
$\mathbf{E}[(b \wedge c)\ \mathbf{U}\ (a)]$	☐	☐	☐	
$((b \wedge \neg c)\ \mathbf{U}\ (a))$	☐	☐	☐	
$\mathbf{EX}(\neg b \wedge c)$	☐	☐	☐	
$\mathbf{EG}c$	☐	☐	☐	

(5 points)

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

i. $\mathbf{G}\,((\mathbf{F}\,b) \wedge \neg a) \to \mathbf{G}\,(\neg a\,\mathbf{U}\,b)$.

ii. $\mathbf{G}\,(\neg a\,\mathbf{U}\,b) \to \mathbf{G}\,((\mathbf{F}\,b) \wedge \neg a)$.

(6 points)

Grading scheme: 0–29 nicht genügend, 30–35 genügend, 36–41 befriedigend, 42–47 gut, 48–60 sehr gut