1	2	3	4	Σ	Grade

6.0/4.0 VU Formale Methoden der Informatik

185.291 **February 26, 2021**

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

1.) Recall the **HALTING** problem which takes a program and a string as input, and consider the following variant thereof:

> **HALTING-X**
>
> INSTANCE: Two program Π_1, Π_2 that take a string as input.
>
> QUESTION: Does there exist at least one input string I such that both Π_1 and Π_2 halt on I.

(a) The following function f provides a polynomial-time many-one reduction from **HALTING** to **HALTING-X**: for a program Π and a string I let $f(\Pi, I) = (\Pi_1, \Pi_2)$ with

$$\begin{aligned}\Pi_1(\texttt{string } S) &= \texttt{return;}\\ \Pi_2(\texttt{string } S) &= \texttt{if } (S = I)\ \{\texttt{call } \Pi(S)\}\ \texttt{else}\ \{\texttt{while(true)}\{\}\};\ \texttt{return;}\end{aligned}$$

Show that (Π, I) is a yes-instance of **HALTING** $\Longleftrightarrow$ (Π_1, Π_2) is a yes-instance of **HALTING-X**.

(9 points)

(b) Tick the correct statements (for ticking a correct statement a certain number of points is given; ticking an incorrect statement results in a substraction of the same amount; you cannot go below 0 points):

- Since **HALTING** is decidable, our reduction from (a) shows that **HALTING-X** is decidable.
- Since **HALTING** is undecidable, our reduction from (a) shows that **HALTING-X** is undecidable.
- Since **HALTING** is semi-decidable, our reduction from (a) shows that **HALTING-X** is semi-decidable.
- Since **HALTING** is not semi-decidable, our reduction from (a) shows that **HALTING-X** is not semi-decidable.
- A reduction from **HALTING-X** to **HALTING** would show that **HALTING-X** is semi-decidable.
- A reduction from **HALTING-X** to **HALTING** would show that **HALTING-X** is undecidable.

(6 points)

2.) (a) Consider the clauses $C_1, \ldots, C_5$ in `dimacs` format (in this order, shown in the box) which are given as input to a SAT solver Apply CDCL to solve the CNF using the convention that if a variable is assigned as a decision, then it is assigned 'false'. Further, select variable `2` as the first decision variable that is assigned.

```
1 0
-1 10 0
-1 2 3 0
-3 -4 -10 0
-3 4 -10 0
```

- Each time when a conflict occurs and after backtracking, draw the implication graph and indicate all UIPs and mark the first UIP. For the first UIP, indicate its asserting conflict clause.
- Is the given CNF satisfiable, unsatisfiable, or valid? Justify your answer.

(4 points)

(b) Consider the theory $\mathcal{T}_A$ of arrays and the following formula

$$\varphi: \quad i_1 \neq j \;\vee\; a[j] \neq v_1 \;\vee\; i_1 \doteq i_2 \;\vee\; a\langle i_1 \triangleleft v_1\rangle\langle i_2 \triangleleft v_2\rangle[j] \doteq a[j] \;\; .$$

If φ is $\mathcal{T}_A$-valid, then provide a proof in the semantic argument method (similarly to the proofs in the lecture and on the extra sheets). If φ is not $\mathcal{T}_A$-valid, then provide a counter-example.

Besides the equality axioms reflexivity, symmetry and transitivity, you have the following ones for arrays.

- $\forall a, i, j \; \big(i \doteq j \;\rightarrow\; a[i] \doteq a[j]\big)$ (array congruence)
- $\forall a, v, i, j \; \big(i \doteq j \;\rightarrow\; a\langle i \triangleleft v\rangle[j] \doteq v\big)$ (read-over-write 1)
- $\forall a, v, i, j \; \big(i \neq j \;\rightarrow\; a\langle i \triangleleft v\rangle[j] \doteq a[j]\big)$ (read-over-write 2)

Please be precise. In a proof indicate exactly why proof lines follow from some other(s) and name the used rule. If you use derived rules you have to prove them. **(11 points)**

3.) (a) Let p be the following program:

$$
\begin{array}{l}
x := 1; y := 1; z := 0; \\
\textbf{while } x < n \textbf{ do} \\
\quad x := x + 1; \\
\quad z := z + 6; \\
\quad y := y + z \\
\textbf{od}
\end{array}
$$

Give a loop invariant and variant for the **while** loop in p and prove the validity of the total correctness triple $[n > 1]\ p\ [2 * y = n * z + 2]$.

Hint: Make sure that your invariant expresses equalities among y, z, x, as well as equalities among z, x.

(10 points)

(b) Provide a non-trivial pre-condition A and a non-trivial post-condition B, such that the total correctness triple $[A]\ p\ [B]$ is valid. Trivial means equivalent to `true` or `false`, so your precondition A and postcondition B should not be equivalent to `true` or `false`. The program p is given below.

Program p:

$$\textbf{if } x \neq 0 \textbf{ then skip else abort}$$

(2 points)

(c) Consider the following partial correctness triple:

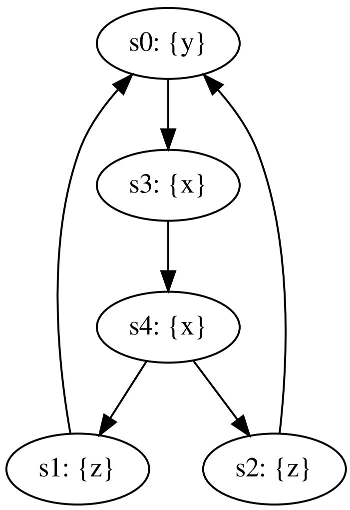
$$\{x = y\}\ x := y + 1; y := x + 1\ \{x = y - 1\}$$

Is the above Hoare triple valid? If so, give a formal proof. Otherwise, give a counterexample.

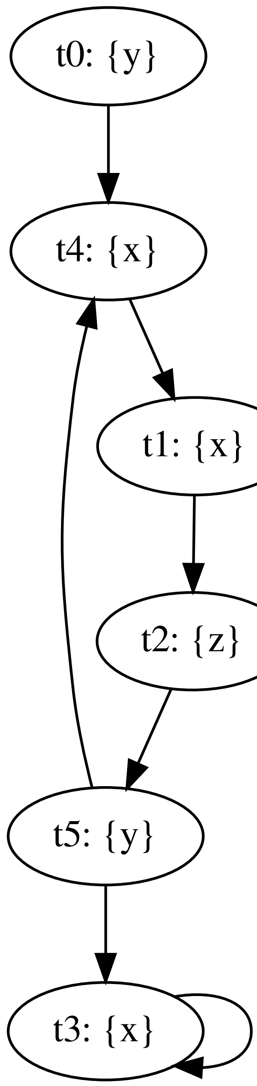
(3 points)

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:

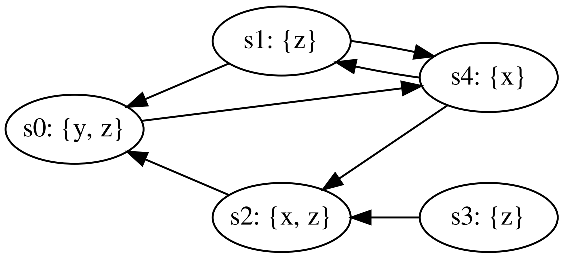


Kripke structure M_2:



(4 points)

(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
$x\ \mathbf{U}\ y$	☐	☐	☐	
$\mathbf{F}(x \wedge z)$	☐	☐	☐	
$\mathbf{EF}(y)$	☐	☐	☐	
$\mathbf{EG}(z)$	☐	☐	☐	
$\mathbf{AX}(z)$	☐	☐	☐	

(5 points)

(c) **LTL tautologies**

Prove that the following 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. $\mathbf{F}(y \wedge \mathbf{G}x) \Rightarrow (y \wedge ((\mathbf{X}y) \ \mathbf{U} \ (\mathbf{G}x)))$

ii. $(y \wedge ((\mathbf{X}y) \ \mathbf{U} \ (\mathbf{G}x))) \Rightarrow \mathbf{F}(y \wedge \mathbf{G}x)$

(6 points)

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