1	2	3	4	Σ	Grade

6.0/4.0 VU Formale Methoden der Informatik

185.291 **December 13, 2022**

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

Block 1.)

Consider the following problem.

> **CHAIN-HALTING**
>
> INSTANCE: Two programs Π_1, Π_2, that take a string as input and output a string, and a string I.
>
> QUESTION: Does $\Pi_2(\Pi_1(I))$ halt or $\Pi_1(\Pi_2(I))$ halt (or both), i.e., does one program halt when we use as input the output of the other program on input I?

1.a) The following function f provides a polynomial-time many-one reduction from the **HALTING** problem to **CHAIN-HALTING**: for a program Π and a string I, let $f((\Pi, I)) = (\Pi_1, \Pi_2, I')$ with $I' = I$, $\Pi_2 = \Pi$, and Π_1 given as follows:

```
Π1(string S){
    return S;
}
```

(You can assume that the program Π that is part the instances (Π, I) of **HALTING** also takes a string as input and outputs a string.)

Show the correctness of the reduction, i.e.:

(Π, I) is a yes-instance of **HALTING** $\iff$ $f((\Pi, I))$ is a yes-instance of **CHAIN-HALTING**.

(10 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.

true	**false**	
○	○	The correctness of the reduction in (a) shows that **CHAIN-HALTING** is undecidable.
○	○	The correctness of the reduction in (a) shows that **CHAIN-HALTING** is semi-decidable.
○	○	The correctness of the reduction in (a) shows that the complement of **CHAIN-HALTING** is decidable.
○	○	If we would have a decision procedure for **CHAIN-HALTING**, we can solve **HALTING** using our reduction from (a).
○	○	If we would have a decision procedure for **HALTING**, we can solve **CHAIN-HALTING** using our reduction from (a).

(5 points)

Block 2.)

2.a) Use Ackermann's reduction and translate

$$A(A(x)) \doteq A(B(x)) \rightarrow B(A(B(x))) \doteq y \vee C(x, y) \doteq C(A(x), B(x))$$

to a *satisfiability-equivalent* E-formula φ^E. A, B, and C are function symbols, x and y are variables. **(4 points)**

2.b) Consider the function M, defined as follows.

Algorithm 1: The function M

Input: x, y, two *positive* integers
Output: The computed positive integer value for x, y

if $x == 1$ **then**
  return $2y$;
else if $y == 1$ **then**
  return x;
else return $\mathrm{M}(x-1, \mathrm{M}(x, y-1))$;

Let $\mathbb{N}$ denote the natural numbers *without* 0. Use well-founded induction to show

$$\forall x \, \forall y \, \big((x \in \mathbb{N} \wedge y \in \mathbb{N}) \rightarrow \mathrm{M}(x, y) \geq 2y\big).$$

(11 points)

Block 3.)

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

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

Give a loop invariant and variant for the **while** loop in p and use them to formally prove the validity of the total correctness triple $[x = n \wedge n > 2]\ p\ [z = n^2]$.

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

(10 points)

3.b) Let p be the following IMP program, containing the integer-valued program variable x:

$$\textbf{while } x \geq 0 \textbf{ do } x := 1$$

Which of the following Hoare triples are correct? Provide short justifications for your answers (no formal proofs are required).

(i) $\{x = 0\} \quad p \; \{x = 1\}$

(ii) $[x = 0] \quad p \; [x = 1]$

(iii) $[x \leq 0] \quad p \; [x = 1]$

(iv) $\{x = -1\} \; p \; \{x = 1\}$

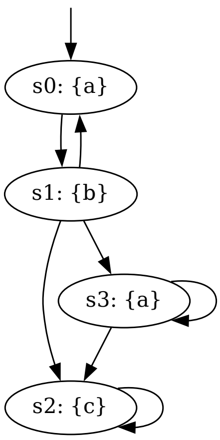
(v) $[false] \quad p \; [x = 1]$

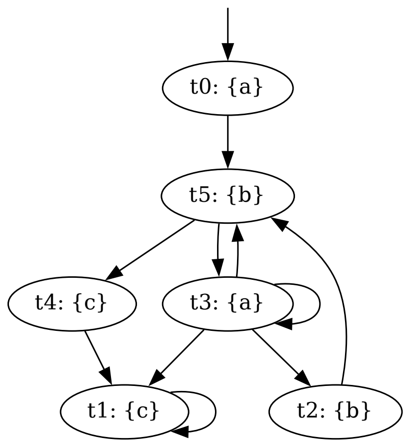
(5 points)

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:**



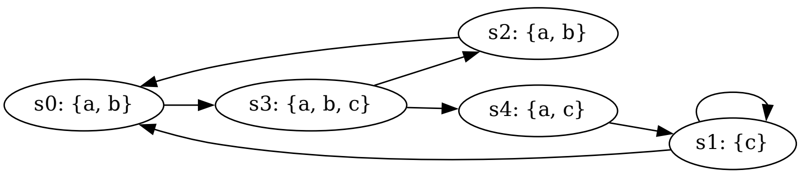


i. Check whether M_1 and M_2 are bisimilar. If they are bisimilar, provide a bisimulation relation that witnesses $M_1 \equiv M_2$. If they are not bisimilar, provide a CTL* formula φ that holds in exactly one of the structures, i.e., either $M_1 \models \varphi$ and $M_2 \not\models \varphi$, or $M_1 \not\models \varphi$ and $M_2 \models \varphi$. Indicate clearly which of the two structures satisfies the formula.

ii. 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.

(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
$\mathbf{EG}(c)$	☐	☐	☐	
$\mathbf{E}(c\ \mathbf{U}\ \mathbf{G}b)$	☐	☐	☐	
$\mathbf{E}(a\ \mathbf{U}\ b)$	☐	☐	☐	
$\mathbf{G}(c)$	☐	☐	☐	
$\mathbf{F}(a \wedge b)$	☐	☐	☐	

(5 points)

4.c) An LTL formula is a *tautology* if it holds for every Kripke structure M and every path π in M. For each of the following formulas, prove that it is a tautology, or find a Kripke structure M and path π in M for which the formula does not hold and justify your answer.

i. $((\mathbf{G}a\ \mathbf{U}\ \mathbf{G}b) \land \neg b) \Rightarrow \mathbf{FG}(a \land b)$

ii. $\mathbf{FG}(a \land b) \Rightarrow ((\mathbf{G}a\ \mathbf{U}\ \mathbf{G}b) \land \neg b)$

(6 points)

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