1	2	3	4	Σ

6.0/4.0 VU Formale Methoden der Informatik (185.291)
25 Januar 2018

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

1.) (a) Consider the following decision problem:

> **AT-LEAST-ONE-HALTS (ALOH)**
>
> INSTANCE: A tuple (Π_1, Π_2, I), where Π_1, Π_2 are programs that take a string as input, and I is a string.
>
> QUESTION: Does either $\Pi_1(I)$ halt, $\Pi_2(I)$ halt, or do both halt?

By providing a suitable reduction from the **HALTING** problem, prove that **AT-LEAST-ONE-HALTS** is undecidable.

(10 points)

(b) Is **AT-LEAST-ONE-HALTS** semi-decidable? Explain your answer.

(5 points)

2.) (a) Show that $a[i] \doteq e \rightarrow a\langle i \triangleleft e\rangle \doteq a$ is $\mathcal{T}_A^{=}$-valid using the semantic argument method from the lecture. Besides the equality axioms, you have the following ones for the arrays.

i. $\forall a, i, j\ \big(i \doteq j \rightarrow a[i] \doteq a[j]\big)$ (array congruence)

ii. $\forall a, v, i, j\ \big(i \doteq j \rightarrow a\langle i \triangleleft v\rangle[j] \doteq v\big)$ (read-over-write 1)

iii. $\forall a, v, i, j\ \big(i \not\doteq j \rightarrow a\langle i \triangleleft v\rangle[j] \doteq a[j]\big)$ (read-over-write 2)

iv. $\forall a, b\, \big(\forall j\ \big(a[j] \doteq b[j]\big) \leftrightarrow a \doteq b\big)$ (extensionality)

Besides the axioms, you are allowed to use *modus ponens* and *modus tollens*.

(12 points)

(b) Let $f(x_1, x_2) = x_1 \oplus x_2$ and $f(x_1, \ldots, x_{n+1}) = f(x_1, \ldots, x_n) \oplus x_{n+1}$ for $n > 2$.
(i) What is the number of clauses in a satisfiability-equivalent CNF version of $f(x_1, \ldots, x_n)$.
(ii) What is the number of clauses in a logically equivalent CNF version of $f(x_1, \ldots, x_n)$.
Explain and justify your answers in detail. **(3 points)**

3.) (a) Consider the program p:

$$
\begin{array}{l}
r := 0; \\
i := 0; \\
s := 1; \\
\textbf{while } i < n \textbf{ do} \\
\quad r := r + s; \\
\quad s := s + 2; \\
\quad i := i + 1 \\
\textbf{od}
\end{array}
$$

Let $p' = \textbf{while } i < n \textbf{ do} \ldots \textbf{od}$ be the while loop of the program p. Give an inductive invariant for p', such that the Hoare triple $\{true\}\ p'\ \{true\}$ is valid with respect to partial correctness. **(2 points)**

(b) Let p be the program from exercise 3a. Show that the Hoare triple $\{n \geq 0\}\ p\ \{r = n*n\}$ is valid with respect to partial correctness. The program p is given again for your convenience:

$$
\begin{array}{l}
r := 0; \\
i := 0; \\
s := 1; \\
\textbf{while } i < n \textbf{ do} \\
\quad r := r + s; \\
\quad s := s + 2; \\
\quad i := i + 1 \\
\textbf{od}
\end{array}
$$

(10 points)

(c) Is the following Hoare triple valid with respect to total correctness? If yes, prove its validity using the Hoare calculus. Otherwise, provide a counterexample, that is, a state that does not satisfy the correctness assertion.

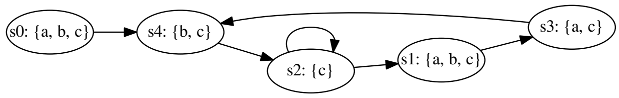
$$[n \geq 0] \textbf{ if } n = 0 \textbf{ then abort else } m := n \ [m = n]$$

(3 points)

4.) (a) Show that the simulation relation for Kripke structures is transitive, i.e., if $A \leq B$ and $B \leq C$ for some Kripke structures A, B, C, then $A \leq C$.

(5 points)

(b) Consider the following Kripke structure M:



For each of the following formulae φ,

i. check the respective box if the formula is in CTL, and
ii. list the states s_i on which the formula φ holds; i.e. for which states s_i do we have $M, s_i \models \varphi$?

Recall that for an LTL formula φ it holds that $M, s \models \varphi \iff M, s \models \mathbf{A}\varphi$.

φ	CTL	States s_i
$\mathbf{X}(a)$	☐	
$((b)\ \mathbf{U}\ (a))$	☐	
$\mathbf{AF}(b \wedge c)$	☐	
$\mathbf{A}[(a)\ \mathbf{U}\ (b)]$	☐	
$\mathbf{EF}(c)$	☐	

(5 points)

(c) **LTL tautologies**

Prove or disprove that the following formulas are tautologies, i.e., they hold for every Kripke structure M and every path π:

i.
$$(\mathbf{GF}p) \wedge (\mathbf{GF}q) \Rightarrow \mathbf{G}(p\ \mathbf{U}\ q)$$

ii.
$$((\mathbf{G}\neg p)\ \mathbf{U}\ p) \wedge \neg p \Rightarrow (\mathbf{G}q) \vee (\neg q\ \mathbf{U}\ r)$$

(5 points)

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