1	2	3	4	Σ	Grade

6.0/4.0 VU Formale Methoden der Informatik

185.291 October 22, 2021 Variant B

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:

> **NON-TRIVIAL-HALTING (NTH)**
>
> INSTANCE: A program Π' that takes a string as input.
>
> QUESTION: Is it true, that there exist input strings I_1, I_2, such that Π' does not halt on I_1 and Π' halts on I_2.

(a) The following function f provides a polynomial-time many-one reduction from the **co-HALTING** problem (the *complement* of **HALTING**) to **NTH**: for a program Π and a string I, let $f(\Pi, I) = (\Pi')$ with

$$\Pi'(\texttt{string } S) \quad = \quad \texttt{if } (S \mathrel{!=} I) \ \{\texttt{print } S;\} \ \texttt{else} \ \{\texttt{call } \Pi(S);\} \ \texttt{return;}$$

Show that (Π, I) is a yes-instance of **co-HALTING** $\Longleftrightarrow$ (Π') is a yes-instance of **NTH**.

(9 points)

(b) Recall that **co-HALTING** is not even semi-decidable and suppose that our reduction from **co-HALTING** to **NTH** is correct. Tick the correct statements that can be concluded from these observations (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):

- **NTH** is undecidable.
- **NTH** is semi-decidable.
- **NTH** is decidable.
- Suppose we have a decision procedure for **NTH**; then we would have a decision procedure for **co-HALTING**.
- Since **HALTING** is semi-decidable, our reduction also shows that the complement of **NTH** is semi-decidable.
- A problem or its complement is semi-decidable, thus the complement of **NTH** is semi-decidable.

(6 points)

2.) (a) Consider the following clause set $\hat{\delta}(\varphi)$ which has been derived from an (unknown) formula φ by Tseitin translation (atoms have not been labeled).

$$\begin{array}{rlrlrl}
C_1: & \ell_1 \vee \neg x \vee \neg y & C_2: & \neg \ell_1 \vee x & C_3: & \neg \ell_1 \vee y \\
C_4: & \neg \ell_2 \vee \neg y \vee z & C_5: & \ell_2 \vee y & C_6: & \ell_2 \vee \neg z \\
C_7: & \neg \ell_3 \vee \neg \ell_1 \vee z & C_8: & \ell_3 \vee \ell_1 & C_9: & \ell_3 \vee \neg z \\
C_{10}: & \neg \ell_4 \vee \neg x \vee \ell_2 & C_{11}: & \ell_4 \vee x & C_{12}: & \ell_4 \vee \neg \ell_2 \\
C_{13}: & \neg \ell_5 \vee \neg \ell_3 \vee \ell_4 & C_{14}: & \ell_5 \vee \ell_3 & C_{15}: & \ell_5 \vee \neg \ell_4
\end{array}$$

(i) Reconstruct φ from $\hat{\delta}(\varphi)$.

(ii) Start from $\hat{\delta}(\varphi)$ and extend it by a single nonempty clause C in such a way that φ is valid iff $\hat{\delta}(\varphi) \wedge C$ is unsatisfiable.

(iii) Prove the validity of φ by resolution (no additional translation to normal form is allowed!).

(4 points)

(b) Use Ackermann's reduction and translate

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

to a validity-equivalent E-formula φ^E. A, B, and C are function symbols, x and y are variables. **(3 points)**

(c) Let φ^{uf} be an equality formula containing uninterpreted functions. Let $FC^E(\varphi^{uf})$ and $flat^E(\varphi^{uf})$ be obtained by Ackermann's reduction. Prove the following.

$$\varphi^{uf} \text{ is satisfiable} \quad \text{iff} \quad FC^E(\varphi^{uf}) \wedge flat^E(\varphi^{uf}) \text{ is satisfiable.}$$

Hint: FC^E is the same for φ^{uf} and $\neg\varphi^{uf}$. **(8 points)**

(15 points)

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

while $i > n$ **do**
  $x := x + 7;$
  $i := i - 1$
od

Which of the following program assertions are inductive loop invariants of p?

- I_1 : $i \leq n$
- I_2 : $i > n$
- I_3 : $x + 7 * i = 0$

Give formal details justifying your answer. That is, if an assertion is an inductive loop invariant, provide a formal proof of it based on Hoare logic. If an assertion is not an inductive loop invariant, give a counterexample.

Note: You need to use the definition of an assertion being inductive invariant.

(11 points)

(b) Let p be the following IMP program loop, containing the integer-valued program variable i:

while $i \leq 3$ **do**
  if $i > 4$ **then abort**
  else $i := i + 1$
od

Which of the following Hoare triple are valid total correctness assertions?

- $[false]\ p\ [i = 4]$
- $[i = 3]\ p\ [i = 4]$
- $[true]\ p\ [i = 4]$

(4 points)

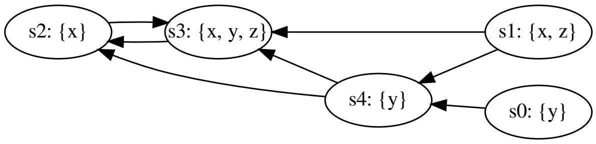
4.) (a) For two LTL formulas φ_1 and φ_2, the release operator $(\varphi_1 \mathbf{R} \varphi_2)$ requires φ_2 to remain true until and including the point where φ_1 first becomes true, but does not require that φ_1 ever becomes true.

i. Give a formal definition of the semantics of the release operator, i.e., provide a first order formula defining $M, \pi \models \varphi_1 \mathbf{R} \varphi_2$ for a Kripke structure M and a path π of M.

ii. Express $\mathbf{R}$ in terms of the LTL operators $\mathbf{U}, \mathbf{X}, \mathbf{F}, \mathbf{G}$ defined in the lecture.

(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
$\mathbf{AX}(x)$	☐	☐	☐	
$\mathbf{F}(z)$	☐	☐	☐	
$\mathbf{E}[(z)\ \mathbf{U}\ (x)]$	☐	☐	☐	
$\mathbf{G}(z)$	☐	☐	☐	
$\mathbf{EG}(y)$	☐	☐	☐	

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

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

(6 points)

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