1	2	3	4	Σ	Grade

6.0/4.0 VU Formale Methoden der Informatik
185.291 January, 23 2024

Kennzahl (study id)	Matrikelnummer (student id)	Familienname (family name)	Vorname (first name)	Gruppe (version)
				A

1.) Recall from the lecture the **HALTING** problem:

> **HALTING**
>
> INSTANCE: A non-empty program Π that takes a string as input, a string I.
>
> QUESTION: Does Π terminate on I.

(Remark: For this exercise, we assume that if (Π, I) is an instance of **HALTING**, then Π is not empty, i.e., Π contains at least one computation step. This assumption does not affect the decidability of the problem.)

Consider now the following decision problem:

> **DIFF-10**
>
> INSTANCE: A program Π that is guaranteed to terminate, and takes an integer as input and returns an integer as output.
>
> QUESTION: Do there exist integers n_1, n_2, such that $\Pi(n_1) = \Pi(n_2) - 10$?

(a) Let Π_{int} be the decision procedure that does the following:

- Π_{int} takes as input a program Π, a string I, and an integer n.
- Π_{int} emulates the first n steps of the run of Π on I. If Π terminates on I within n steps, then Π_{int} returns true. Otherwise, Π_{int} returns false.

The following describes a reduction from **HALTING** to **DIFF-10**. Given an arbitrary instance (Π, I) of **HALTING**, we construct an instance Π' of **DIFF-10** as follows:

```
Boolean Π′ (Int n)
if Π_int(Π, I, n) return 10; // Π and I are hard-coded in Π′
return 0;
```

Show the correctness of the reduction above, i.e., show that (Π, I) is a positive instance of **HALTING** $\Longleftrightarrow$ Π' is a positive instance of **DIFF-10**.

(9 points)

(b) Please answer the following questions and explain your answers:

- Is **DIFF-10** undecidable?
- Is **DIFF-10** semi-decidable?

(6 points)

2.) (a) Consider the function M.

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 M(x − 1, M(x, y − 1));
```

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

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

ii. Suppose $\mathtt{M}_{\mathtt{C}}$ is an implementation of M in the `C` programming language with x and y of type unsigned integers of size 32 bit (i.e., of type `uint32_t`). Is

$$\mathrm{M}(x', y') = \mathtt{M}_{\mathtt{C}}(x', y')$$

true for all integers x', y' satisfying $1 \leq x', y' \leq$ `UINT32_MAX`, where `UINT32_MAX` is the largest value for a variable of type `uint32_t`?

If so, then prove this fact. Otherwise provide a counterexample with an exact explanation of what is computed and what is happening.

(12 points)

(b) Let $f(x_1, x_2) = x_1 \leftrightarrow x_2$ and $f(x_1, \ldots, x_{n+1}) = f(x_1, \ldots, x_n) \leftrightarrow x_{n+1}$ for $n > 2$.

i. Apply Tseitin's translation to $f(x_1, x_2)$. What clauses do we get?

ii. What is the number of clauses in terms of n in a satisfiability-equivalent CNF version $f(x_1, \ldots, x_n)$ obtained by a traditional CNF translation. O-notaition is sufficient here.

iii. What is the exact number of clauses in terms of n in a logically equivalent CNF version of $f(x_1, \ldots, x_n)$ obtained by Tseitin's translation.

Explain and justify your answers in detail. **(3 points)**

3.)

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

$$
\begin{array}{l}
x := 0; y := 0; z := n; \\
\textbf{while } y < n \textbf{ do} \\
\quad x := x + 3 * y; \\
\quad y := y + 1; \\
\quad z := z - 3 * y + 3; \\
\textbf{od}
\end{array}
$$

Provide a loop inductive invariant and loop variant and use them to prove the total correctness of the Hoare triple:

$$[n > 0] \;\; p \;\; [x + z \geq y]$$

(9 points)

(b) Let x be an integer-valued. For each of the triples below, is there a state σ and non-trivial assertion A such that

(i) $\sigma \not\models [x > 0]$ **skip** $[A]$?

(ii) $\sigma \not\models [x > 0]$ **abort** $[A]$?

(iii) $\sigma \not\models [x > 0]$ $x := x + 1$ $[A]$?

In each of the cases above, if such a state σ and non-trivial assertion A exist, provide a concrete σ and A and justify your answers. Otherwise, explain why there exist no such state σ and assertion A.

A non-trivial assertion is an assertion that is not equivalent to `true` nor `false`. Recall that $\sigma \not\models [P]\ p\ [Q]$ means that σ does not satisfy the Hoare triple $[P]\ p\ [Q]$.

(6 points)

4.) (a) If there exists a simulation from Kripke structure M to Kripke structure M' we write $M \preceq M'$, and if there exists a bisimulation between M and M' we write $M \equiv M'$. Consider the following two statements. Either present a proof if the statement is valid or state a counterexample otherwise.

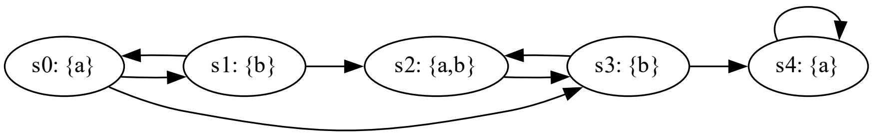
i) The relation $\preceq$ is transitive, i.e. for all Kripke structures K, L, M:

$$\text{If } K \preceq L \text{ and } L \preceq M \text{ then } K \preceq M.$$

ii) From $M \preceq M'$ and $M' \preceq M$ follows $M \equiv M'$ for all Kripke structures M and M'.

(6 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
$\mathbf{AX}b$	☐	☐	☐	
$\mathbf{E}[a\ \mathbf{U}\ (\mathbf{G}b)]$	☐	☐	☐	
$\mathbf{F}(\mathbf{G}a \vee \mathbf{G}b)$	☐	☐	☐	

(3 points)

(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. $\mathbf{G}(\mathbf{F}a \to a) \to a \ \mathbf{U}\ (\mathbf{G}\neg a)$

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

(6 points)