1	2	3	4	Σ	Grade

6.0/4.0 VU Formale Methoden der Informatik

185.291 **May 26, 2025**

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

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.

Consider now the following decision problem:

> **SYM**
>
> INSTANCE: Program Π that is guaranteed to terminate and takes two positive numbers as input and returns a positive number as output.
>
> QUESTION: Do there exist two different numbers n_1, n_2, such that $\Pi(n_1, n_2) = \Pi(n_2, n_1)$?

(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 **SYM**. Given an arbitrary instance (Π, I) of **HALTING**, we construct an instance (Π') of **SYM** as follows:

```
Boolean Π′ (Int i, j)
if Π_int(Π, I, i + j) return i + j; // Π and I are hard-coded
return i;
```

Show the correctness of the reduction above, i.e., show that (Π, I) is a positive instance of **HALTING** $\iff$ (Π') is a positive instance of **SYM**. **(9 points)**

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

- Assume **SYM** is semi-decidable; given the above reduction what can be said about semi-decidability of the complement of **SYM**, **co**$-$**SYM**?
- Now, show that **SYM** is semi-decidable.

(6 points)

2.) (a) Consider the clauses $C_0, \ldots, C_6$ in `dimacs` format (in this order from top to bottom, shown in the box) which are given as input to a SAT solver.

- Apply CDCL using the convention that if a variable is assigned as a decision, then it is assigned 'true'. Select variables as decisions in increasing order of their respective integer IDs in the `dimacs` format, starting with variable `1`. Recall that unit clauses require a special treatment.

- When the *first* conflict occurs, draw the complete implication graph, mark the first UIP, give the resolution derivation of the learned asserting clause that corresponds to the first UIP, and stop CDCL. You do not have to solve the formula!

```
2 0
-1 -2 4 0
-4 5 0
-2 -4 6 0
-3 -6 7 0
-7 9 0
-5 -6 -7 -9 0
```

(3 points)

(b) Use the sparse method to translate the following formula φ^E

$$\neg\big(a \not\doteq b \lor a \doteq c \lor ((a \doteq d \land e \not\doteq f \land g \not\doteq h) \lor g \not\doteq i \lor h \not\doteq j \lor (b \not\doteq c \land g \doteq i \land i \not\doteq j))\big)$$

into a propositional formula φ^p such that φ^E is E-satisfiable if and only if φ^p is satisfiable. Simplify your formula before you construct the propositional skeleton and the transitivity constraints. In the simplifications steps, indicate the simple contradictory cycles and the pure literals.

Present an E-model for φ^E in a formally correct way. **(12 points)**

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

$$
\begin{array}{l}
x := 0; y := -1; z := 1; \\
\textbf{while } x < n \textbf{ do} \\
\quad x := x + 1; \\
\quad y := y - 4 * x; \\
\quad z := z + 2 * x \\
\textbf{od}
\end{array}
$$

Give an inductive invariant for the loop in p and prove the validity of the partial correctness triple:

$$\{n > 0\} \;\; p \;\; \{y + z + n * (n + 1) = 0\}$$

(9 points)

(b) Given the following IMP program, containing the integer-valued program variables x, y:

$$
\begin{array}{l}
\textbf{if } x + y > 0 \textbf{ then} \\
\quad x := y * y - 2 * x \\
\textbf{else} \\
\quad \textbf{abort} \\
\textbf{end if}
\end{array}
$$

For each case below, provide *non-trivial* pre- and postconditions A, B such that:

(i) $\{A\} \ \ p \ \ \{B\}$ is not valid.

(ii) $\{A\} \ \ p \ \ \{B\}$ is valid, but $[A] \ \ p \ \ [B]$ is not valid.

(iii) $[A] \ \ p \ \ [B]$ is valid.

A non-trivial precondition/postcondition is a precondition/postcondition that is not equivalent to the always true or the always false formula. **(6 points)**

4.) (a) Let M_0, M_1 be Kripke structures. Suppose M_0 can be simulated by M_1, that is, there exists a simulation relation witnessing $M_0 \preceq M_1$.

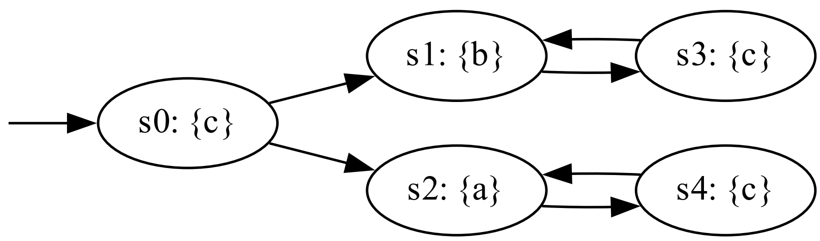
For the following statements, either prove that they are true for any such structures, or provide Kripke structues M_0, M_1, a simulation relation witnessing $M_0 \preceq M_1$, and briefly explain why they constitute a counter-example.

i If every state in M_0 is labeled with the atomic proposition a, then every state in M_1 labeled with the atomic proposition a.

ii If every state in M_1 is labeled with the atomic proposition a, then every state in M_0 labeled with the atomic proposition a.

(6 points)

(b) Consider the following Kripke structure M:



For each of the following formulae φ,

i. indicate whether the formula is in LTL, CTL, 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	LTL	CTL	CTL*	States s_i
$\mathbf{F}c$	☐	☐	☐	
$\mathbf{EG}(a \vee c)$	☐	☐	☐	
$[a\ \mathbf{U}\ b]$	☐	☐	☐	
$\mathbf{AXX}c$	☐	☐	☐	
$\mathbf{E}[c\ \mathbf{U}\ a]$	☐	☐	☐	

(5 points)

(c) Recall that a LTL formula φ is *satisfiable* if there exists a Kripke structure M and a path π in M such that $M, \pi \models \varphi$. Show that the following LTL formulas are satisfiable by providing an appropriate Kripke structure M, a path π in M, and briefly explaining why $M, \pi \models \varphi$ holds.

i $\mathbf{GF}a \land \neg\mathbf{FG}a$

ii $[(\mathbf{X}a)\ \mathbf{U}\ b] \land \neg a$

(4 points)

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