1	2	3	4	Σ

6.0/4.0 VU Formale Methoden der Informatik (185.291)

Dec 9, 2020

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

1.) An undirected graph (V, E) is called a *distance2-graph* if the there exists a vertex in V such that all vertices are reached via a path of length at most 2.

For example, the graph $(\{a_1, b_1, c_1, a_2, b_2, c_2, x\}, \{[a_1, a_2], [b_1, b_2], [c_1, c_2], [a_2, x], [b_2, x], [c_2, x]\})$ is a distance2-graph (since each vertex is reached from x via a path of length at most 2), while $(\{a, b, c, d\}, \{[a, b], [b, c], [c, d]\})$ is not.

Consider the following new problem **3COLD2** and recall the NP-complete problem **3COL** defined below:

> **3-COLORABILITY-DISTANCE2-GRAPH (3COLD2)**
>
> INSTANCE: A distance2-graph $G = (V, E)$.
>
> QUESTION: Does there exists a function μ from vertices in V to values in $\{0, 1, 2\}$ such that $\mu(v_1) \neq \mu(v_2)$ for any edge $[v_1, v_2] \in E$.

> **3-COLORABILITY (3COL)**
>
> INSTANCE: An undirected graph $G = (V, E)$.
>
> QUESTION: Does there exists a function μ from vertices in V to values in $\{0, 1, 2\}$ such that $\mu(v_1) \neq \mu(v_2)$ for any edge $[v_1, v_2] \in E$.

(a) The following function f provides a polynomial-time many-one reduction from **3COL** to **3COLD2**: for a directed graph $G = (\{v_1, \ldots, v_n\}, E)$, let

$$f(G) = \Big(\{v_1', v_1'', \ldots, v_n', v_n''\} \cup \{x\}, \\ \{[v_i', v_i''], [v_i'', x] \mid i = 1 \ldots n\} \cup \\ \{[v_i', v_j'] \mid [v_i, v_j] \in E\}\Big)$$

Show that G is a yes-instance of **3COL** $\iff$ $f(G)$ is a yes-instance of **3COLD2**.

(9 points)

(b) Tick the correct statements (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):

- Since **3COLD2** is a sub-problem of **3COL**, **3COLMG** must be NP-hard.
- Since **3COLD2** is a sub-problem of **3COL**, **3COLD2** must be contained in NP.
- Given that **3COL** is NP-hard, our reduction shows that **3COLD2** is also NP-hard.
- Given that **3COL** is NP-hard, our reduction shows that **3COLD2** is in NP.
- Given that **3COL** is in NP, our reduction shows that **3COLD2** is also in NP.
- Given that **3COL** is in NP, our reduction shows that **3COLD2** is NP-hard.

(6 points)

2.) (a) First define the concept of a theory and of a $\mathcal{T}$-interpretation. Then use them to define:

i. the $\mathcal{T}$-satisfiability of a formula;

ii. the $\mathcal{T}$-validity of a formula.

Additionally define the completeness of a theory $\mathcal{T}$. **(3 points)**

(b) Consider the function P, defined as follows.

Algorithm 1: The function P

Input: x, y, two non-negative integers

Output: The computed non-negative integer value for x, y

if $x == 0$ **then**
    return $y + 1$;
else if $y == 0$ **then**
    return $\mathrm{P}(x-1, 1)$;
else return $\mathrm{P}(x-1, \mathrm{P}(x, y-1))$;

Show, using well-founded induction, that

$$\forall x \, \forall y \, \big((x \in \mathbb{N}_0 \wedge y \in \mathbb{N}_0) \rightarrow \mathrm{P}(x, y) > y\big)$$

(10 points)

(c) Suppose $\texttt{P}_\texttt{C}$ is a correct implementation of P in the `C` programming language with x and y of type unsigned integers of size 64 bit (i.e., of type `uint64_t`). Is

$$\mathrm{P}(x', y') = \texttt{P}_\texttt{C}(x', y')$$

true for all integers x', y' satisfying $0 \leq x', y' \leq$ `UINT64_MAX`, where `UINT64_MAX` is the largest value for a variable of type `uint64_t`?

If so, then prove this fact. Otherwise provide a counterexample with an exact explanation of what is computed and what is happening. **(2 points)**

3.) (a) Let p be the following program:

$$
\begin{array}{l}
y := n; \\
\textbf{while } y > 0 \textbf{ do} \\
\quad x := x - 8 * y + 4; \\
\quad y := y - 1 \\
\textbf{od}
\end{array}
$$

Give a loop invariant for the **while** loop in p and prove the validity of the partial correctness triple $\{n > 0 \wedge x = 4 * n^2 + 2\}\ p\ \{x = 2\}$.

(9 points)

(b) Provide a non-trivial pre-condition A and a non-trivial post-condition B, such that the total correctness triple $\{A\}\ p\ \{B\}$ is valid. Trivial means equivalent to `true` or `false`, so your precondition A and postcondition B should not be equivalent to `true` or `false`. The program p is given below.

Program p:

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

(2 points)

(c) Explain why $wp(\texttt{abort}, B) = \texttt{false}$, whereas $wlp(\texttt{abort}, B) = \texttt{true}$ for any B.

Give a concise answer!

(2 points)

(d) Explain the role of t in the following verification condition which ensures the total correctness **while** loop:

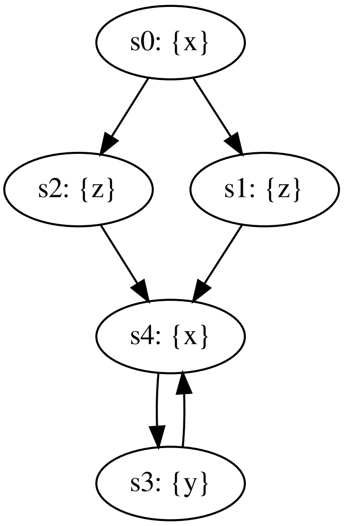
$$
\mathrm{VC}(\textbf{while } b \textbf{ do } p \textbf{ od}, B) = \begin{array}{l}
(I \wedge \neg b) \implies B \\
\bigwedge \\
(I \wedge b) \implies t \geq 0 \\
\bigwedge \\
(I \wedge b \wedge t = t_0) \implies \mathrm{wp}(p, I \wedge t < t_0) \\
\bigwedge \\
\mathrm{VC}(p, I \wedge t < t_0)
\end{array}
$$

Why do we need t_0?

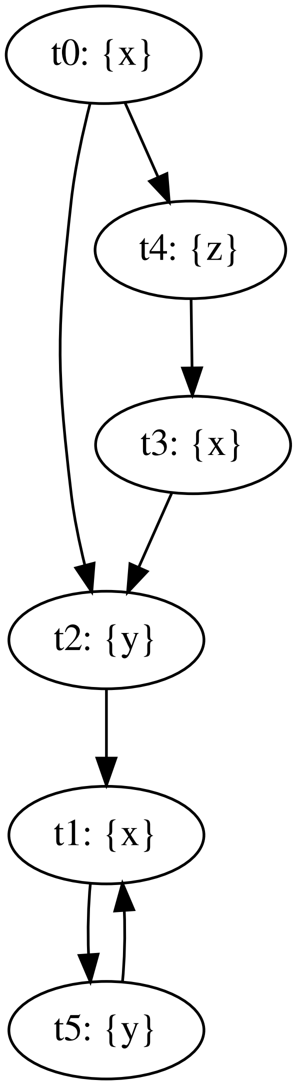
Give a concise answer! **(2 points)**

4.) (a) Provide a non-empty simulation relation H that witnesses $M_1 \leq M_2$, where M_1 and M_2 are shown below. The initial state of M_1 is s_0, the initial state of M_2 is t_0:

Kripke structure M_1:

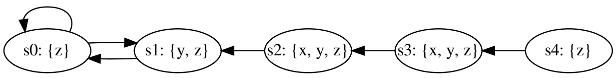


Kripke structure M_2:



(4 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, 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 $M, s_i \models \varphi$?

φ	CTL	LTL	CTL*	States s_i
$\mathbf{AX}(y)$	☐	☐	☐	
$\mathbf{EG}(z)$	☐	☐	☐	
$\mathbf{F}(y)$	☐	☐	☐	
$\mathbf{G}(x \wedge z)$	☐	☐	☐	
$\mathbf{E}[(z)\ \mathbf{U}\ (x)]$	☐	☐	☐	

(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. $((p\ \mathbf{U}\ \neg q)\ \mathbf{U}\ \neg r) \Leftrightarrow (p\ \mathbf{U}\ (\neg q\ \mathbf{U}\ \neg r))$

ii. $\mathbf{F}(p \wedge \mathbf{XG}p) \Rightarrow \mathbf{GF}p$

(6 points)

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