6.0/4.0 VU Formale Methoden der Informatik

185.291 SS 2012 7 December 2012

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

1.) Consider the following problem:

> **RUN-FOREVER-NO-INPUT (RFNO)**
>
> INSTANCE: A program Π such that Π takes no input.
>
> QUESTION: Does Π not terminate, i.e. does Π run forever?

By providing a reduction from an undecidable problem to **RFNO**, prove that **RFNO** is undecidable. Argue formally that your reduction is correct.

Hint: If a problem $\mathcal{P}$ is undecidable, then its complement $\bar{\mathcal{P}}$ is also undecidable.

(15 points)

2.) (a) Assume *fmiSAT* is a SAT solver that uses unit-propagation (BCP) to build an implication graph (IG). Decisions are done using the DLIS heuristics, and backtracking is done using dependency-directed backtracking.

i. Prove that the implication graph IG built by *fmiSAT* is acyclic at any time. **(6 points)**

ii. Show that in a conflict graph the first UIP is uniquely defined, i.e., there is exactly one node in the implication graph which is a first UIP. **(5 points)**

(b) Use Ackermann's reduction and translate

$$\varphi : \quad F(x_1) = F(a) \rightarrow G(F(a), a) = G(F(x_1), b) \wedge F(F(a)) \neq F(x_1)$$

to a validity-equivalent E-formula φ^E. **(4 points)**

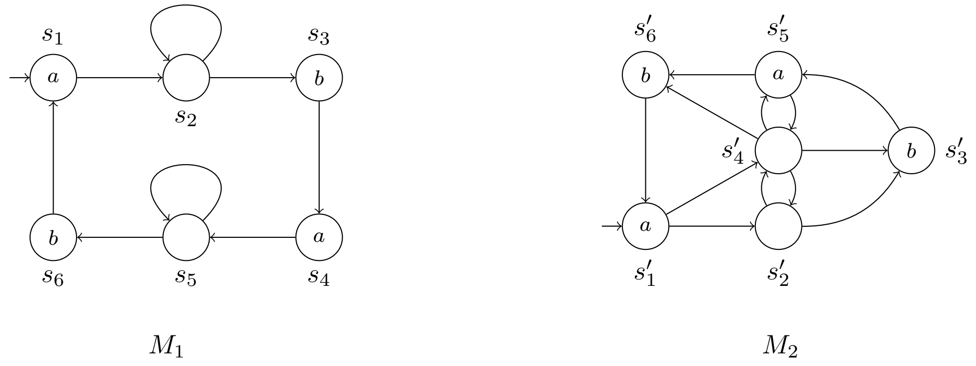
3.) (a) Determine the strongest postcondition of the weakest liberal precondition of an assignment statement, i.e., compute $\text{sp}(\text{wlp}(v := e, G), v := e)$. Why is it different from G?
Remember the following properties of wlp and sp:
$\text{wlp}(v := e, G) = G[v/e]$
$\text{sp}(F, v := e) = \exists v' \, (F[v/v'] \wedge v = e[v/v'])$ **(5 points)**

(b) Show that the following correctness assertion is totally correct. Depending on how you choose the variant, use one of the following annotation rules:
$\mathsf{while}\ e\ \mathsf{do} \cdots \mathsf{od} \mapsto \{\ Inv\ \}\mathsf{while}\ e\ \mathsf{do}\ \{\ Inv \wedge e \wedge t{=}t_0\ \} \cdots \{\ Inv \wedge 0{\leq}t{<}t_0\ \}\mathsf{od}\{\ Inv \wedge \neg e\ \}$
$\mathsf{while}\ e\ \mathsf{do} \cdots \mathsf{od} \mapsto \{\ Inv\ \}\mathsf{while}\ e\ \mathsf{do}\ \{\ Inv \wedge e \wedge t{=}t_0\ \} \cdots \{\ Inv \wedge (e \rightarrow 0{\leq}t{<}t_0)\ \}\mathsf{od}\{\ Inv \wedge \neg e\ \}$

$\{\ \mathbf{1\colon} x \geq 0\ \}$
$z := x;$
$y := 0;$
$\{\ \boldsymbol{Inv\colon x = y + z \wedge z \geq 0}\ \}$
while $z \neq 0$ do
  $y := y + 1;$
  $z := z - 1$
od
$\{\ \mathbf{2\colon} x = y\ \}$

(10 points)

4.) Simulation.



(a) Show that M_2 simulates M_1, i.e., that $M_1 \leq M_2$. **(6 points)**

(b) For Kripke structures M_1 and M_2 given above, provide an LTL formula φ such that $M_1 \models \varphi$ but $M_2 \not\models \varphi$. Justify why $M_1 \models \varphi$ and $M_2 \not\models \varphi$ hold, respectively. **(6 points)**

(c) Show that the following theorem for $\mathsf{ACTL}^\star$ does not hold for all $\mathsf{CTL}^\star$ formulas.

> **Theorem.**
>
> Let φ be an $\mathsf{ACTL}^\star$ formula and let M_1, M_2 be Kripke structures satisfying $M_1 \leq M_2$. Then, $M_2 \models \varphi$ implies $M_1 \models \varphi$.

Hint: Give a $\mathsf{CTL}^\star$ formula which contradicts the theorem for M_1 and M_2 and explain why this formula is a counterexample to the theorem. **(3 points)**