

Advanced Model Checking Summer term 2014

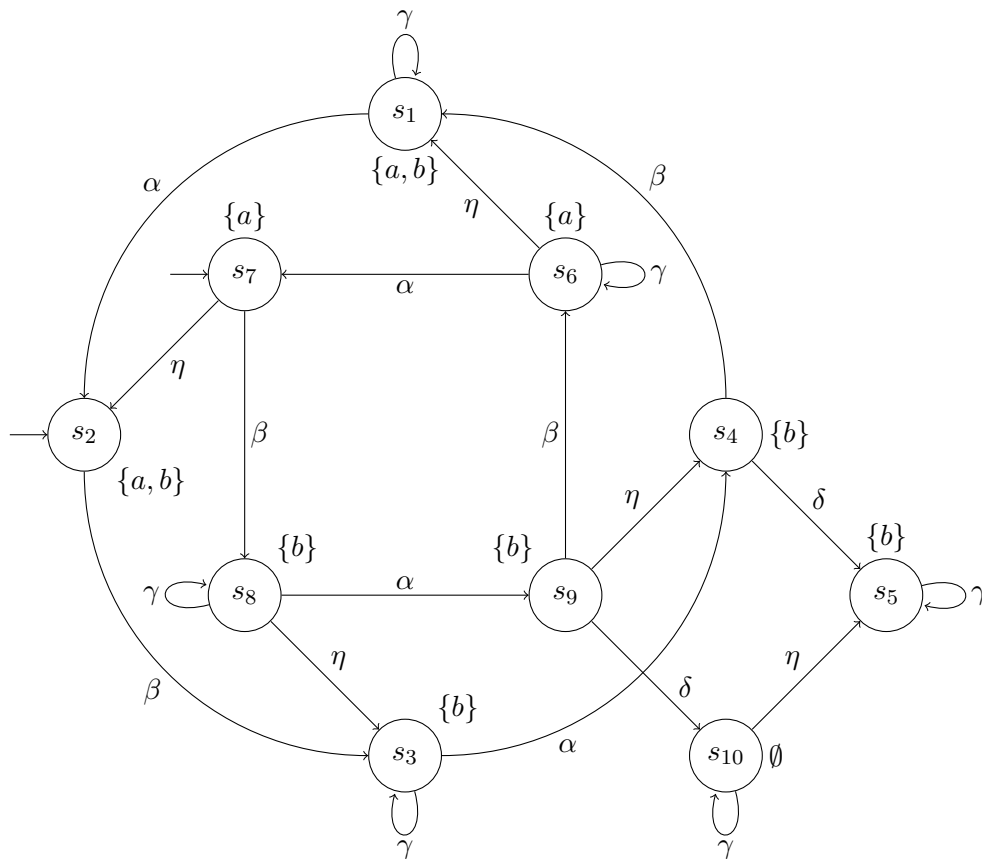
– Series 8 –

Hand in on **18 June** before the exercise class.

Exercise 1

(4 points)

Consider the following transition system TS .



(i) For each of the following ample sets, indicate whether it satisfies conditions (A1) to (A3). Justify your answer in case a condition is violated.

a) $ample(s_6) = \{\alpha, \gamma\}$

b) $ample(s_7) = \{\beta\}$

c) $ample(s_8) = \{\alpha\}$

d) $ample(s_9) = \{\beta, \delta, \eta\}$

e) $ample(s_{10}) = \{\gamma, \eta\}$

and $ample(s_i) = Act(s_i)$ for all other states.

(ii) Check if condition (A4) is met if the ample sets are chosen according to (i). Justify your answer.

- (iii) If the conditions (A1) through (A4) do not hold, provide a minimal extension of the ample sets that fixes this issue. Justify your answer.

Solution:

- (i)
- η is independent of α , β , γ and δ .
 - δ is independent of α , γ , η
 - α and γ are dependent and so are β and δ .
 - α and γ are stutter actions.
- a) $ample(s_6) = \{\alpha, \gamma\}$
- (A1) Yes.
- (A2) Yes.
- (A3) Yes.
- b) $ample(s_7) = \{\beta\}$
- (A1) Yes.
- (A2) Yes.
- (A3) No.
- c) $ample(s_8) = \{\alpha\}$
- (A1) Yes.
- (A2) No. $s_8 \xrightarrow{\gamma} s_8$ and γ is dependent from α implies γ must be in $ample(s_8)$.
- (A3) Yes.
- d) $ample(s_9) = \{\beta, \delta, \eta\}$
- (A1) Yes.
- (A2) Yes.
- (A3) Yes.
- e) $ample(s_{10}) = \{\gamma, \eta\}$
- (A1) Yes.
- (A2) Yes.
- (A3) Yes.
- (ii) Condition (A4) is violated in s_6 since there is a γ loop in the reduced system in which η is continuously enabled yet never contained in an ample set.
- (iii) First of all, $ample(s_7)$ should be $\{\beta, \eta\}$ to fix condition (A3). $ample(s_6)$ has to be extended with η to fix (A4). To make (A2) hold as well, $ample(s_8)$ needs to be extended with γ and because of the resulting γ loop and (A4) also with η .

In summary, the following ample sets are a minimal extension to make (A1) - (A4) hold:

- $ample(s_6) = \{\alpha, \gamma\} \cup \{\eta\}$
- $ample(s_7) = \{\beta\} \cup \{\eta\}$
- $ample(s_8) = \{\alpha\} \cup \{\gamma, \eta\}$

Exercise 2

(2 points)

Definition 1 Let $TS_i = (S_i, Act_i, \rightarrow_i, I_i, AP, L_i)$ be transition systems for $i \in \{1, 2\}$. A normed simulation for (TS_1, TS_2) is a triple $(\mathcal{R}, \nu_1, \nu_2)$ consisting of a binary relation $\mathcal{R} \in S_1 \times S_2$ such that:

$$\forall s_1 \in I_1 \exists s_2 \in I_2 (s_1, s_2) \in \mathcal{R}$$

and functions $\nu_1, \nu_2: S_1 \times S_2 \rightarrow \mathbb{N}$ such that for all $(s_1, s_2) \in \mathcal{R}$:

(i) $L_1(s_1) = L_2(s_2)$

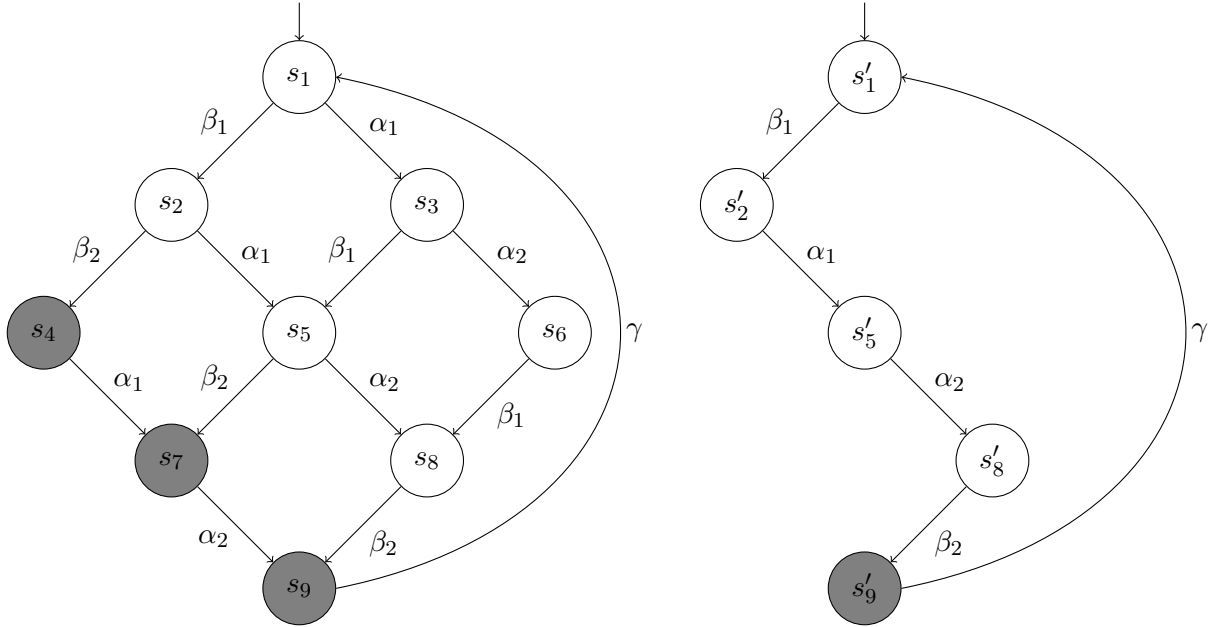
(ii) For all $s'_1 \in \text{Post}(s_1)$, at least one of the following three conditions holds:

- (1) $\exists s'_2 \in \text{Post}(s_2)$ such that $(s'_1, s'_2) \in \mathcal{R}$
- (2) $(s'_1, s_2) \in \mathcal{R}$ and $\nu_1(s'_1, s_2) < \nu_1(s_1, s_2)$
- (3) $\exists s'_2 \in \text{Post}(s_2)$ such that $(s_1, s'_2) \in \mathcal{R}$ and $\nu_2(s_1, s'_2) < \nu_2(s_1, s_2)$

A normed bisimulation for (TS_1, TS_2) is a normed simulation $(\mathcal{R}, \nu_1, \nu_2)$ for (TS_1, TS_2) such that $(\mathcal{R}^{-1}, \nu_1^-, \nu_2^-)$ is a normed simulation for (TS_2, TS_1) . Here, ν_i^- denotes the function $S_2 \times S_1 \rightarrow \mathbb{N}$ that results from ν_i by swapping the arguments, i.e., $\nu_i^-(u, v) = \nu_i(v, u)$ for all $u \in S_2$ and $v \in S_1$.

TS_1 and TS_2 are normed bisimilar, denoted $TS_1 \approx^n TS_2$, if there exists a normed bisimulation for (TS_1, TS_2) .

Consider the following two transition systems TS and $\widehat{TS}$ where $\widehat{TS}$ results from TS by choosing the appropriate ample sets. States of equal color are labeled equally.

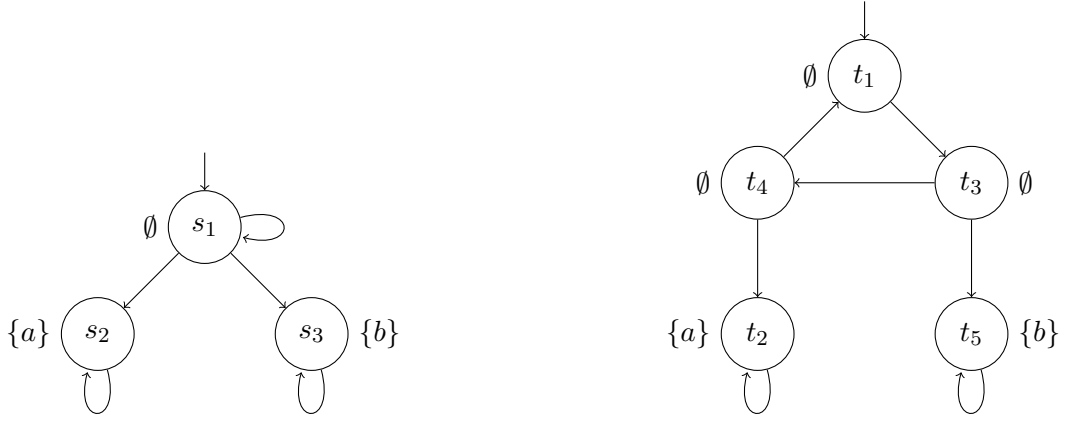


(i) How is normed bisimulation equivalence related to divergence-sensitive stutter bisimulation equivalence?

(ii) Provide a normed bisimulation for $(TS, \widehat{TS})$.

Solution:

(i) Normed bisimulation can be thought of as a divergence-sensitive stutter bisimulation with an upper bound on the number of allowed stutter steps. Hence, normed bisimulation equivalence is finer than divergence-sensitive stutter bisimulation (for a proof, see proof of Lemma 7.122 in the book). To see that normed bisimulation is strictly finer, consider the following two transition systems.



Then $s_1 \approx_{TS}^{div} t_1$, but $s_1 \not\approx_{TS}^n t_1$.

Proof: Assume there exists a normed bisimulation $(\mathcal{R}, \nu_1, \nu_2)$ with $(s_1, t_1) \in \mathcal{R}$. Since $s_1 \rightarrow s_2$, from the definition of normed simulation we have $(s_1, t_3) \in \mathcal{R}$ and $\nu_2(s_1, t_3) < \nu_2(s_1, t_1)$ (since there is no other way for t_1 to try and copy the step of s_1). But then we have $s_1 \rightarrow s_2$ and by the same argument as above it follows that $(s_1, t_4) \in \mathcal{R}$ and $\nu_2(s_1, t_4) < \nu_2(s_1, t_3)$. Finally, since $s_1 \rightarrow s_3$, we need to have $\nu_2(s_1, t_1) < \nu_2(s_1, t_4)$.

In summary, we have $\nu_2(s_1, t_1) < \nu_2(s_1, t_4) < \nu_2(s_1, t_3) < \nu_2(s_1, t_1)$, which is a contradiction.

(ii) Consider the normed simulation for $(TS, \widehat{TS})$ given by

$$\mathcal{R} = (\{s_2, s_5, s_8\} \times \{s'_2, s'_5, s'_8\}) \cup (\{s_1, s_3, s_6\} \times \{s'_1\}) \cup (\{s_4, s_7, s_9\} \times \{s'_9\})$$

and

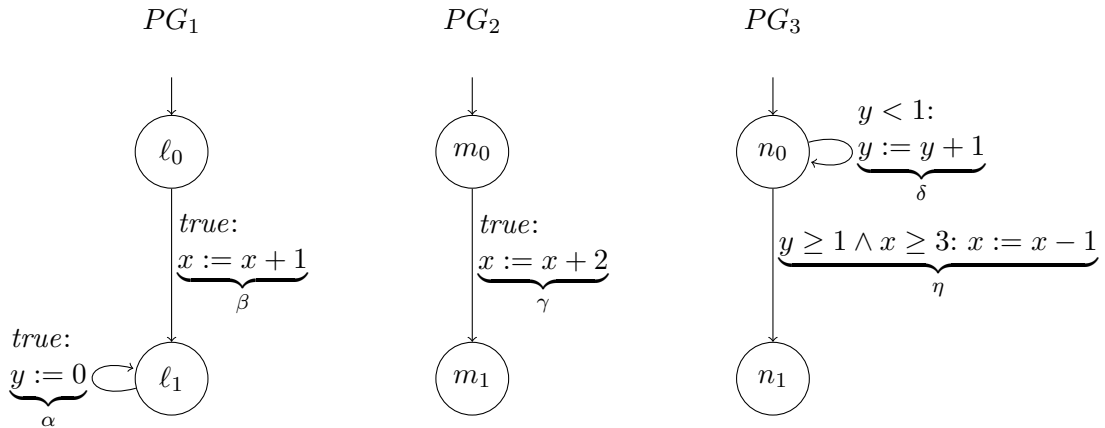
ν_1	s'_1	s'_2	s'_5	s'_8	s'_9	ν_2	s'_1	s'_2	s'_5	s'_8	s'_9
s_1	2					s_1					
s_2				2		s_2		2	1	0	
s_3	1					s_3					
s_4					2	s_4					2
s_5				1		s_5		2	1	0	
s_6	0					s_6					
s_7					1	s_7					1
s_8		2	1	0		s_8		2	1	0	
s_9					0	s_9					0

Claim: $(\mathcal{R}, \nu_1, \nu_2)$ is a normed bisimulation for $(TS, \widehat{TS})$.

Exercise 3

(4 points)

Consider the following three program graphs PG_1, PG_2, PG_3 over the shared variables x and y .



- (i) Prove or refute that the invariant $\varphi = \Box \neg n_1$ holds on $TS(PG_1 \parallel PG_2 \parallel PG_3)$ (where only n_1 is considered as an atomic proposition) with the initial condition $x = 0 \wedge y = 0$. For this, use the POR-based algorithm presented in the lecture (slide 151); in particular use the presented method to derive ample sets (slide 219) and the local criterion (slide 262) for (A2). Whenever you are required to pick a process (i.e., program graph) by any of the algorithms, try PG_2 first, then PG_1 and only then PG_3 . Choosing the order of explored actions (in the ample set) is up to you. Write down all steps that you performed.

Solution:

Let $\langle l_1, l_2, l_3, a, b \rangle$ denote the state where PG_i is in location l_i , the value of x is a and the value of y is b . Clearly, the only relevant atomic proposition is n_1 .

- (1) $\pi := \varepsilon, V := \emptyset$
- (2) select (only) initial state $s_1 := \langle \ell_0, m_0, n_0, 0, 0 \rangle, \pi := s_1, V := \{s_1\}$
- (3) compute $ample(s_1)$
- (4) try the actions of PG_2 first, $ample(s_1) = \{\gamma\} = Act(s_1) \cap Act(PG_2)$ satisfies the local criterion for (A2) and also (A1) and (A3) so choose it
- (5) take $s = s_1$
- (6) choose γ and compute $\gamma(s_1) = \langle \ell_0, m_1, n_0, 2, 0 \rangle =: s_2 \notin V$
- (7) $\varepsilon := s_1 s_2, V := \{s_1, s_2\}$
- (8) compute $ample(s_2)$
- (9) since PG_2 has no possible actions, try the actions of PG_1 first, $ample(s_2) = \{\beta\} = Act(s_2) \cap Act(PG_1)$ satisfies the local criterion for (A2) and also (A1) and (A3) so choose it
- (10) take $s = s_2$
- (11) compute $\beta(s_2) = \langle \ell_1, m_1, n_0, 3, 0 \rangle =: s_3 \notin V$
- (12) $\varepsilon := s_1 s_2 s_3, V := \{s_1, s_2, s_3\}$
- (13) compute $ample(s_3)$
- (14) since PG_2 has no possible actions, try the actions of PG_1 first, $ample(s_3) = \{\alpha\} = Act(s_3) \cap Act(PG_1)$ satisfies (A1) and (A3), but not the local criterion for (A2) since, e.g., α and δ are dependant; by the same reasoning choosing the actions of PG_3 will not work, so s_3 needs to be fully expanded, i.e., $ample(s_3) = \{\alpha, \delta, \eta\}$. (Even if (A2) was satisfied, α produces a backward edge by means of a self-loop and the state needs to be fully expanded because of (A4).)
- (15) take $s = s_3$
- (16) compute $\delta(s_3) = \langle \ell_1, m_1, n_0, 3, 1 \rangle =: s_4 \notin V$
- (17) $\varepsilon := s_1 s_2 s_3 s_4, V := \{s_1, s_2, s_3, s_4\}$
- (18) compute $ample(s_4)$
- (19) since PG_2 has no possible actions, try the actions of PG_1 first, $ample(s_4) = \{\alpha\} = Act(s_4) \cap Act(PG_1)$ satisfies (A1) and (A3), but not the local criterion for (A2) since, e.g., α and η are dependant; by the same reasoning choosing the actions of PG_3 will not work (in fact, η is not a stutter action, so choosing $ample(s_4) = Act(s_4) \cap Act(PG_3)$ would also violate (A3)), so s_4 needs to be fully expanded, i.e., $ample(s_4) = \{\alpha, \eta\}$
- (20) take $s = s_4$

- (21) compute $\eta(s_4) = \langle \ell_1, m_1, n_1, 2, 1 \rangle =: s_5 \notin V$
- (22) $\varepsilon := s_1 s_2 s_3 s_4 s_5$, $V := \{s_1, s_2, s_3, s_4, s_5\}$
- (23) compute $ample(s_5)$
- (24) since neither PG_1 nor PG_2 have any actions enabled in s_5 , choosing the actions of PG_1 amounts to fully exploring the state, which always satisfies (A1) to (A3), so $ample(s_5) = \{\alpha\} = Act(s_5)$
- (25) take $s = s_5$
- (26) $s_5 \models n_1$, so **return** “invariant violated”