Aalto University, Department of Information and Computer Science Antti Siirtola

T-79.4302 Parallel and Distributed Systems
Examination, 18 December 2013

Write down on every answer sheet: the name of the course, the course code, the date, your name, your student id, and your signature.

Calculators are NOT allowed.

To pass the course, you also need to have passed the quizes and home assignments in Autumn 2013.

Assignment 1. Consider the following Promela model.

```
chan msg = [1] of {bit};
bit t = 1;

active proctype P1()
{
    bit i = 0;
a:
    msg!i;
    do
    :: atomic{(t == i) -> i = (i + 1) % 2; goto a}
    :: msg!i
    od
}

active proctype P2()
{
    do
    :: msg?t
    od
}
```

(a) What pieces of data are needed to identify a single state of this model? Design a compact state vector representation for the states of the model. (2pt)

(b) Construct the reachability graph of the model. (6pt)

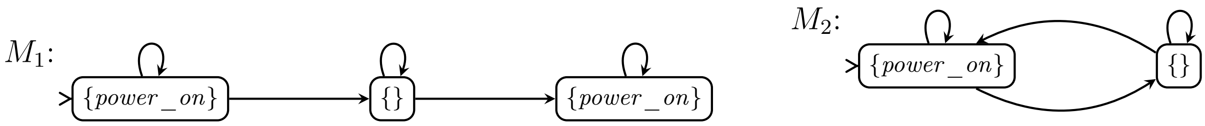
(c) Describe briefly an algorithm that inputs a Kripke structure (S, s^0, R, L) and a predicate (function) $F : S \mapsto \{0, 1\}$. The algorithm outputs either "fail", if there is a reachable state s such that $F(s) = 1$, or "pass" if $F(s) = 0$ for all reachable states s. (4pt)

Assignment 2.

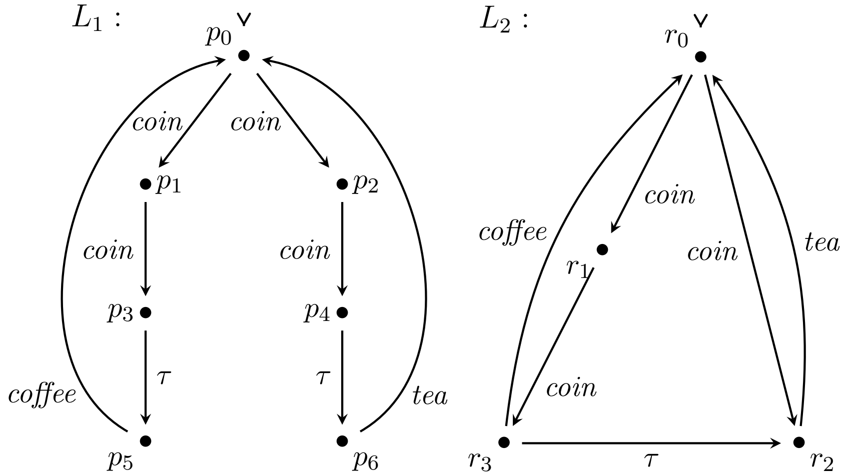
(a) Write an LTL formula ϕ that is satisfied by an infinite word π, if and only if π is an alternating sequence of atomic propositions cs_0 and cs_1, i.e., $\pi = cs_0\, cs_1\, cs_0\, cs_1\, cs_0\, cs_1\, \ldots$. Write also a past safety formula ψ that is satisfied by a finite word π, if and only if π is an alternating sequence of atomic propositions cs_0 and cs_1, i.e., π is a finite prefix of $cs_0\, cs_1\, cs_0\, cs_1\, cs_0\, cs_1\, \ldots$. (4pt)

(b) Formalise the following property both as an LTL and a past safety formula: "if there is an acknowledgement, then a message has been sent". (4pt)

(c) Devise an LTL formula that is satisfied by the Kripke structure M_1 but not by the Kripke structure M_2. Devise also a past safety formula that is satisfied by M_1 but not by M_2. (4pt)



Assignment 3. Consider the following LTSs over $\Sigma = \{coin, coffee, tea\}$.



(a) Construct a deterministic FSA A_2 that recognizes the language $traces(L_2)$. (3pt)

(b) By complementing A_2, construct a deterministic FSA $\overline{A}_2$ that recognizes the language $\Sigma^* \setminus traces(L_2)$ (3pt)

(c) Consider $\overline{A}_2$ as an LTS $\overline{L}_2$ and construct the parallel composition (asynchronous product) $\overline{L}_2 \parallel L_1$. (3pt)

(d) By analysing the parallel composition $\overline{L}_2 \parallel L_1$, decide whether $L_1 \leq_{tr} L_2$? Explain how you arrived at the answer. (3pt)

Assignment 4. In each of the following cases, decide whether the claim is true or not. Give a short proof if the claim is true. Otherwise, if the claim is false, provide a counter example, i.e., find LTSs that satisfy the "if" part but do not satisfy the "then" part.

(a) If $L_1 \sim L_2$, then L_1 has a reachable deadlock state if and only if L_2 has a reachable deadlock state. (3pt)

(b) If $L_1 \not\leq_{tr} L_2$, then $L_1 \parallel L \not\leq_{tr} L_2 \parallel L$ for every LTS L. (3pt)

(c) If $L_1 \leq_{sim} L_2$ and $L_2 \leq_{sim} L_1$, then $L_1 \sim L_2$. (3pt)

(d) If $L_1 \leq_{sim} L_2$ and $L_2 \leq_{sim} L_3$, then $L_1 \leq_{sim} L_3$. (3pt)

Please remember to give course feedback by 23 December — find a link to the online feedback form in Noppa.