Aalto University, Department of Information and Computer Science Antti Siirtola

T-79.4302 Parallel and Distributed Systems
Examination, 17 February 2014

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.

```
mtype = {foo, bar};
chan msg = [1] of {mtype, bit};

active proctype P1()
{
    bit i = 0;
    bit j = 0;
    do
    :: msg!foo(i)
    :: atomic{msg?bar(j) -> i = (j + 1) % 2; j = 0}
    od
}

active proctype P2()
{
    bit k = 0;
    do
    :: msg?foo(k) ->
        if
        :: atomic{skip; k = 0};
        :: atomic{msg!bar(k); k = 0};
        fi
    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) Present and describe an algorithm that inputs a Kripke structure (S, s^0, R, L) and outputs either "deadlock", if there is a reachable deadlock state s, or "live" if there is no reachable deadlock state. (4pt)

Assignment 2.

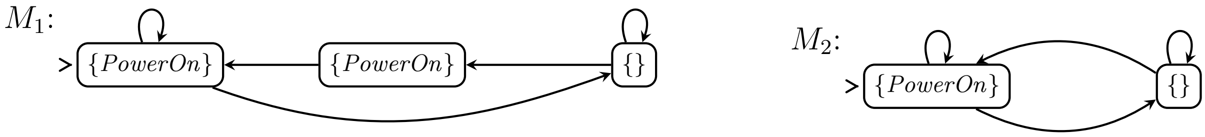
(a) Describe informally but as precisely as possible what do the following temporal formulae mean:

(1) $\mathbf{G}\,(\mathit{PowerOn} \rightarrow (\mathit{PowerOn}\,\mathbf{U}\,(\mathit{SwitchOff} \wedge \mathbf{F}\,\neg \mathit{PowerOn})))$, (2pt)

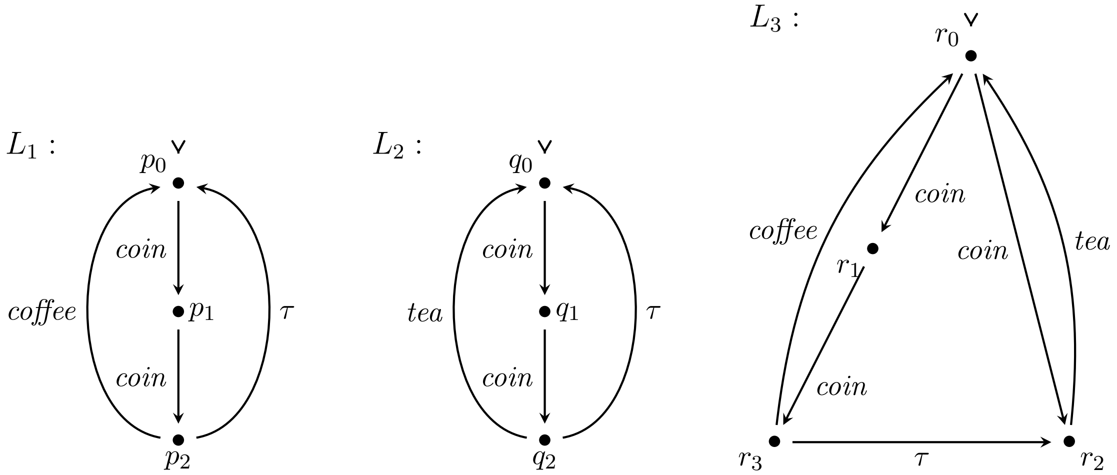
(2) $\mathbf{G}\,(\mathit{Ack} \rightarrow \mathbf{Y}\,(\neg \mathit{Ack}\,\mathbf{S}\,\mathit{Send}))$. (2pt)

(b) Formalise the following property both as an LTL and a past safety formula: "at most two messages are 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 L_1, L_2, L_3 over alphabets $\Sigma_1 = \{coin, coffee\}$, $\Sigma_2 = \{coin, tea\}$, $\Sigma_3 = \{coin, coffee, tea\}$, respectively.



(a) Construct the reachable part of the parallel composition (asynchronous product) $L_{12} := L_1 \parallel L_2$. (3pt)

(b) Construct a deterministic FSA A_3 that recognizes the language $traces(L_3)$. (3pt)

(c) By complementing A_3, construct a deterministic FSA $\overline{A}_3$ that recognizes the language $\Sigma^* \setminus traces(L_3)$ (3pt)

(d) Consider $\overline{A}_3$ as an LTS $\overline{L}_3$ and construct the parallel composition $\overline{L}_3 \parallel L_{12}$. By analysing the parallel composition $\overline{L}_3 \parallel L_{12}$, decide whether $(L_1 \parallel L_2) \leq_{tr} L_3$? Explain how you arrived at the answer. (3pt)

Assignment 4.

(a) Describe the steps of a typical verification process, i.e., how would you proceed if your task is to verify the design of a software system? (4pt)

(c) What is meant by the state explosion problem? Describe briefly two techniques that may help you in alleviating the state explosion problem. (4pt)

(b) What are the major benefits of verification methods, e.g., model checking, over traditional testing-based approaches? Besides the state explosion problem, what kind of problems/inconveniences you may encounter during the verification process? (4pt)