Prof. Dr. Ir. Joost-Pieter Katoen
apl. Prof. Dr. Thomas Noll
S. Chakraborty, B. Kaminski, H. Wu

Concurrency Theory WS 2013/2014

— 1st Exam —

First Name: ______________________

Second Name: ______________________

Matriculation Number: ______________________

Degree Programme (please mark):

- ○ CS Bachelor
- ○ CS Master
- ○ CS Lehramt
- ○ SSE Master
- ○ Other: ______________________

General Information:

- Mark every sheet with your **matriculation number**.
- Check that your copy of the exam consists of **14 sheets (28 pages)**.
- Duration of exam: **120 minutes**.
- No helping materials (e.g. books, notes, slides) are permitted.
- Give your solution on the respective sheet. Also use the backside if necessary. If you need more paper, ask the assistants.
- Write with blue or black ink; do **not** use a pencil or red ink.
- Make sure all electronic devices are switched off and are nowhere near you.
- Any attempt at deception leads to failure for this exam, even if it is detected only later.

	Σ Points	**Points obtained**
Task 1	**16**	
Task 2	**21**	
Task 3	**28**	
Task 4	**21**	
Task 5	**23**	
Task 6	**11**	
Σ	**120**	

Task 1 (Modelling with CCS) (10 + 6 Points)

The task is to model a car's central locking system, which has following three components:

- a *door* can be either open or closed, if it is closed and locked, it cannot be opened.
- a *locker* for the door which can only be activated if the door is closed, otherwise it will trigger an alarm;
- a *key* (has one button) which controls the locker (activate or deactivate).

(a) Give a CCS process for the system, which should be functionally correct and deadlock–free!

(b) Give a partial LTS of your CCS definition which illustrates that the user first closes the door, and then presses the button to lock the door! Additionally argue why the user immediately afterwards cannot open the door!

Task 2 (Labeled Transition Systems) (15 + 6 + 6 Points)

(a) Consider the following CCS process definition:

$$\begin{aligned} A &= (B \parallel C) \setminus \{\mathrm{com}\} \\ B &= D + E \\ D &= \mathrm{com}.B \\ E &= \mathrm{a}.\mathrm{com}.B \\ C &= \overline{\mathrm{com}}.C + \mathrm{b}.nil \end{aligned}$$

Derive all legal outgoing transitions $A \xrightarrow{\alpha} A'$ by giving their derivation tree!

(b) Reconsider the CCS process definition from Task 2 (a):

$$\begin{aligned} A &= (B \parallel C) \setminus \{\text{com}\} \\ B &= D + E \\ D &= \text{com}.B \\ E &= \text{a.com}.B \\ C &= \overline{\text{com}}.C + \text{b}.nil \end{aligned}$$

Draw LTS(A) and label the nodes with the corresponding CCS processes!

(c) Give the trace language $\mathrm{Tr}(A)$ of A!

Task 3 (HML and Bisimulation) (15 + 7 Points)

Consider the following three CCS processes A, D and I.

$A = a.E + a.b.C + b.C$	$D = a.E + a.F + b.(G + H)$	$I = a.J + b.K$
$B = b.C + a.b.C$	$E = a.F + b.H$	$J = a.L + b.K$
$C = c.A$	$F = b.G$	$K = c.(I + L)$
	$G = c.D$	$L = b.M$
	$H = c.E$	$M = c.I$

(a) Draw the LTS for A, D and I, respectively! Prove or disprove: $A \sim D$, $A \sim I$ and $D \sim I$.

For proving or disproving that two processes are strongly bisimilar, you may use the game characterization of bisimilarity. For disproving you may alternatively provide an HML formula which is satisfied by only one of two processes.

(b) Express the property that action b will eventually occur after any occurrence of action a in HML, and check whether D satisfies this property or not!

Task 4 (Preservation of Strong Bisimilarity) (6+15 Points)

(a) Let $\circ$ be a CCS operator with the following semantics:

$$\text{(comp)}\ \frac{P[nil \mapsto Q] \xrightarrow{\alpha} P'}{P \circ Q \xrightarrow{\alpha} P'},$$

where $P[nil \mapsto Q]$ is the process P in which every occurrence of *nil* is replaced by Q.

Prove or disprove: $\circ$ preserves strong bisimilarity, i.e. for any processes S, T and R with $S \sim T$ it holds that both $S \circ R \sim T \circ R$ and $R \circ S \sim R \circ T$.

(b) Let ; be a CCS operator with the following semantics:

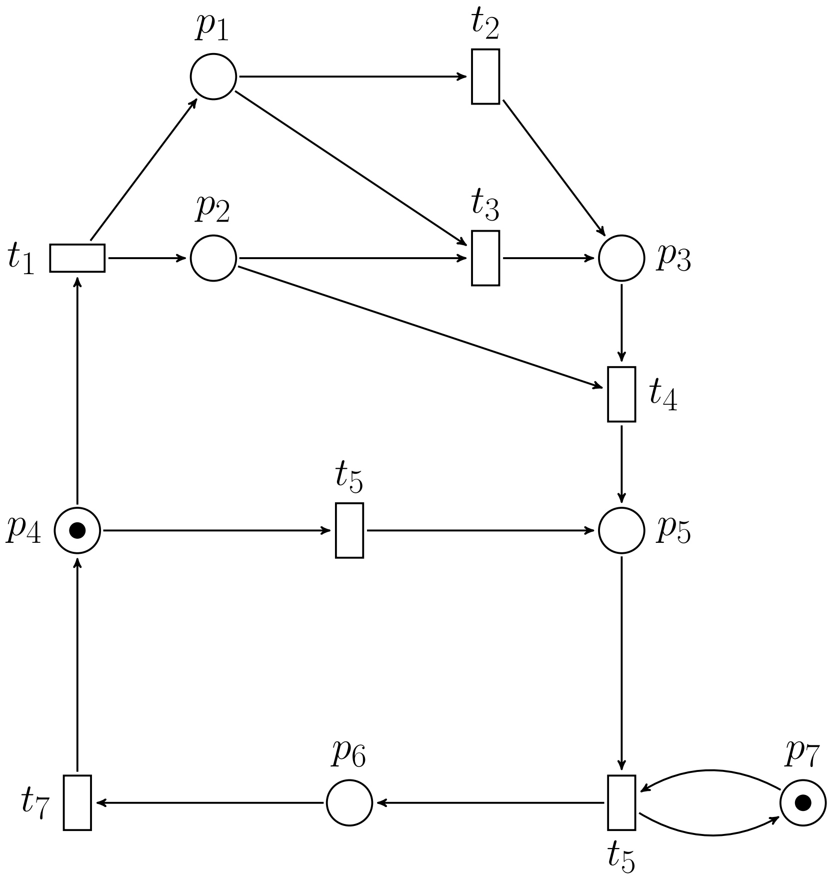
$$\text{(seq1)}\ \frac{P \xrightarrow{\alpha} P'}{P;Q \xrightarrow{\alpha} P';Q}$$

$$\text{(seq2)}\ \frac{P \not\rightarrow \qquad Q \xrightarrow{\alpha} Q'}{P;Q \xrightarrow{\alpha} Q'}$$

Prove or disprove: ; preserves strong bisimilarity, i.e. for any processes S, T and R with $S \sim T$ it holds that both $S;R \sim T;R$ and $R;S \sim R;T$.

Task 5 (Semantics of Petri nets) (8 + 6 + 9 Points)

Consider the following Petri net N:



(a) Give the marking graph of N!

(b) Give at least three distributed runs which cover at least four transitions of N! If N admits an infinite distributed run, provide it!

(c) Compute the McMillan prefix of the Petri net N and the cut–off transitions in the unfolding!

Task 6 (Petri net Acceptable Languages) (11 Points)

Let Σ be a finite alphabet and let $N = (P, T, F, M_0, \lambda)$ be a labelled Petri net in which all transitions in T are labelled by a labeling function $\lambda\colon T \to \Sigma$. Then the trace language $\mathrm{Tr}(N)$ of N is defined as the following set:

$$\Big\{ w = \lambda(a_1) \cdots \lambda(a_k) \;\Big|\; M_0 \xrightarrow{a_1} M_1 \xrightarrow{a_1} \cdots \xrightarrow{a_k} M_k \text{ is a complete sequential run of } N \Big\}.$$

A language $L \subseteq \Sigma$ is called Petri net recognizable, if there exists a labelled Petri net N such that $\mathrm{Tr}(N) = L$.

Provide a Petri net which recognizes the language $a^n c\, b^n$!