Prof. Dr. Ir. Joost-Pieter Katoen
apl. Prof. Dr. Thomas Noll
B. Kaminski, C. Matheja

Concurrency Theory WS 2015/2016

— 2nd 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 **10 sheets (20 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 detected only later.

	Σ Points	**Points obtained**
Task 1	**26**	
Task 2	**28**	
Task 3	**16**	
Task 4	**14**	
Task 5	**24**	
Task 6	**12**	
Σ	**120**	

Task 1 (Labeled Transition Systems) (18+5+3 Points)

(a) Consider the following CCS process definition:

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

Derive all legal outgoing transitions $A \xrightarrow{\alpha} A'$ (for some $A' \in \mathsf{Prc}$) by giving a corresponding derivation tree.

(b) Reconsider the CCS process definition from Task 1 (a):

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

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

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

Task 2 (HML and Bisimulation) (18+8+2 Points)

Consider the following CCS processes:

$$
\begin{array}{lll}
A = a.B + a.C & B = a.A + b.B + b.D & C = b.C + a.B + b.D \\
D = a.D + a.B & E = b.E + a.E + b.D & F = a.B + a.C \\
G = b.C + a.B + b.D & H = a.B + a.C + a.E & I = a.B + a.J \\
J = b.C + a.B + b.D + b.G & K = a.L + a.C & L = a.F + b.B + b.D + a.A
\end{array}
$$

(a) Draw $LTS(H)$, $LTS(I)$ and $LTS(K)$, respectively. Prove or disprove: $H \sim I$, $I \sim K$ and $H \sim K$, where $\sim$ denotes strong bisimilarity.

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) Provide a (possibly recursive) HML specification expressing that pattern *aba* is enabled in each state until action c is *eventually* enabled (hence it must be guaranteed that c is enabled at some point).

(c) Check whether H satisfies your HML specification provided in (b).

Task 3 (Trace Languages) (4+4+4+4 Points)

Consider the following CCS process:

$$\begin{aligned} Q &= (Q_1 || Q_3) \setminus \{c, d\} \\ Q_1 &= a.(Q_1 || c.b.\text{nil}) \; + \; \bar{d}.\text{nil} \\ Q_2 &= \bar{c}.Q_2 \\ Q_3 &= d.Q_2 \end{aligned}$$

(a) Provide the trace language $Tr(Q)$.

(b) Provide the *completed* trace language $CTr(Q)$.

(c) Prove or disprove: $Tr(Q)$ is a regular language.

(d) Prove or disprove: There exists a CCS process P such that $Tr(P)$ is infinite and the set of traces of P (except for the empty trace ε) and the set of *completed* traces of P coincide, i.e. $Tr(P) \setminus \{\varepsilon\} \; = \; CTr(P)$.

Task 4 (Preservation of Strong Bisimilarity) (14 Points)

Let $\gg$ be a binary CCS operator with the following semantics:

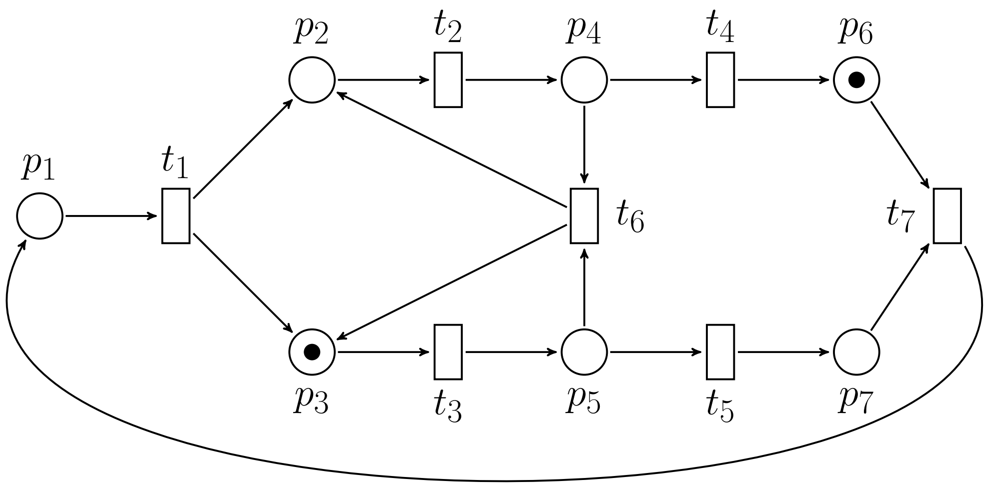
$$\text{(pref1)}\ \frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\alpha} Q'}{P \gg Q \xrightarrow{\alpha} P' \gg Q'}$$

$$\text{(pref2)}\ \frac{Q \xrightarrow{\alpha} Q'}{P \gg Q \xrightarrow{\alpha} Q'}$$

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

Task 5 (True Concurrency Semantics) (14+10 Points)

Consider the following elementary net N:



(a) Give the marking graph of N.

(b) Provide three non-isomorphic branching processes B_1, B_2, B_3 of N such that $B_1 \sqsubseteq B_2$ and $B_2 \not\sqsubseteq B_3 \not\sqsubseteq B_1$.

Task 6 (Petri nets vs. CCS) (12 Points)

Let N be an elementary net. We define the *trace language* $Tr(N)$ as the set of all traces of the marking graph of N.

Prove or disprove:

(a) For each one-bounded elementary net N there exists a CCS process P such that $Tr(N) = Tr(P)$.

(b) For each CCS process P there exists a one-bounded elementary net N such that $Tr(N) = Tr(P)$.