Concurrency Theory WS 2015/2016
Chair for Software Modeling and Verification
RWTH Aachen

Prof. Dr. Ir. Joost-Pieter Katoen
apl. Prof. Dr. Thomas Noll
Benjamin Kaminski, Christoph Matheja

Concurrency Theory WS 2015/2016

— Series 2 —

Hand in until November 16 before the exercise class.

Exercise 1 (CCS and LTS's) (3+1 Points)

(a) Decide whether the following CCS process definitions induce infinite LTS's and whether their trace languages are regular. Justify your answers.

i) $B = (B \parallel B) + a.\text{nil}$

ii) $D = a.(D \parallel b.\text{nil})$

iii) $B = (B \parallel B) \setminus \{a\} + a.\text{nil}$

(b) Prove or disprove: If a CCS process C is defined as $C = P$ where $C \parallel P'$ occurs as a subterm in P [1], then $\text{LTS}(C)$ is infinite.

Exercise 2 (Complementation of HML Formulae) (2+1+1 Points)

(a) Suppose it holds for some arbitrary but fixed HML formula F that $[\![F^c]\!] = S \setminus [\![F]\!]$ for every LTS $(S, Act, \rightarrow)$. Prove that $[\![([\alpha]F)^c]\!] = S \setminus [\![[\alpha]F]\!]$.

(b) Prove or disprove: $(F^c)^c$ and F is semantically equivalent for every HMF formula F.

(c) Prove or disprove: $(F^c)^c$ and F is syntactically equivalent for every HMF formula F.

Exercise 3 (Construction of HML Formulae) (1+1+1 Points)

Let $A = a.b.d.\text{nil} + a.c.d.\text{nil}$ and $B = a.(b.d.\text{nil} + c.d.\text{nil})$.

(a) Construct an HML formula which is satisfied by both A and B!

(b) Construct an HML formula which is satisfied by B but not by A!

(c) Construct an HML formula which is satisfied by A but not by B!

[1] P and P' are process expressions.