

Concurrency Theory WS 2015/2016 — Series 1 —

Hand in until November 2nd before the exercise class.

Exercise 1

(2 Points)

(3 Points)

Consider the following process definition:

 $B = a.\overline{a}.B + b.\overline{b}.B$

Draw LTS(B)! Also write down all necessary derivation trees for drawing LTS(B)!

Exercise 2

In this exercise, we extend CCS by a new syntactical element. Intuitively, the *sequential composition* P; Q of two processes P and Q means that first P is executed until no further rule is applicable and then Q is executed.

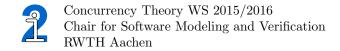
- (a) Extend the semantics of CCS (Definition 2.4) by rules for sequential composition.
- (b) Consider the following two process definitions:

$$C = (\overline{a}.Q \mid\mid P[a.nil / nil]) \setminus \{a, \overline{a}\},$$

$$C' = P; Q,$$

where P and Q are arbitrary processes, a does not occur in P and Q, and P[a.nil / nil] denotes the syntactic replacement of every occurrence of nil by a.nil. Prove or disprove: LTS(C) and LTS(C') are isomorphic.¹

 $^{^1\}mathrm{Two}\ \mathrm{LTS}$ are isomorphic if and only if they are identical up to the names of states and actions.



Exercise 3

(2 Points)

For any word w we write $v \preceq w$, if v is a prefix of w. The prefix set pref(L) of a language $L \subseteq \Sigma^*$ is defined as

$$\operatorname{pref}(L) = \{ v \in \Sigma^* \mid v \preceq w, \ w \in L \}.$$

A language L is called prefix-closed if L = pref(L).

Prove that every regular language can be prefix-completed while preserving regularity, i.e. prove that for every regular language L the language pref(L) is again regular!