Software Modeling and Verification
Lehrstuhl für Informatik 2
RWTH Aachen University

Prof. Dr. Ir. J.-P. Katoen
Priv.-Doz. Dr. T. Noll
T. Han, M. Neuhäußer

Modeling Concurrent and Probabilistic Systems

Winter Term 07/08

– Series 1 –

Hand in until October 26 before the exercise class.

Exercise 1 (3 points)

Construct the derivation tree for the τ-step in the LTS of the parallel two place buffer.

Exercise 2 (4 points)

Give the LTS of the following process definitions:

a) $A(a, b, c) = a.b.c.A(a, b, c) + a.\mathsf{nil}$

b) $B(a) = a.\overline{a}.\mathsf{nil} + a.\mathsf{nil}$

c) $C(a) = a.C(a) \mid\mid D$
$D = D$

d) $E = \mathsf{new}\ a\ \big(a.\mathsf{nil} \mid\mid F(a)\big)$
$F(a) = \overline{a}.F(a)$

Exercise 3 (4 points)

Which of the following process definitions induce an infinite LTS? Justify your answers!

a) $A = A$

b) $B(a) = \big(B(a) \mid\mid B(a)\big) + a.\mathsf{nil}$

c) $C(a) = \big(C(a) + C(a)\big) + a.\mathsf{nil}$

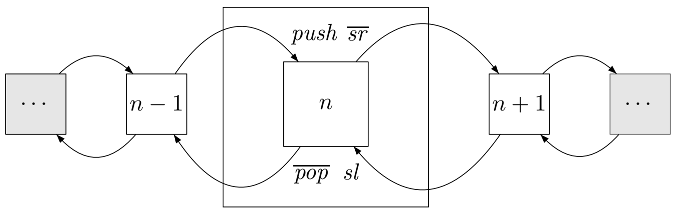
d) $D(a, b) = a.\big(D(a, b) \mid\mid b.\mathsf{nil}\big)$

Exercise 4 (3* points)

The aim of this exercise is to infer a finite stack data structure, i.e. a bounded LIFO queue:

To this aim, consider the following process definition of a shift register which stores any value received via actions *push* (or *sl*) and shifts its old value to the shift register on the right (on the left, resp.) via action $\overline{sr}$ (resp. $\overline{pop}$). Let $\vec{a} = (push, pop, sl, sr)$ and define

$$E(\vec{a}) = push.F(\vec{a}) + sl.F(\vec{a})$$
$$F(\vec{a}) = \overline{pop}.E(\vec{a}) + push.\overline{sr}.F(\vec{a})$$



Derive a process definition of a stack for at most three elements by appropriately composing three instances of the shift register! Construct the derivation tree of the *push*-transition of the empty stack!
Hint: You do not have to consider cases where the stack's capacity is exceeded!

---

*Additional exercise with extra points only.