AOR Priv.-Doz. Dr. Thomas Noll

Christina Jansen, Sabrina von Styp

Exercise 1 (CSP Semantics): (4 Points)

Consider the following CSP program c:

$c :=$
$\quad y := 4;$ **if** $(y > 0) \to ((x := y) \parallel (x := 3))$ **fi**
$\quad$ **do** $(x == 3 \wedge \alpha?x \to \beta!x) \,\square\, (x == 3 \to \alpha!y)$ **od**

Provide all "meanings" of c using the formal semantics of CSP as given in the lecture.

Exercise 2 (LTS and Deadlocks): (2+1 Points)

The aim of this exercise is to develop a (simplified) model of a car's central locking system. Assume the following components:

- a door which is either open or closed
- a locker for the door which can be activated if the door is not open (otherwise an alarm should be issued), and
- a key which controls the whole mechanism.

a) Design a corresponding process definition and give its transition system!

b) Check if the car locking system you developed in part a.) has a deadlock. If this is the case, provide a deadlock free specification of the system.

Exercise 3 (Parallel Composition of CCS): (2+3 Points)

An engineer is charged with developing an elevator control for a building with five floors, starting with a CCS model. His subspecification for requesting the elevator and selecting the target floor looks as follows:

$$Elevator(req, fl_1, \ldots, fl_5) = req.fl_1.Elevator(req, fl_1, \ldots, fl_5) + \ldots + req.fl_5.Elevator(req, fl_1, \ldots, fl_5).$$

A computer scientist who was called for supporting the engineer suggests the following solution instead:

$$Elevator(req, fl_1, \ldots, fl_5) = req.(fl_1.Elevator(req, fl_1, \ldots, fl_5) + \ldots + fl_5.Elevator(req, fl_1, \ldots, fl_5)).$$

a) Are both systems trace equivalent?

b) Test the elevator subsystem together with the specification of a user who would like to reach the fourth floor:

$$User(req, fl_4) = \overline{req}.\overline{fl_4}.nil.$$

Do both specifications of the elevator guarantee that the user is satisfied?