

## Tutorial for Cyber-Physical Systems - Discrete Models

### Exercise Sheet 6

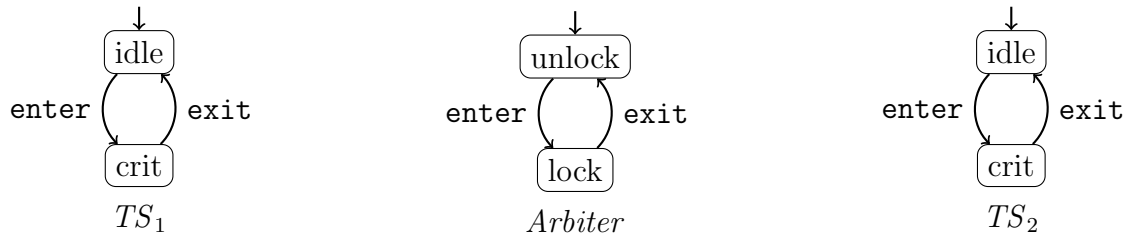
The goal of this exercise sheet is to familiarize yourself with different communication methods in transition systems.

#### Exercise 1: Mutual Exclusion without Request

6 Points

The goal of this exercise is to help you understand in detail the SOS-rules for parallel compositions.

The transition systems below describe a mutual-exclusion protocol with an arbiter. In contrast to the system discussed in the lecture, we omit the *request* action.



- (a) Draw the transition system for the pure interleaving  $TS_1 \parallel TS_2$ . There must be no synchronization between the two transition systems.

For every transition in the interleaving, justify why it must exist using one of the two SOS-rules for pure interleaving.

**Example:** The interleaving must contain the transition  $\langle \text{idle}, \text{idle} \rangle \xrightarrow{\text{enter}} \langle \text{crit}, \text{idle} \rangle$  due to the SOS-rule

$$\frac{\text{idle} \xrightarrow{\text{enter}}_1 \text{crit}}{\langle \text{idle}, \text{idle} \rangle \xrightarrow{\text{enter}} \langle \text{crit}, \text{idle} \rangle}$$

where  $\rightarrow_1$  is the transition relation for  $TS_1$ . This is an instance of the first of the two SOS-rules,

$$\frac{s_1 \xrightarrow{\alpha}_1 s'_1}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s'_1, s_2 \rangle}$$

where we set  $s_1 = \text{idle}$ ,  $\alpha = \text{enter}$ ,  $s'_1 = \text{crit}$  and  $s_2 = \text{idle}$ .

- (b) Draw the transition system for the parallel composition  $(TS_1 \parallel TS_2) \parallel \text{Arbiter}$  of the transition system from (a) with the arbiter. The transition systems must synchronize (“handshake”) on the actions  $\{\text{enter}, \text{exit}\}$ .

For every transition in the composition, justify why it must exist using one of the three SOS-rules for the synchronization operator.

**Exercise 2: Alternating Bit Protocol**

1 Point

*In this exercise we work with a system whose components communicate over asynchronous and synchronous channels.*

Consider the Alternating Bit Protocol as discussed in the lecture. Why do the sender and the timer communicate synchronously? I.e., why do they not use an asynchronous channel like sender and receiver?