Sheet 1

Due date: October 31, 2024

Exercise 1 All the King's Media (4 Points)

Consider a CCS specification $(Cell, Eq_1, Eq_2)$ of a one-place buffer, where Eq_1 and Eq_2 are the following equations:

$$\begin{aligned} Cell &= in(x).Cell_1(x) \\ Cell_1(x) &= \overline{out}(x).Cell \end{aligned}$$

On the basis of these equations, give CCS specifications of the following variants of a two-place buffer

1. FIFO (First-in-first-out) values are output in the order in which they are received.
2. Multiset: The order of transmission is unspecified, as long as all values received are eventually output, and no new values are generated.

In both variants, draw the corresponding labelled transition system, assuming that x takes only two values, 0 and 1.

Exercise 2 Mutex (5 Points)

Mutex (*mutual exclusion*) is the problem of regulating access to *critical regions* for concurrent processes.

Briefly, the problem is about two or more concurrent processes that are not allowed to enter certain critical portions of code at the same time (such as writing to shared memory). To this end, one uses a dedicated process called a *semaphore*, which ensures that at any given point in time, at most one of the processes is in its critical region.

Consider the processes

$$\begin{aligned} User &= \bar{p}.enter.exit.\bar{v}.User \\ Sem &= p.v.Sem \end{aligned}$$

which represent a user process and a semaphore, respectively.

1. Derive the LTS for the process
   $$Mutex_1 = (User \mid Sem)\backslash\{p, v\}$$
   according to the rules of the semantics, and represent the LTS graphically.
2. Proceed in the same way with the process
   $$Mutex_2 = ((User_1 \mid Sem) \mid User_2)\backslash\{p, v\}.$$
   where $User_1 = \bar{p}.enter_1.exit_1.\bar{v}.User_1,\ User_2 = \bar{p}.enter_2.exit_2.\bar{v}.User_2.$

3. Argue why Mutex_2 solves the Mutex problem.

4. Argue why the modified definitions

$$\begin{aligned} \mathit{User}'_1 &= \bar{p}.\mathit{enter}_1.\bar{v}.\mathit{exit}_1.\,\mathit{User}'_1 \\ \mathit{User}'_2 &= \bar{p}.\mathit{enter}_2.\bar{v}.\mathit{exit}_2.\,\mathit{User}'_2 \end{aligned}$$

   of User_1 and User_2 do not solve the Mutex problems correctly.

Exercise 3 Parallelism (4 Points)

Define

$$Q = a.\,b.\,Q.$$

For $n \geq 0$, let P_n be the process

$$\underbrace{Q \mid \cdots \mid Q}_{n \text{ mal}}.$$

Express the size of the LTS represented by P_n according to the CCS semantics as a closed expression in n, and prove this formula by induction on n, using the rules of the semantics.

Exercise 4 From Semantics to Syntax (7 Points)

We refer to the fragment of CCS featuring only prefix and nondeterministic choice (sums) as *Basic Process Algebra* (BPA).

1. For a given LTS T, construct a process definition in BPA whose semantics is T.

2. Conclude that there is a semantics-preserving (what does that mean?) translation of CCS process definitions into BPA process definitions.

3. Show that the LTS defined by a BPA process definition is of at most linear size. *Hint:* This follows from a more precise description of this LTS, proved by induction on semantic derivations.

4. We write CCS/BPA for the fragment of CCS in which the right-hand sides of *recursive* definitions must be BPA terms (while unrestricted CCS terms are allowed as the right-hand sides of non-recursive definitions). Using Exercise 3 and the previous items of the present exercise, what can you say about the worst-case size of the translation of a CCS/BPA process definition into a BPA process definition?